Topic	DATA SCIENCE - 2	
Class Description	**Students will be applying machine learning algorithms and statistics to filter out more planets.**	
Class	**PRO C132**	
Class time	**45 mins**	
Goal	● Plot various charts on our data ● Filter out more planets ● Apply machine learning algorithms to real-world data for multiple use cases	
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**	**5 mins** **15 mins** **20 mins** **05 mins**
Credit & Permissions:	**Exoplanet Exploration by NASA** **Plotly Express (Open-source MIT license)** **Scikit Learn (Open source, commercially usable - BSD license)**	
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 student.

Teacher Action	Student Action
In the last class, we learned about gravity, found out the gravity for all the exoplanets and filtered out data based on that!	

Can you tell me what is the formula of gravity that we applied in the last class? Excellent! Now in today's class, we will try to filter out some more planets by applying machine learning algorithms and statistics, and also we will try to find if there are any interesting relations between the data points. Are you excited? Let's start!	**ESR:** It is the mass of the planet divided by the square of the radius of the planet. We then multiply this with the gravitational constant.
Teacher Ends Slideshow	
TEACHER-LED ACTIVITY - 10 mins	
Teacher Initiates Screen Share	
<u>ACTIVITY</u> ● **To plot charts and apply statistics** ● **To make the student understand the different planet types** ● **To apply Machine Learning Algorithm (Clustering) on the data**	
Teacher Action	**Student Action**
***Note**: Open [Teacher Activity 1]() for boilerplate code.* The previous class code is given as boilerplate code or we can continue in the same colab file. We found out the gravity of all the planets and then based on the fact that our body can withstand 90 times more gravity than what Earth offers, we found a list of 3,951	

planets that have a gravity of 10 times or lower than what we have on Earth, just to be comfortable.	

```
low_gravity_planets = []
for index, gravity in enumerate(planet_gravity):
  if gravity < 10:
    low_gravity_planets.append(planet_data_rows[index])

print(len(low_gravity_planets))
```

```
1012
```

```
low_gravity_planets = []
for index, gravity in enumerate(planet_gravity):
  if gravity < 100:
    low_gravity_planets.append(planet_data_rows[index])

print(len(low_gravity_planets))
```

```
3951
```

Now, if we look at our headers, we can see that we have a header as **planet_type**.	

```
print(headers)
```

```
['row_num', 'name', 'light_years_from_earth', 'planet_mass', 'stellar_magnitude', 'discovery_date', 'planet_type',
```

Let's try to find out different values of **planet_type**. Here, we are creating an empty list to store the values of all the **planet_types** and we are iterating over our **planet_data_rows** to append each planet's **planet_type** (7th element of the list) in the empty list. Finally, we are printing the result. We want to print the **planet_type_values** but it contains all the planet types and there will be a lot of duplicates. To get all the unique values from the list, we have the **set()** function. Finally, we are converting the result again into a list with the **list()** function.	

```
planet_type_values = []
for planet_data in planet_data_rows:
  planet_type_values.append(planet_data[6])

print(list(set(planet_type_values)))
```

```
['Neptune-like', 'Terrestrial', 'Super Earth', 'Gas Giant']
```

Now that we have the types of planets that are out there, let's understand these terms:

Neptune-like: These planets are like Neptune! They are big and they are also made of Ice.

Super-Earth: These are the planets that have a mass greater than Earth but smaller than that of Neptune! (**Neptune is 17 times Earth**)

Terrestrial: It is a planet that is composed primarily of silicate rocks or metals. **(Like Earth, Mars)**

Gas Giant: These are the planets that are composed of Gas. **(Hydrogen and Helium)**

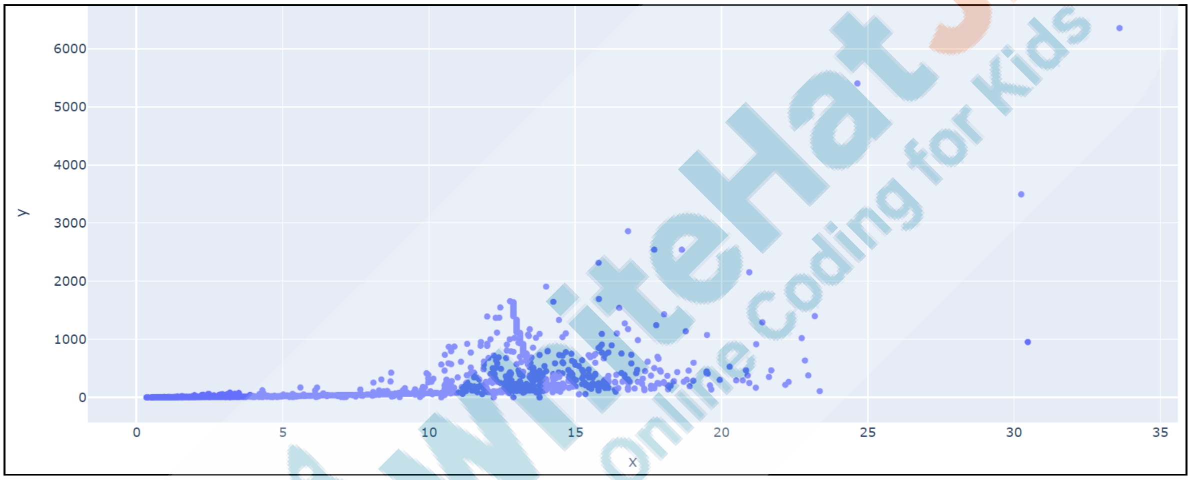
Based on this, let's try to do some clustering to see if there is any relation between planet type and mass of the planet. It looks like there is but let's see.

Let's code a machine learning model and plot it accordingly. Let's first plot all the planets with the **planet_mass** and **planet_radius**.

We'll first collect all the planet **mass** and **radius** and store them in a list. Then we are plotting a scatter plot just like how we did earlier, but this time by only providing the X and the Y coordinates.

```
planet_masses = []
planet_radiuses = []
for planet_data in low_gravity_planets:
  planet_masses.append(planet_data[3])
  planet_radiuses.append(planet_data[7])

fig = px.scatter(x=planet_radiuses, y=planet_masses)
fig.show()
```

Using the **scatter()** method we got the following plot.



With the following result, we don't know if there exist any clusters in this, and if yes, then how many such clusters exist? Let's find out the K for this cluster. Do you remember what K is in clustering and how do we find it out? To choose the right K, we use the **WCSS perimeter** to evaluate the choice of K. It stands for **Within Cluster Sum of Squares**. This means that we will be choosing the center point of the cluster from where all the points falling inside that cluster will be closest. We need to build a model to find the value of K. Great!	**ESR:** K signifies the number of clusters that the algorithm finds in the dataset.

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.	
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 • **To build the clustering model** • **To plot graph** • **To filter out more data**	
Teacher Action	**Student Action**
Okay, let's start by building a model for clustering. For this, we will have to merge our two lists into 1 as a list of lists. Then, we will find out the WCSS for our clusters. Open Student Activity 1 for the boilerplate code.	

Note: *Guide the student to open boilerplate code and run all the cells. It is the same code we finished in our last class.* This is the code we have written earlier in the clustering class, but let's again go through it line by line: 1. Import the KMeans from the **sklearn** library. sklearn stand for **Scikit Learn,** it is used to perform various Machine Learning algorithms like classification, regression, clustering, etc. [Teacher Activity 2: Scikit Learn] 2. import **matplotlib** and **seaborn** libraries to prettify our plots. 3. Then, create an empty list **X** which will contain multiple lists for all the planets, having their mass and radius. We are doing the same in the first **for** loop where we are using the **enumerate()** function to get the index as well. Then, we are creating another empty list **wcss** which will contain all the WCSS values.	

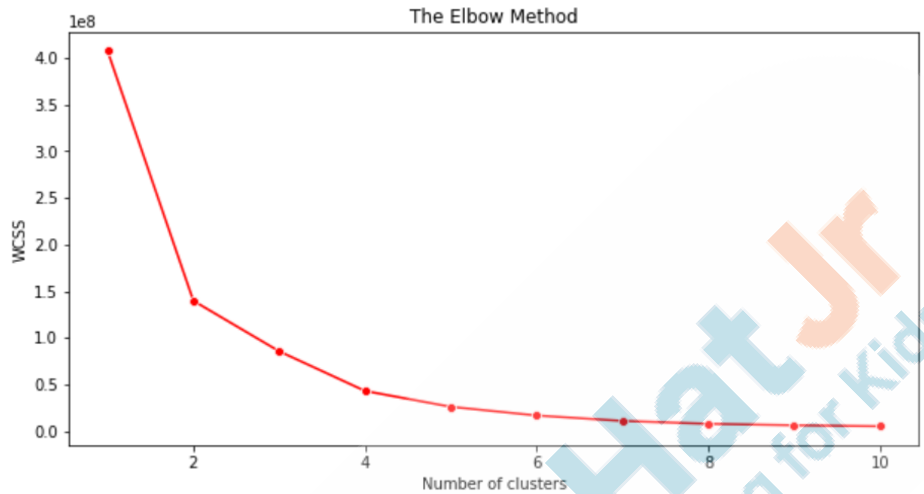
```python
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt
import seaborn as sns

X = []
for index, planet_mass in enumerate(planet_masses):
  temp_list = [
                planet_radiuses[index],
                planet_mass
            ]
  X.append(temp_list)

wcss = []
```

4. Iterate 10 times (Since we expect our number of clusters to lie anywhere between 1 to 10). 5. Prepare the **K-means** model where we are giving it the number of clusters, asking it to use the **k-means++** method and giving a random state of 42. **k-means++** is just the name of the algorithm that we want to use. It is the most famously used algorithm. 6. Fit our **X** in our model then append the **wcss** of this into the wcss list. (**inertia_ returns** the value of wcss for a KMeans model) 7. Finally, plot the WCSS chart. We are giving the figure size to be 10 units wide and 5 units long. Do you remember what we call this method?	**ESR:** Elbow Method

```
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt
import seaborn as sns

X = []
for index, planet_mass in enumerate(planet_masses):
  temp_list = [
                  planet_radiuses[index],
                  planet_mass
              ]
  X.append(temp_list)

wcss = []
for i in range(1, 11):
    kmeans = KMeans(n_clusters=i, init='k-means++', random_state = 42)
    kmeans.fit(X)
    # inertia method returns wcss for that model
    wcss.append(kmeans.inertia_)

plt.figure(figsize=(10,5))
sns.lineplot(range(1, 11), wcss, marker='o', color='red')
plt.title('The Elbow Method')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS')
plt.show()
```

Run this to get the line plot as follows.	

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning:

Pass the following variables as keyword args: x, y. From version 0.12, the only
```



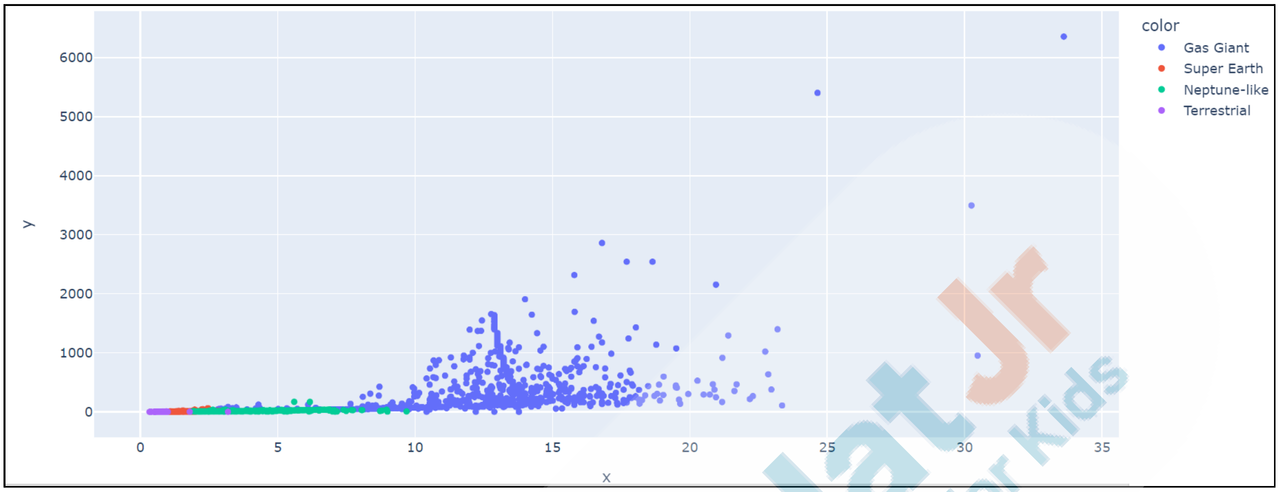
Awesome! Here, we can see a decent up until 4, after which it is not significant.
When we found the number of unique planet types earlier, that came out to be 4 as well but when we looked at the scatter plot, we couldn't detect the number of clear clusters.

That's the power of machine learning!

Let's plot the previous chart, this time with the color-coding based on the planet type.

```
planet_masses = []
planet_radiuses = []
planet_types = []
for planet_data in low_gravity_planets:
  planet_masses.append(planet_data[3])
  planet_radiuses.append(planet_data[7])
  planet_types.append(planet_data[6])

fig = px.scatter(x=planet_radiuses, y=planet_masses, color=planet_types)
fig.show()
```

Run this to get the scatter plot as follows.



Okay, and now the big question. Out of the 4 types of planets that we studied, which are the ones that can support life?

Okay, then from our **low_gravity_planets** list, let's filter out planets with planet type as **Neptune-like & Gas Giant**. Here, we are creating a new list where we are checking all the low gravity planets. Also, if they are **terrestrial** or **super earth-like**, we are adding them to a new list.

ESR:
~ Terrestrial
~ Super-Earth

```
suitable_planets = []
for planet_data in low_gravity_planets:
  if planet_data[6].lower() == "terrestrial" or planet_data[6].lower() == "super earth":
    suitable_planets.append(planet_data)

print(len(suitable_planets))

1452
```

Great! We are still left with 1,452 planets that still support us!

But, were these the only factors that we needed in order to survive? What else do you think we need to survive on a

ESR:
Varied.

planet that we need from nature? So, in this class, we learned about different types of planets, and also, we witnessed how Data Science can help us identify things that we otherwise cannot do by ourselves! How was your experience?	**ESR:** Varied.

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>

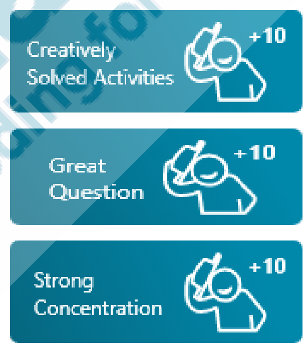
Activity Details

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! Amazing. While working on this project, we also made sure that we are on top of all the concepts we have acquired so far. Next class, we will filter out more planets based on various other factors. We will also learn new concepts and techniques!	*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

ACTIVITY LINKS		
Activity Name	**Description**	**Links**
Teacher Activity 1	Boilerplate Code	https://colab.research.google.com/drive/1pCx_fZJ3vJDcwYsnG7zDdfAgZmfv0MQ8?usp=sharing
Teacher Activity 2	Scikit learn	https://scikit-learn.org/
Teacher Activity 3	Reference Code	https://colab.research.google.com/drive/1Kgr0Fxy-F6P4QFcQzYuQIoC5_oqUfNl0?usp=sharing
Teacher Reference 1	Project	https://s3-whjr-curriculum-uploads.whjr.online/2348b3e8-6e88-4abf-8f61-a346bbbf6142.pdf
Teacher Reference 2	Project Solution	https://colab.research.google.com/drive/1BP8UuSBf8UlxA4eFq8MhxOm9rdxeyDcb?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/6c09597a-d3a0-4e17-87c1-1488ba7c91e7.pdf
Student Activity 1	Boilerplate Code	https://colab.research.google.com/drive/1jy8aGn9I7WoHPQVBzwwZhPYUM9pydDLc?usp=sharing