Setup

The questions below will give you feedback on your work. Run the following cell to set up the feedback system.

In [1]:

from learntools.core import binder

binder.bind(globals())

from learntools.data\_cleaning.ex3 import \*

print("Setup Complete")

Setup Complete

linkcode

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 oc cured between 1965 and 2016.

*# 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

You'll be working with the "Date" column from the earthquakes dataframe. Investigate this column now: does it look like it contains dates? What is the dtype of the column?

In [3]

*# TODO: Your code here!*

check\_date = earthquakes.Date.dtype

check\_date

Out[3]:

dtype('O')

Once you have answered the question above, run the code cell below to get credit for your work.

In [4]:

*# Check your answer (Run this code cell to receive credit!)*

q1.check()

Correct:

The "Date" column in the earthquakes DataFrame does have dates. The dtype is "object".

In [5]:

*# Line below will give you a hint*

q1.hint()

Hint: Use earthquakes['Date'].head() to check that the column contains dates and verify that it has dtype "object". You can also use earthquakes['Date'].dtype to verify the dtype

In [6].

earthquakes[3378:3383]

date\_lengths = earthquakes.Date.str.len()

date\_lengths.value\_counts()

Out[7]:

Date

10 23409

24 3

Name: count, dtype: int64

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

In [8]:

indices = np.where([date\_lengths == 24])[1]

print('Indices with corrupted data:', indices)

earthquakes.loc[indices]

Indices with corrupted data: [ 3378 7512 20650]

Out[8]:

|  | 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, it's your turn to create a new column "date\_parsed" in the earthquakes dataset that has correctly parsed dates in it.

In [9]:

*# 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')

*# Check your answer*

q2.check()

Correct

In [10]:

*# Lines below will give you a hint or solution code*

q2.hint()

q2.solution()

Hint: Since there are only three rows with a fancy type, you might consider manually editing them. For instance, you can begin by setting earthquakes.loc[3378, "Date"] = "02/23/1975".

Solution:

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.

In [11]:

*# try to get the day of the month from the date column*

day\_of\_month\_earthquakes = earthquakes['date\_parsed'].dt.day

*# Check your answer*

q3.check()

Correct

In [12]:

*# Lines below will give you a hint or solution code*

q3.hint()

q3.solution()

Hint: Use the .dt accessor.

Solution:

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.

In [13]:

*# TODO: Your code here!*

sns.histplot(earthquakes['date\_parsed'].dt.day)

Out[13]:

<Axes: xlabel='date\_parsed', ylabel='Count'>

Does the graph make sense to you?

In [14]:

*# Check your answer (Run this code cell to receive credit!)*

q4.check()

Correct:

The graph should make sense: it shows a relatively even distribution in days of the month,which is what we would expect.

In [15]:

*# Line below will give you a hint*

q4.hint()

Hint: Remove the missing values, and then use sns.distplot() as follows:

*# 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)