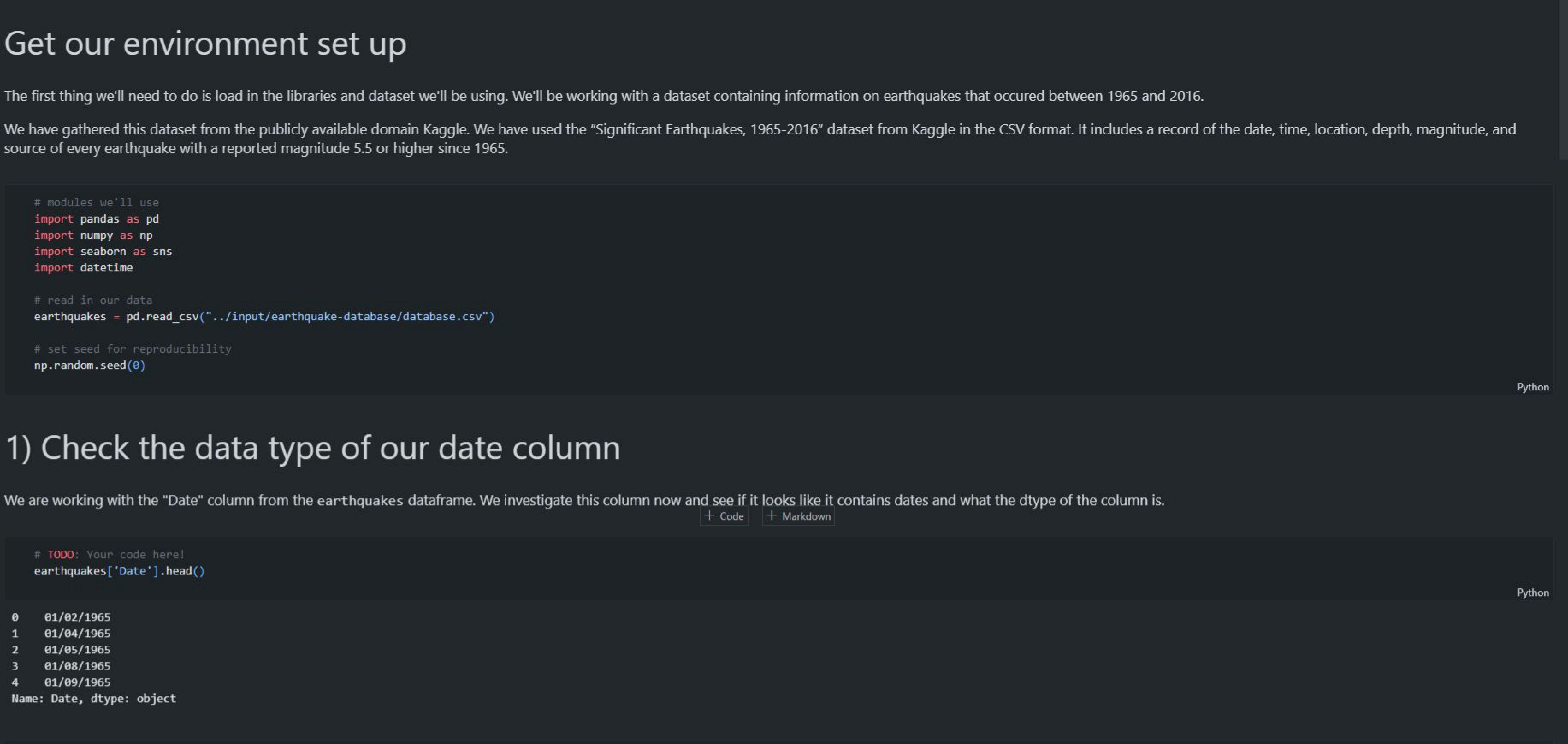


Get our environment set up

The first thing we'll need to do is load in the libraries and dataset we'll be using. We'll be working with a dataset containing information on earthquakes that occured between 1965 and 2016.

We have gathered this dataset from the publicly available domain Kaggle. We have used the “Significant Earthquakes, 1965-2016” dataset from Kaggle in the CSV format. It includes a record of the date, time, location, depth, magnitude, and source of every earthquake with a reported magnitude 5.5 or higher since 1965.

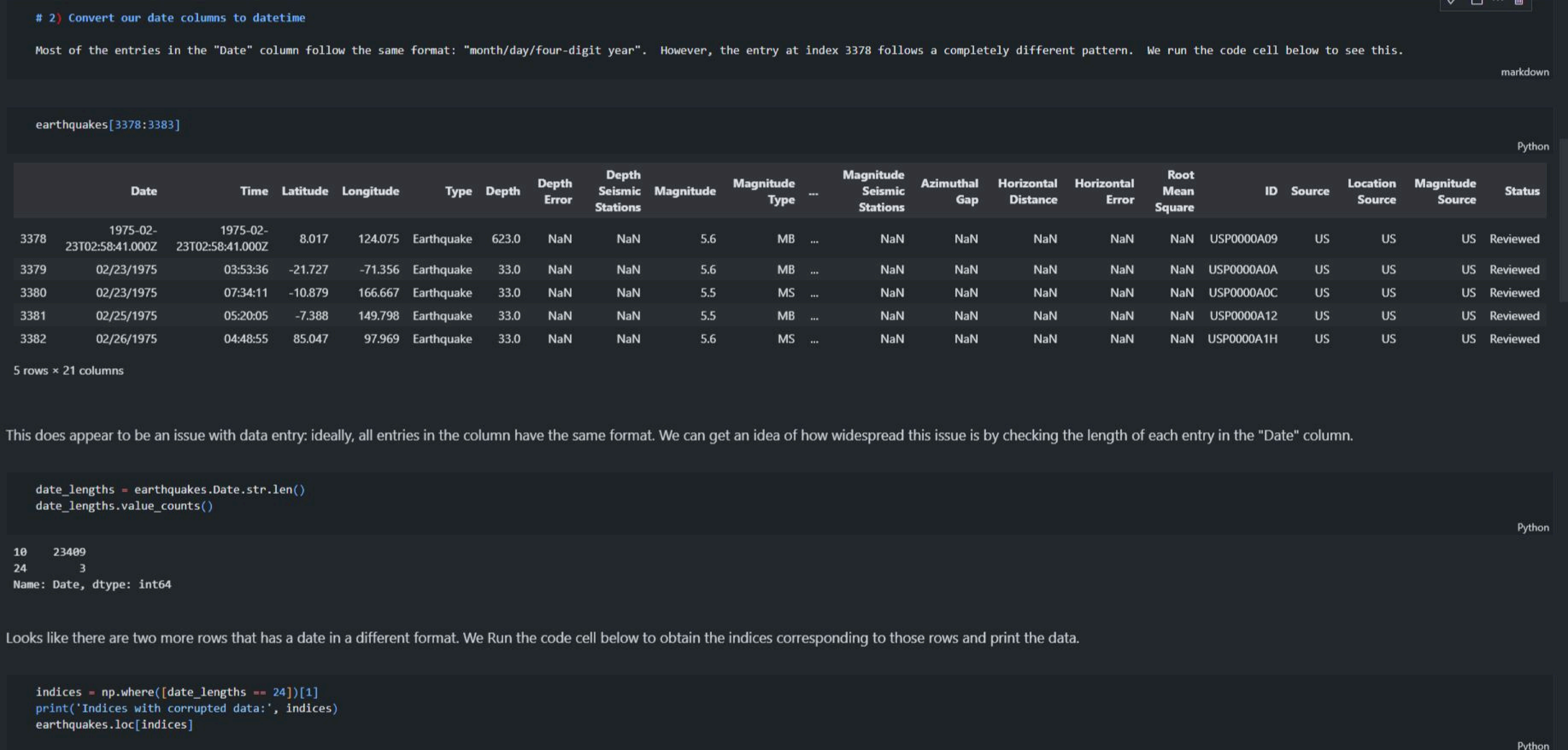
```python
# modules we'll use
import pandas as pd
import numpy as np
import seaborn as sns
import datetime

# read in our data
earthquakes = pd.read_csv("../input/earthquake-database/database.csv")

# set seed for reproducibility
np.random.seed(0)
```

1) Check the data type of our date column

We are working with the "Date" column from the `earthquakes` dataframe. We investigate this column now and see if it looks like it contains dates and what the dtype of the column is.

```python
# TODO: Your code here!
earthquakes['Date'].head()
```

```
0    01/02/1965
1    01/04/1965
2    01/05/1965
3    01/08/1965
4    01/09/1965
Name: Date, dtype: object
```



```
# 2) Convert our date columns to datetime

Most of the entries in the "Date" column follow the same format: "month/day/four-digit year".  However, the entry at index 3378 follows a completely different pattern.  We run the code cell below to see this.
```

```
earthquakes[3378:3383]
```

	Date	Time	Latitude	Longitude	Type	Depth	Depth Error	Depth Seismic Stations	Magnitude	Magnitude Type	...	Magnitude Seismic Stations	Azimuthal Gap	Horizontal Distance	Horizontal Error	Root Mean Square	ID	Source	Location Source	Magnitude Source	Status
3378	1975-02-23T02:58:41.000Z	1975-02-23T02:58:41.000Z	8.017	124.075	Earthquake	623.0	NaN	NaN	5.6	MB	...	NaN	NaN	NaN	NaN	NaN	USP0000A09	US	US	US	Reviewed
3379	02/23/1975	03:53:36	-21.727	-71.356	Earthquake	33.0	NaN	NaN	5.6	MB	...	NaN	NaN	NaN	NaN	NaN	USP0000A0A	US	US	US	Reviewed
3380	02/23/1975	07:34:11	-10.879	166.667	Earthquake	33.0	NaN	NaN	5.5	MS	...	NaN	NaN	NaN	NaN	NaN	USP0000A0C	US	US	US	Reviewed
3381	02/25/1975	05:20:05	-7.388	149.798	Earthquake	33.0	NaN	NaN	5.5	MB	...	NaN	NaN	NaN	NaN	NaN	USP0000A12	US	US	US	Reviewed
3382	02/26/1975	04:48:55	85.047	97.969	Earthquake	33.0	NaN	NaN	5.6	MS	...	NaN	NaN	NaN	NaN	NaN	USP0000A1H	US	US	US	Reviewed

5 rows × 21 columns

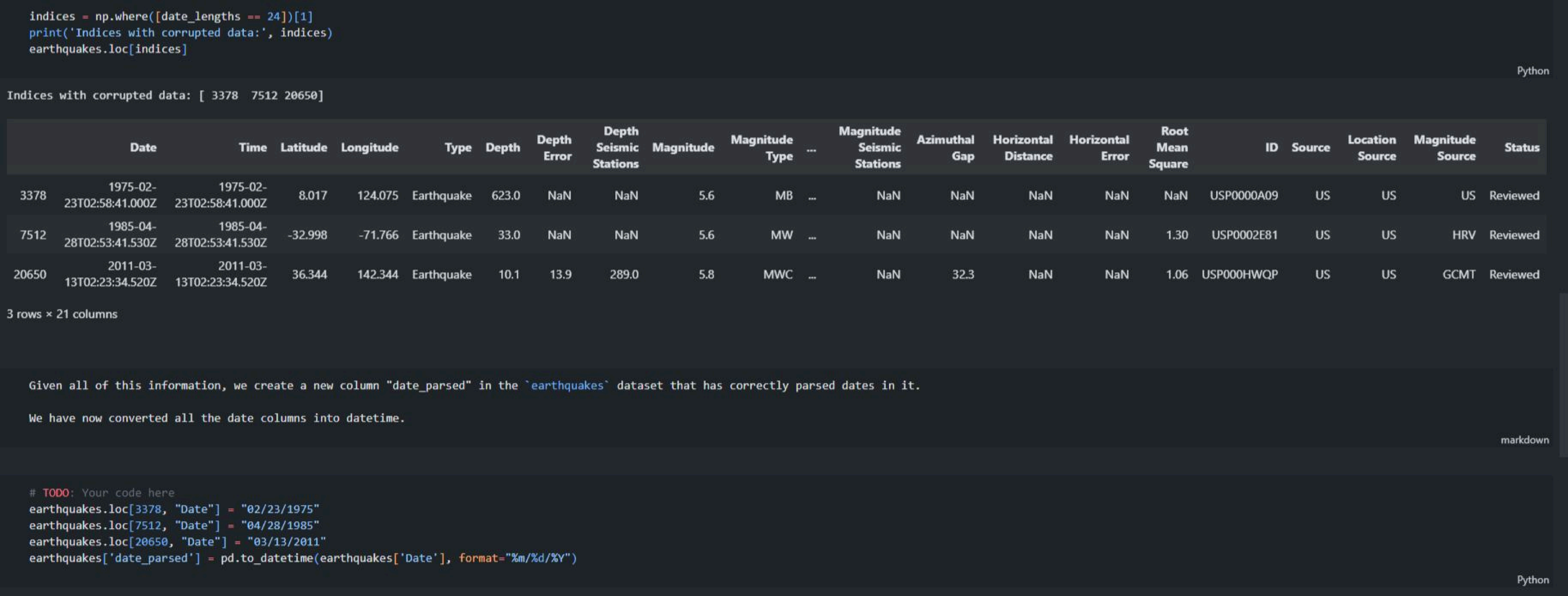
This does appear to be an issue with data entry: ideally, all entries in the column have the same format. We can get an idea of how widespread this issue is by checking the length of each entry in the "Date" column.

```
date_lengths = earthquakes.Date.str.len()
date_lengths.value_counts()
```

```
10    23409
24        3
Name: Date, dtype: int64
```

Looks like there are two more rows that has a date in a different format. We Run the code cell below to obtain the indices corresponding to those rows and print the data.

```
indices = np.where([date_lengths == 24])[1]
print('Indices with corrupted data:', indices)
earthquakes.loc[indices]
```



```python
indices = np.where([date_lengths == 24])[1]
print('Indices with corrupted data:', indices)
earthquakes.loc[indices]
```

```
Indices with corrupted data: [ 3378  7512 20650]
```

	Date	Time	Latitude	Longitude	Type	Depth	Depth Error	Depth Seismic Stations	Magnitude	Magnitude Type	...	Magnitude Seismic Stations	Azimuthal Gap	Horizontal Distance	Horizontal Error	Root Mean Square	ID	Source	Location Source	Magnitude Source	Status
3378	1975-02-23T02:58:41.000Z	1975-02-23T02:58:41.000Z	8.017	124.075	Earthquake	623.0	NaN	NaN	5.6	MB	...	NaN	NaN	NaN	NaN	NaN	USP0000A09	US	US	US	Reviewed
7512	1985-04-28T02:53:41.530Z	1985-04-28T02:53:41.530Z	-32.998	-71.766	Earthquake	33.0	NaN	NaN	5.6	MW	...	NaN	NaN	NaN	NaN	1.30	USP0002E81	US	US	HRV	Reviewed
20650	2011-03-13T02:23:34.520Z	2011-03-13T02:23:34.520Z	36.344	142.344	Earthquake	10.1	13.9	289.0	5.8	MWC	...	NaN	32.3	NaN	NaN	1.06	USP000HWQP	US	US	GCMT	Reviewed

3 rows × 21 columns

Given all of this information, we create a new column "date_parsed" in the `earthquakes` dataset that has correctly parsed dates in it.

We have now converted all the date columns into datetime.

```python
# TODO: Your code here
earthquakes.loc[3378, "Date"] = "02/23/1975"
earthquakes.loc[7512, "Date"] = "04/28/1985"
earthquakes.loc[20650, "Date"] = "03/13/2011"
earthquakes['date_parsed'] = pd.to_datetime(earthquakes['Date'], format="%m/%d/%Y")
```

3) Select the day of the month

Create a Pandas Series day_of_month_earthquakes containing the day of the month from the "date_parsed" column.

```
# try to get the day of the month from the date column
day_of_month_earthquakes = earthquakes['date_parsed'].dt.day
```

4) Plot the day of the month to check the date parsing

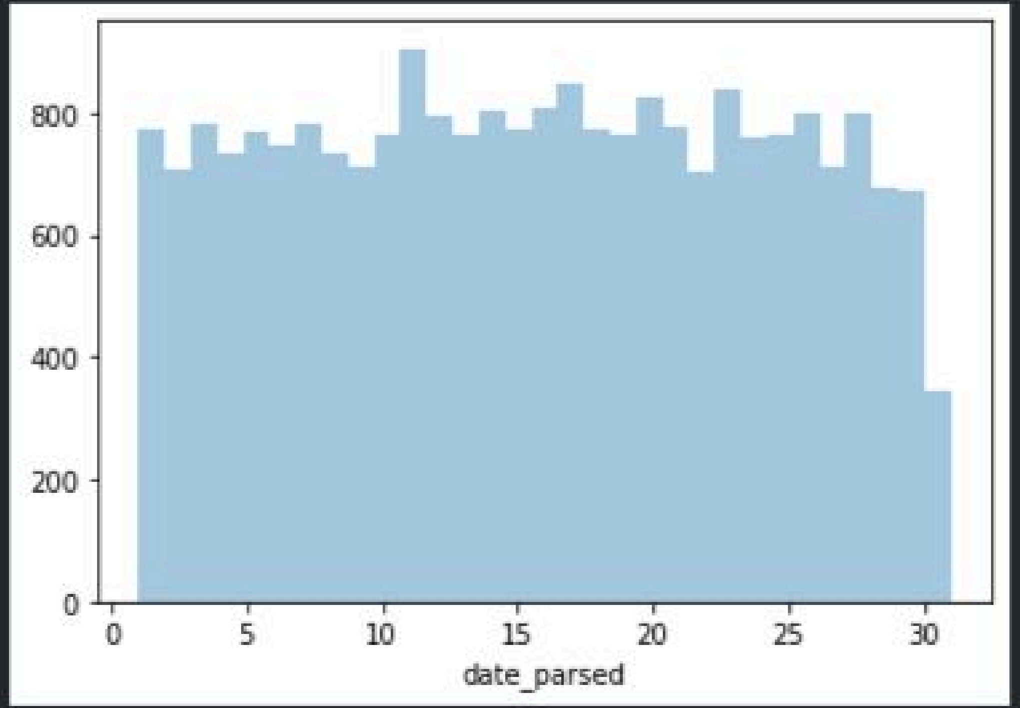
Plot the days of the month from your earthquake dataset.

```
# TODO: Your code here!
# remove na's
day_of_month_earthquakes = day_of_month_earthquakes.dropna()

# plot the day of the month
sns.distplot(day_of_month_earthquakes, kde=False, bins=31)
```

Python

```
/opt/conda/lib/python3.7/site-packages/seaborn/distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function
  warnings.warn(msg, FutureWarning)
```

```
<AxesSubplot:xlabel='date_parsed'>
```



Now we have visualized a graph that shows the days of the month.