Reading a CSV data file

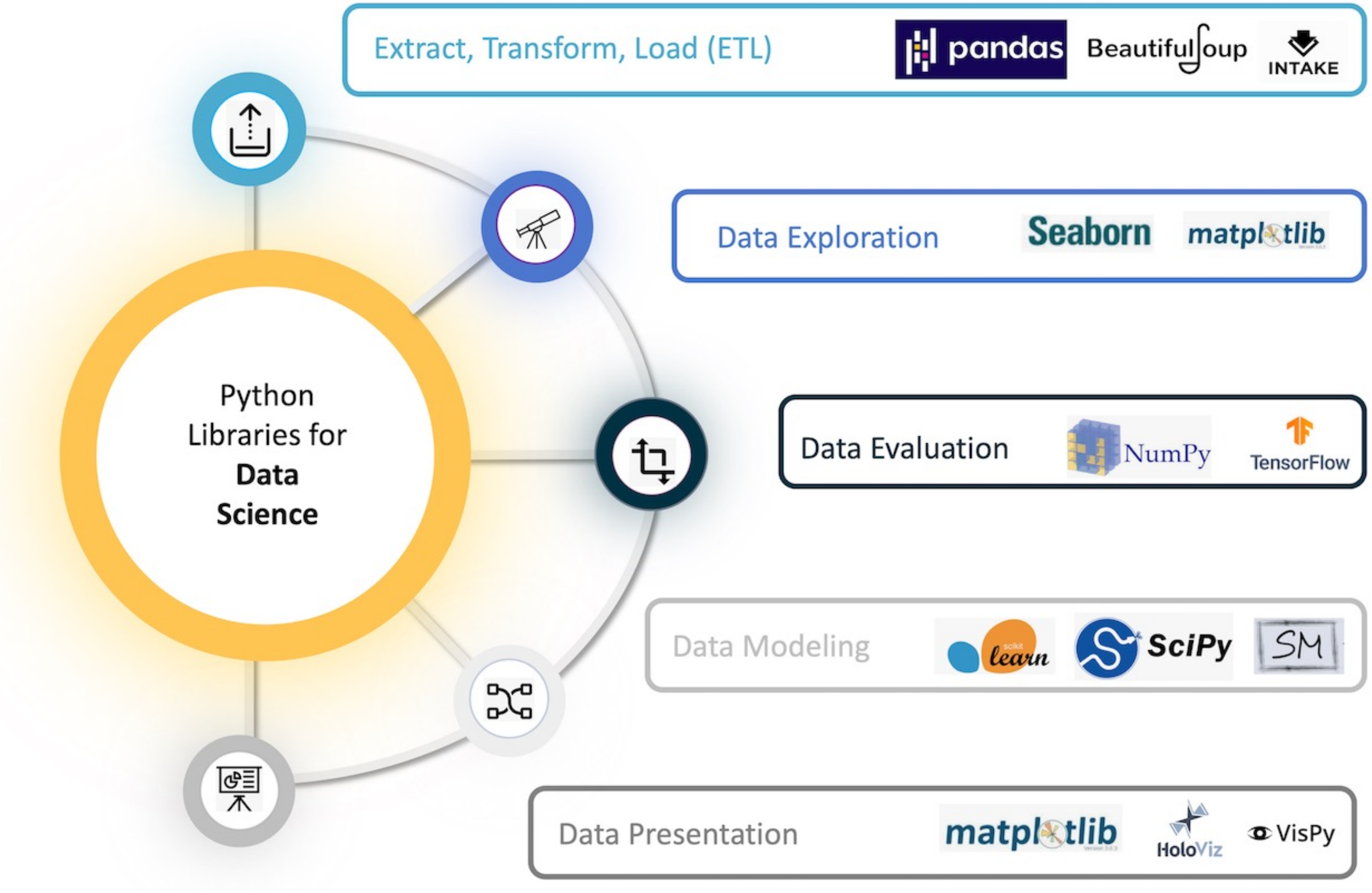
Objective

- Become familiar with libraries in python
- Become familiar with pandas library and dataframes
- Read a csv file
- Display a csv file

Library

- Collection of predefined code that can be used for specific purpose
- Eliminates the need to write the code from scratch.
- Built by developers and are made available for others to use.
- Used to make frequently used tasks more efficient.
- Different types of Libraries in python:
    - **Statsmodels**: For different statistical analysis
    - **Pandas**: working with csv
    - **Math**: for different mathematical operation
- To Use a library you can simply **import** a library (after installing the library)

Libraries in Python for Data Science



Python Libraries for **Data Science**

Extract, Transform, Load (ETL) — pandas, BeautifulSoup, INTAKE

Data Exploration — Seaborn, matplotlib

Data Evaluation — NumPy, TensorFlow

Data Modeling — scikit-learn, SciPy, SM

Data Presentation — matplotlib, HoloViz, VisPy

Python has a vast ecosystem of libraries that cover a wide range of domains, such as

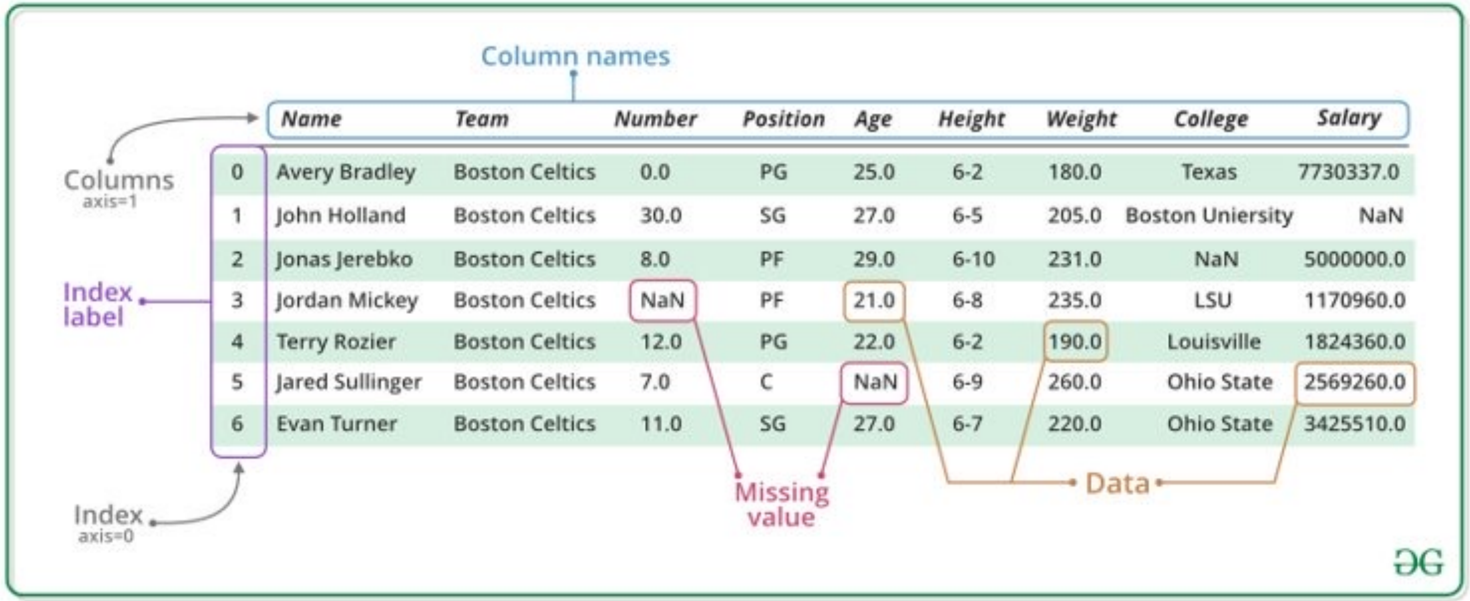
- data analysis,
- web development,
- machine learning, and more.

Pandas

- Powerful and widely used library in Python for data manipulation and analysis.
- Provides high-performance data structures, such as the DataFrame, that allows you to work with structured data effectively.

Dataframes:

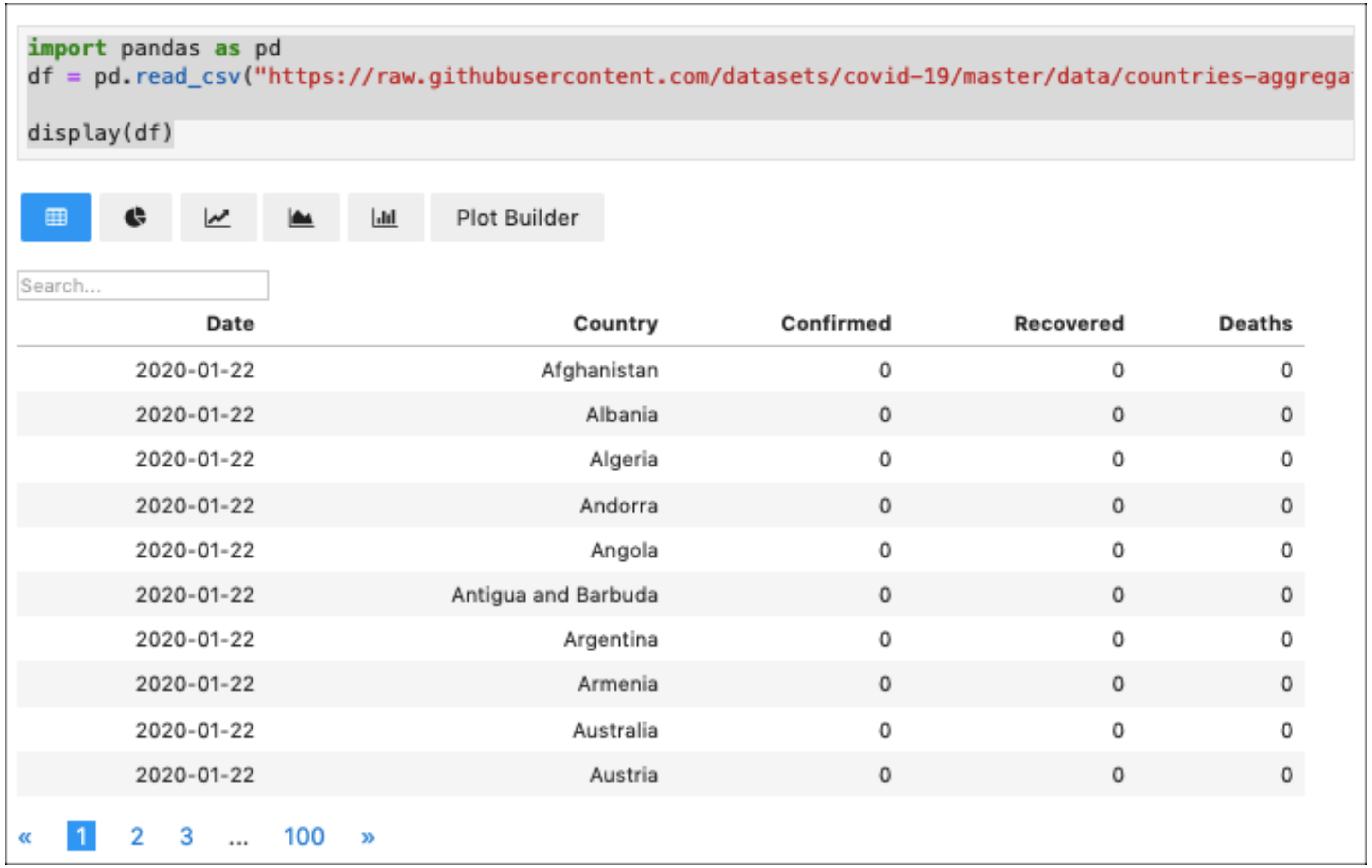
- Two-dimensional tabular data structure, similar to a table in a spreadsheet or a SQL database.
- Consists of rows and columns
- Each column can have a different data type (e.g., numbers, strings, dates, etc.).



	Name	Team	Number	Position	Age	Height	Weight	College	Salary
0	Avery Bradley	Boston Celtics	0.0	PG	25.0	6-2	180.0	Texas	7730337.0
1	John Holland	Boston Celtics	30.0	SG	27.0	6-5	205.0	Boston Uniersity	NaN
2	Jonas Jerebko	Boston Celtics	8.0	PF	29.0	6-10	231.0	NaN	5000000.0
3	Jordan Mickey	Boston Celtics	NaN	PF	21.0	6-8	235.0	LSU	1170960.0
4	Terry Rozier	Boston Celtics	12.0	PG	22.0	6-2	190.0	Louisville	1824360.0
5	Jared Sullinger	Boston Celtics	7.0	C	NaN	6-9	260.0	Ohio State	2569260.0
6	Evan Turner	Boston Celtics	11.0	SG	27.0	6-7	220.0	Ohio State	3425510.0

Dataframes:

- In Python Notebooks, dataframes are commonly used to work with the CSVs
- Dataframes provides various functionalities for indexing, selecting, filtering, transforming, plotting, and analyzing data
- Usually effective in reusing the same data for different tasks

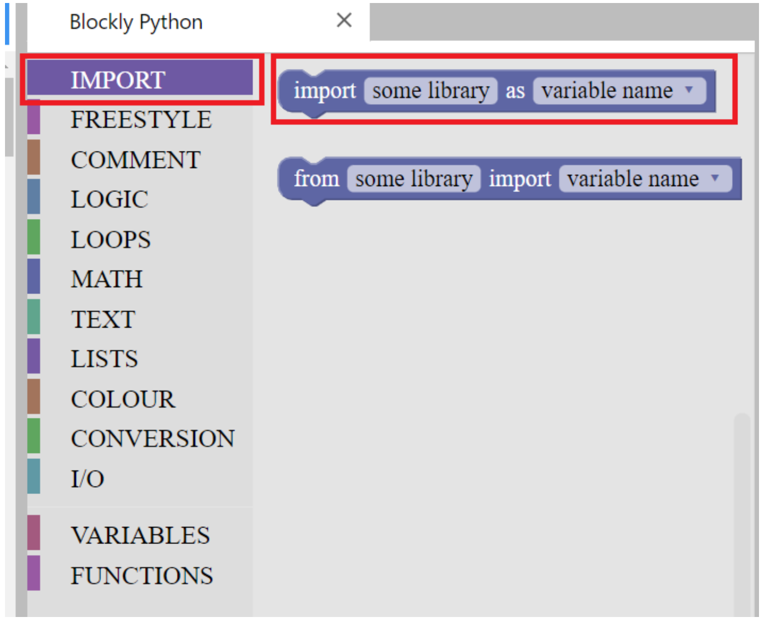


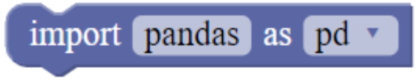
```
import pandas as pd
df = pd.read_csv("https://raw.githubusercontent.com/datasets/covid-19/master/data/countries-aggrega

display(df)
```

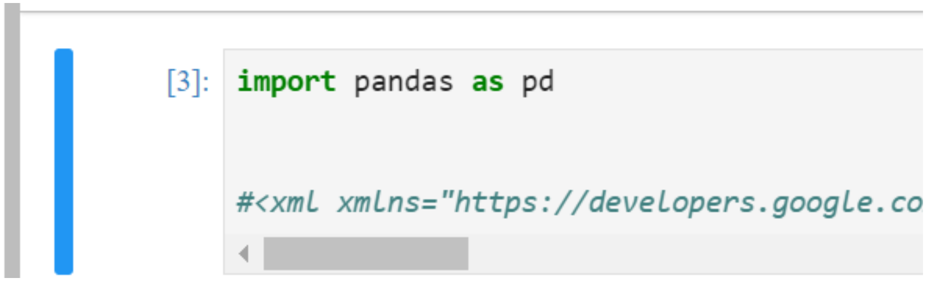
Date	Country	Confirmed	Recovered	Deaths
2020-01-22	Afghanistan	0	0	0
2020-01-22	Albania	0	0	0
2020-01-22	Algeria	0	0	0
2020-01-22	Andorra	0	0	0
2020-01-22	Angola	0	0	0
2020-01-22	Antigua and Barbuda	0	0	0
2020-01-22	Argentina	0	0	0
2020-01-22	Armenia	0	0	0
2020-01-22	Australia	0	0	0
2020-01-22	Austria	0	0	0

Read a dataset using pandas:

Importing pandas using Blockly :







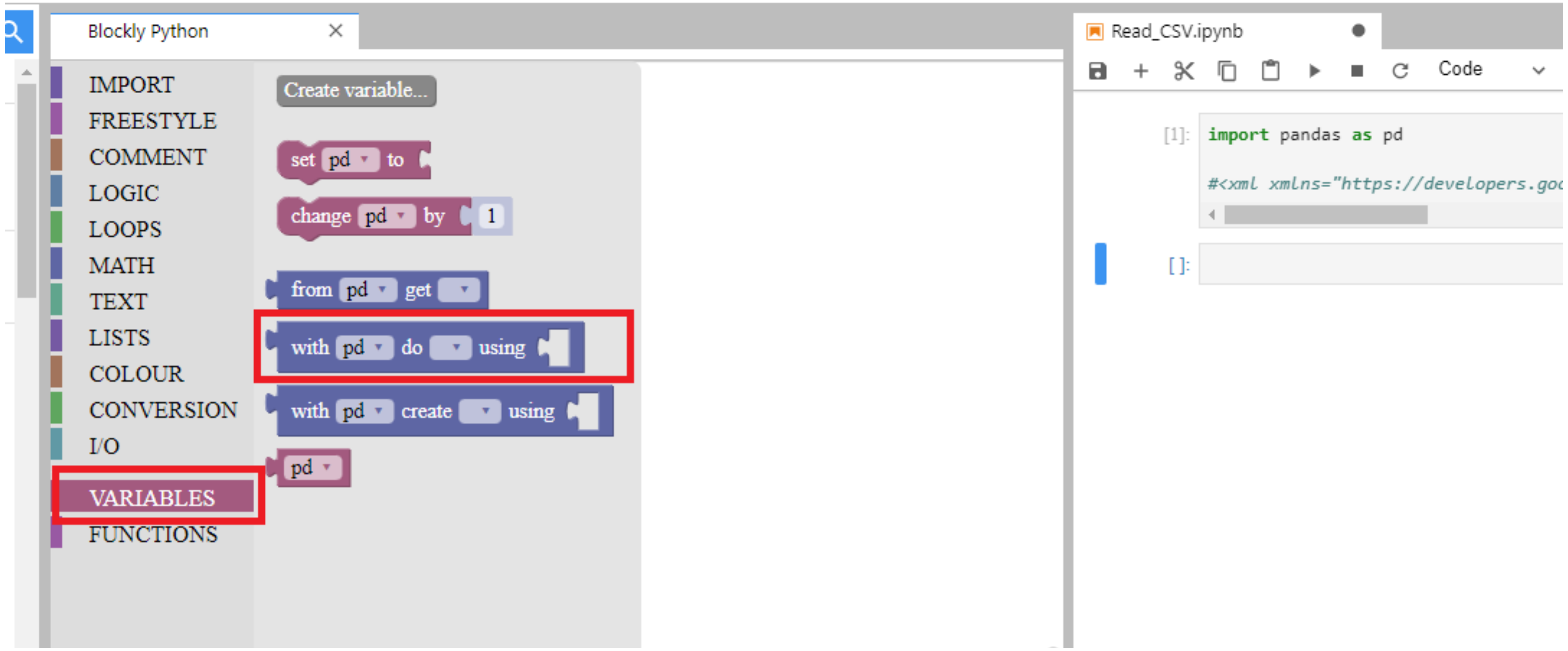
```
[3]: import pandas as pd


     #<xml xmlns="https://developers.google.co
```

- To work with pandas, you can import it in Python using the *import* statement.
- We import pandas and give it the alias **pd**, which is a common convention used by the **pandas** community.
- The alias is then used while using its functions instead of the **pandas** name.

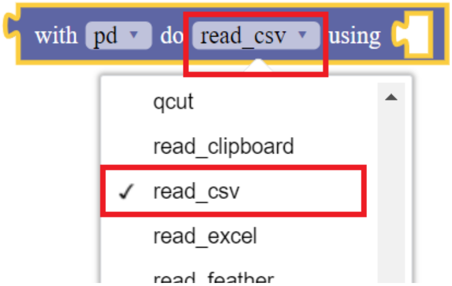
Steps:

- Go to *import*
- Import pandas as pd

Using pandas to read the CSV data...



```
import pandas as pd

#<xml xmlns="https://developers.goo
```

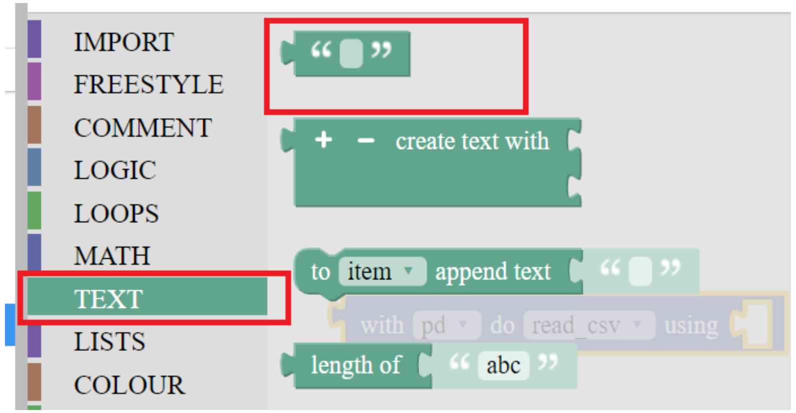


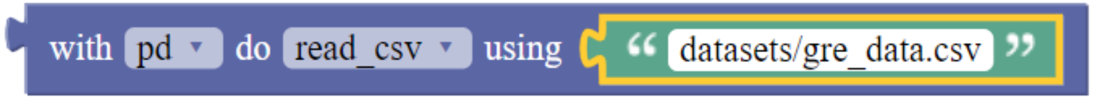
- We will now read a csv file **gre_data** with *GPA, gender* and *GRE score* information using pandas.
- To read a csv, we will use the pandas library we've already imported
- We use the *read_csv* method in pandas which allows us to read a csv file

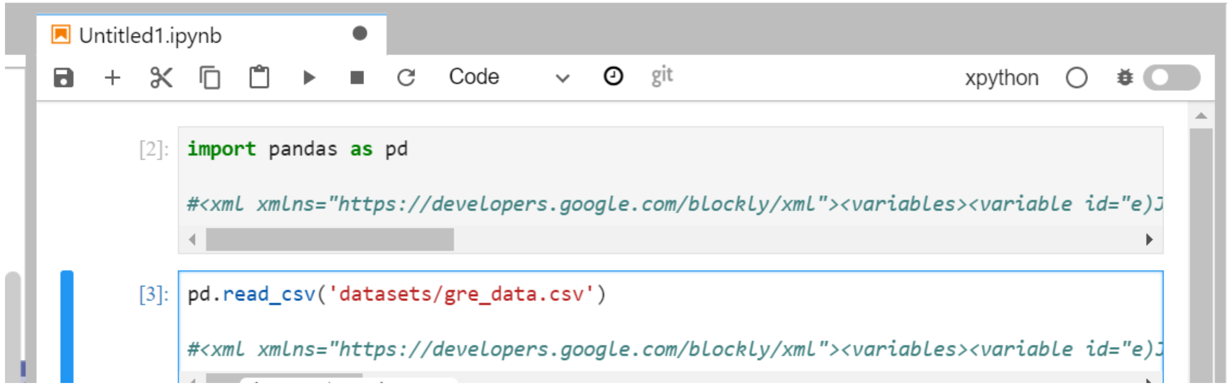
Steps:

- Goto *Variables*
- Create a variable pd
- Select the option "*with pd do...*"

Using pandas to read the CSV data







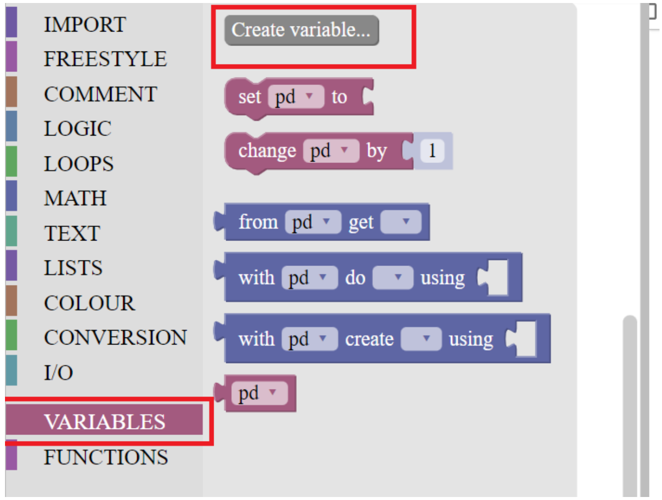
```
[2]: import pandas as pd

     #<xml xmlns="https://developers.google.com/blockly/xml"><variables><variable id="e)J
```

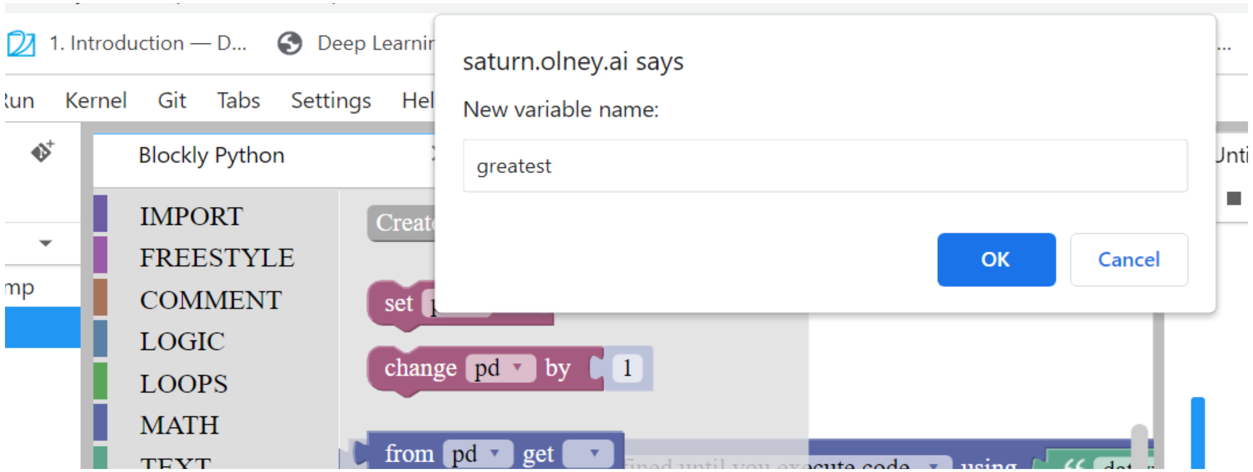
```
[3]: pd.read_csv('datasets/gre_data.csv')

     #<xml xmlns="https://developers.google.com/blockly/xml"><variables><variable id="e)J
```

Steps continued...

- Go to *Variable*
- Select the option "*with pd do...*"
- Select the option *read_csv* (Make sure to execute the import pd first)
- Go to *text* option
- Set the text as the *path to your csv file*
- Put the text block with filepath inside the "*using ...*" block
- Convert blocks to code
- Execute the cell

Create the dataframe:





saturn.olney.ai says

New variable name:

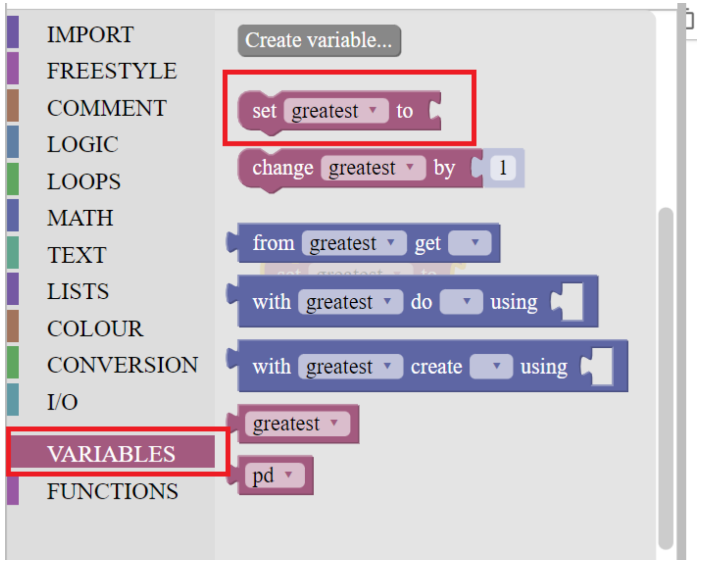
greatest

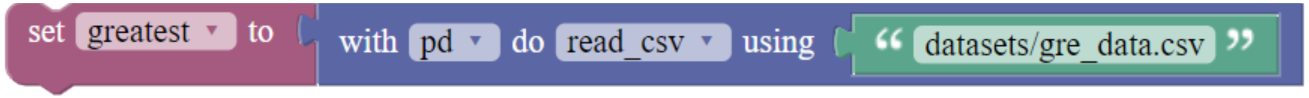
- We will now store the read data into a dataframe so that we can refer to the dataframe for data manipulation
- We will name the dataframe ***greatest***

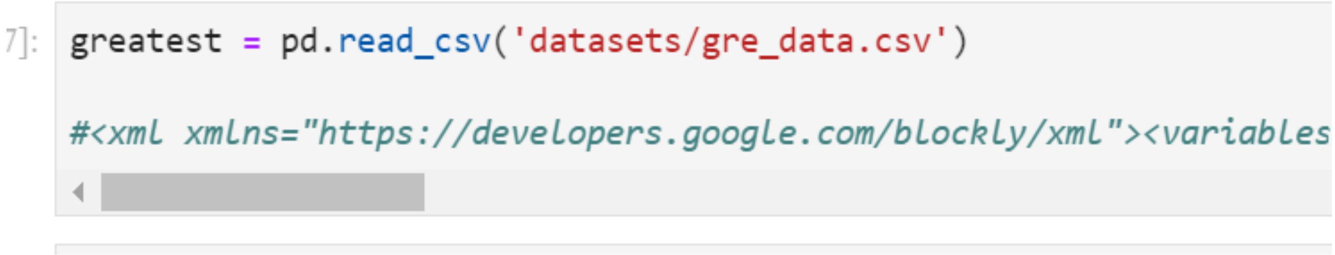
Steps:

- Go to *Variables*
- Create a new Variable as **greatest**

Set the dataframe to read data:





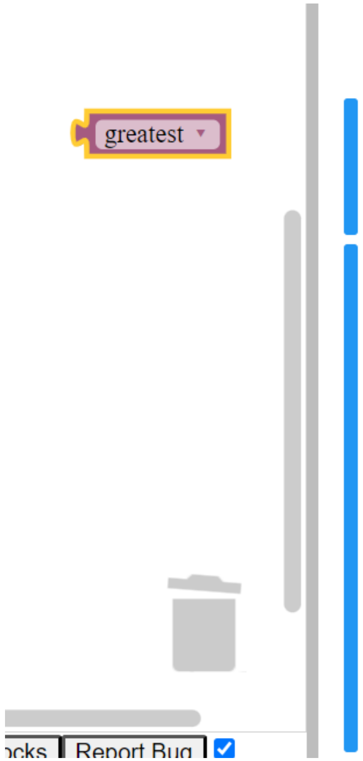


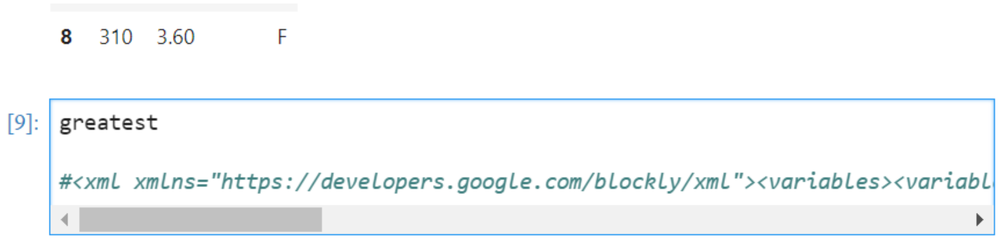
```
7]: greatest = pd.read_csv('datasets/gre_data.csv')

#<xml xmlns="https://developers.google.com/blockly/xml"><variables
```

Steps:

- Go to *Variables* and select "*set greatest to ..*" option
- Set the variable *greatest* to the previously read data by combining the two blocks as shown in the fig.
- Use **blocks to code** and execute the code cell

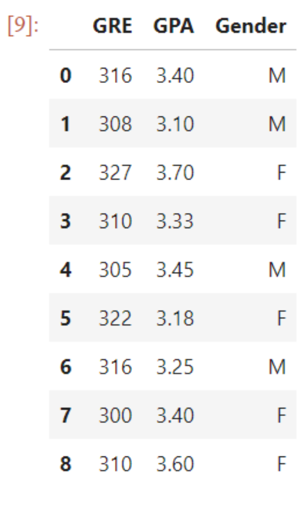
View the dataframe:





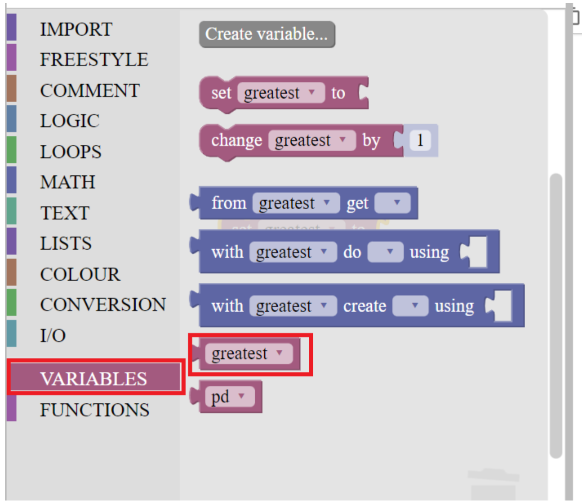
```
8   310   3.60   F
```

```
[9]: greatest

     #<xml xmlns="https://developers.google.com/blockly/xml"><variables><variabl
```



[9]:

	GRE	GPA	Gender
0	316	3.40	M
1	308	3.10	M
2	327	3.70	F
3	310	3.33	F
4	305	3.45	M
5	322	3.18	F
6	316	3.25	M
7	300	3.40	F
8	310	3.60	F



- We will now print/view the dataframe **greatest** which stored our data i.e. the tabular data containing GRE, GPA and Gender

Steps:

- Go to *Variables*
- Select *greatest*
- Convert blocks to code and execute the code cell