Exploring Dataset

How We Visualize the Dataset and Know More

Catch Up

- We've
  - Learnt about CSV file
  - Learnt about Dataframe
  - Learnt about Libraries
  - Imported the CSV into a dataframe

Q & A: Do you remember which library we used to import csv into a dataframe?

Catch Up

- We've
  - Learnt about CSV file
  - Learnt about Dataframe
  - Learnt about Libraries
  - Imported the CSV into a dataframe

Q & A: Do you remember which library we used to import csv into a dataframe?

pandas

Exploring Datasets

- We’ve imported the CSV into a dataframe.
- Now what?

- Now that we have stored our data in the dataframe, let's get some more information from it.
- Each row represents an observation (an instance)
- Each column represents a characteristic (feature) of each observation.

Exploring Datasets

- We can see how many rows are present in the dataset.
- Each row represents an observation (an instance). Here each row represents a student

Q & A: How many rows are shown in the CSV?

```
GRE,GPA,Gender
316,3.4,M
308,3.1,M
327,3.7,F
310,3.33,F
305,3.45,M
322,3.18,F
316,3.25,M
300,3.4,F
310,3.6,F
```

Exploring Datasets

- We can see how many rows are present in the dataset.

Q & A: How many rows in the CSV shown?

9

```
GRE,GPA,Gender
316,3.4,M
308,3.1,M
327,3.7,F
310,3.33,F
305,3.45,M
322,3.18,F
316,3.25,M
300,3.4,F
310,3.6,F
```

Exploring Datasets

- Usually, before starting working in a dataset, we need to know the features (columns) in it.
- Each column represents a characteristic (feature) of each observation
- We can see how many columns are present in the dataset.

Q & A: What are the columns shown in the CSV?

```
GRE,GPA,Gender
316,3.4,M
308,3.1,M
327,3.7,F
310,3.33,F
305,3.45,M
322,3.18,F
316,3.25,M
300,3.4,F
310,3.6,F
```

Exploring Datasets

- We can see how many columns are present in the dataset.

Q & A: What are the columns in the CSV shown?

GRE, GPA, Gender

```
GRE,GPA,Gender
316,3.4,M
308,3.1,M
327,3.7,F
310,3.33,F
305,3.45,M
322,3.18,F
316,3.25,M
300,3.4,F
310,3.6,F
```

Know the Dataset

- How to know how many columns and rows are there in a dataset?
- To know the dataset we have to first read the csv data and store the content into a dataframe

Know the Dataset

Read the CSV data

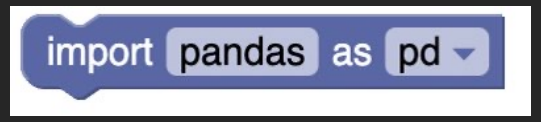
First, we create a dataframe and store the content of the dataset into the dataframe

- Import pandas Library

  Python code:

```python
import pandas as pd
```

  blockly block:



- Read CSV data and Save in Variable
  - Python code:

```python
gredata = pd.read_csv('datasets/gre_data.csv')
```

  - Blockly block:

To read from a csv file, first we will import pandas library because it has a read_csv function which we will use to automatically parse the csv file and load it into the notebook.

The read_csv function requires us to supply the relative path to the csv file and returns a Pandas Dataframe object which we will store in a variable so we can use it later in the notebook.

Know the Dataset

Displaying the dataframe content:

By displaying the dataset content, we can know about the columns and rows.

- In previous slide, we've already stored the csv data in "gredata" dataframe
- Now we display dataframe content

Python code:

```
gredata
```

Blockly code:

Calling the dataframe variable in a cell by itself will print the contents of the dataframe to the screen so we can confirm the dataset was imported correctly.

	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

Know the Dataset

shape of the dataset

Sometimes, we need to know the size of the dataset for analysis convenience. And by size, we mean the number of rows and columns in the dataset. We can know the size of the dataset (rows and columns) by using the `shape` function.

- Use shape method

  Python code:

```
gredata.shape
```

Syntax: dataframe_variable_name.shape

  Blockly block:

Output

```
(9, 3)
```

	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

Crosscheck with our csv data

Know the Dataset

- Datasets usually come out of research studies and most often they are Very Large.
- For large datasets we often can't see all the rows or columns in notebook

BsmtFinSF2	BsmtUnfSF	TotalBsmtSF	Heating	...	CentralAir	Electrical	1stFlrSF	2ndFlrSF	LowQualFinSF	GrLivArea	BsmtFullBat
0	150	856	GasA	...	Y	SBrkr	856	854	0	1710	
0	284	1262	GasA	...	Y	SBrkr	1262	0	0	1262	
0	434	920	GasA	...	Y	SBrkr	920	866	0	1786	
0	540	756	GasA	...	Y	SBrkr	961	756	0	1717	
0	490	1145	GasA	...	Y	SBrkr	1145	1053	0	2198	
...	...	...	..	...	...	...	...	...	...	...	.
0	953	953	GasA	...	Y	SBrkr	953	694	0	1647	
163	589	1542	GasA	...	Y	SBrkr	2073	0	0	2073	
0	877	1152	GasA	...	Y	SBrkr	1188	1152	0	2340	
1029	0	1078	GasA	...	Y	FuseA	1078	0	0	1078	
290	136	1256	GasA	...	Y	SBrkr	1256	0	0	1256	

Know the Dataset

Column names of the dataset

We can know the column names of the dataset by using the `columns` function.

- Use columns function

Python code:

```
gredata.columns
```

Syntax: dataframe_variable_name.columns

Blockly block:

Output

```
Index(['GRE', 'GPA', 'Gender'], dtype='object')
```

	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

Crosscheck with our csv data

Select Any Specific Data in Dataset

- We can use the dataframe data mainly in 2 ways
  - Row wise
  - Column wise

- Each row can be defined as datapoint
- Each column can be defined as a feature/attribute of each datapoint

Select Any Specific Data in Dataset

- In our gre_data.csv dataset each row represents a student
- And each column represents a student's various attributes/features
  - except the first one with no column name – known as index number

	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

Select Any Rows in Dataset

Index:

- Each row is assigned with an index number
- Usually starts from 0 and increase by 1 for each additional row
- Index number of nth row is n-1

Q&A: What will be the index number of 5th row?

	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

Select Any Rows in Dataset

Index:

- Each row is assigned with an index number
- Usually starts from 0 and increase by 1 for each additional row
- Index number of nth row is n-1

Q&A: What will be the index number of 5th row?

4

	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

Select Any Rows in Dataset

Select and view first x rows from our gre_data.csv dataset?

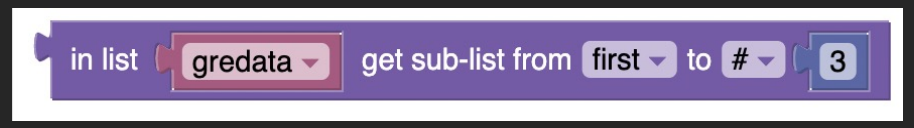
To accomplish this we can use the index number. The syntax for selecting 1st x rows is: "dataframe_variable_name[:x]"

- Suppose we want see 1st 3 rows of the dataset. We have to use “dataframe_variable_name[:3]" :

Python code:

```
gredata[ : 3]
```

Blockly block



Output

	GRE	GPA	Gender
0	316	3.4	M
1	308	3.1	M
2	327	3.7	F

	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

Crosscheck with our csv data

Select Any Column in Dataset

Selecting columns is a bit different from selecting rows.

Unlike rows, dataset columns have their own names.

The values under a column are presented as a list. So, we can think of a column as a list of values.

The syntax for selecting any specific column is:
"dataframe_variable_name[['column_name']]"

Select Any Column in Dataset

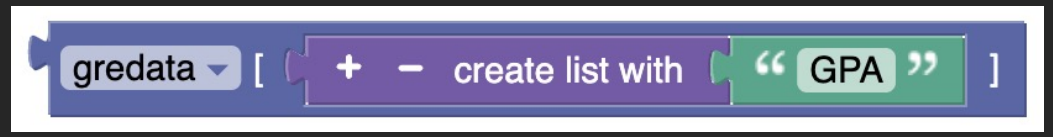
The syntax for selecting any specific column is: "dataframe_variable_name[['column_name']]"

- Suppose, we want to see "GPA" column of our gre dataset. We can use gredata[['GPA']]

Python code:

```
gredata[['GPA']]
```

Blockly block



Output

	GPA
0	3.40
1	3.10
2	3.70
3	3.33
4	3.45
5	3.18
6	3.25
7	3.40
8	3.60

Select Multiple Columns in Dataset

Just like selecting a single colum, we can also select multiple columns from a datframe.

For selecting multiple columns, we follow the same process as selecting a single column.

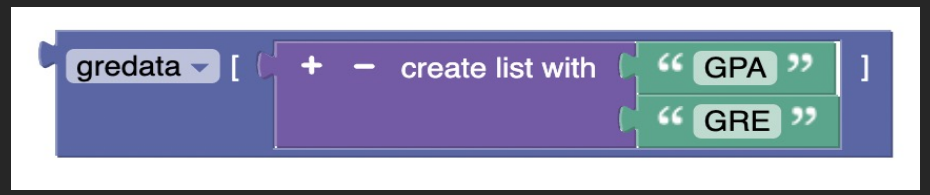
The syntax for selecting multple specific column is: "dataframe_variable_name[['column_1_name'],['column_2_name'],..., ['column-n_name ']]"

- Suppose, we want to see "GPA" and "GRE" column of our gre dataset. We can use gredata[['GPA'],['GRE']]

Python code:

```
gredata[['GPA', 'GRE']]
```

Blockly block



Output

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

Reference Notebook:

- read_csv_2_ex.ipynb