

Introduction to Data Science

Reading Data and Data Types

Joanna Bieri DATA101

Reading in Your Own Data

ASK

You run `pd.read_csv('mapdataaall.csv')`
and get `FileNotFoundError`. What went wrong?

Reading in Your Own Data

ANSWER

- The file is not in the same folder as your notebook (it is probably still in Downloads).
- The name does not match exactly. A second download is often called `mapdataa11 (1).csv`.
- Fix: move the file into your Day8 folder, or rename it, then run the cell again.

Data Types

PREDICT

value
'1'
1
1.0
True

`type(value)`

What type is each one?

Data Types

ANSWER

value	type
'1'	str
1	int
1.0	float
True	bool

The quotes are what make '1' a string. It looks like a number, but Python treats it like a word.

Changing Data Types

PREDICT

```
int('7')  
int(4.9)  
float('1.2')
```

What comes out of each line?

Changing Data Types

ANSWER

```
int('7')      -> 7  
int(4.9)      -> 4  
float('1.2')  -> 1.2
```

`int(4.9)` is 4, not 5. `int()` chops off the decimal part, it does not round.

Fun with Strings

PREDICT

```
year = '1931/32'  
year.split('/')
```

What comes out?

Fun with Strings

ANSWER

```
year.split('/')
```

```
['1931', '32']
```

A list of two strings. `[0]` would give you `'1931'`, and it is still a string.

What Type is my Column?

PREDICT

DF

	Film	Year	Awards
0	Wings	1927	2
1	Cavalcade	1932/33	3
2	Rebecca	1940	2

DF.dtypes

What type does pandas give each column?

What Type is my Column?

ANSWER

`DF.dtypes`

column	dtype
Film	object
Year	object
Awards	int64

One value like 1932/33 is enough to make the whole Year column object (strings).

Help my Data is Broken!

ASK

`DF['Awards'].mean()` gives the error
Could not convert string to numeric.
What is it telling you?

Help my Data is Broken!

ANSWER

- The Awards column has strings in it, not numbers.
- Python tried to turn them into numbers and could not.
- First step: look at the column with `.dtypes` and `.value_counts()` to find the weird values.

.apply() and lambda

PREDICT

DF

	Year
0	1927
1	1932/33
2	1940

```
DF['Year'].apply(lambda x: x.split('/')[0])
```

What comes out? Is it a number yet?

.apply() and lambda

ANSWER

```
DF['Year'].apply(lambda x: x.split('/')[0])
```

	Year
0	1927
1	1932
2	1940

The /33 is gone, but the dtype is still object. These are strings until you use `int()`.

Fixing a Column

REVERSE

DF

	Awards
0	3
1	0 (1)
2	2 (1)

want

	Awards
0	3
1	0
2	2

The answer should be whole numbers (int64). What is the command?

Fixing a Column

ANSWER

```
DF['Awards'].apply(lambda x: int(x.split('(')[0]))
```

	Awards
0	3
1	0
2	2

Split at the (, keep the part on the left with [0], then int() turns it into a number.

Questions?

Questions?