19. Pandas - Part 2

October 27, 2022

[ ]:
```
import pandas as pd
```

1 How to sort a DataFrame or a Series?

[ ]:
```
# read a dataset of top-rated IMDb movies into a DataFrame
df = pd.read_csv('data/imdb_1000.csv')
```

[ ]:
```
df.head()
```

[ ]:
```
df.shape
```

Note: None of the sorting methods below affect the underlying data. (In other words, the sorting is temporary).

[ ]:
```
# sort the 'title' Series in ascending order (returns a Series)
df.title.sort_values().head(10)
```

[ ]:
```
# sort in descending order instead
df.title.sort_values(ascending=False).head()
```

[ ]:
```
# sort the entire DataFrame by the 'title' Series (returns a DataFrame)
df.sort_values('title').head()
```

[ ]:
```
# sort in descending order instead
df.sort_values('title', ascending=False).head()
```

[ ]:
```
# sort the DataFrame first by 'content_rating', then by 'duration'
df.sort_values(['content_rating', 'duration'])
```

2 How to filter rows of a pandas DataFrame by column value?

[ ]:
```
df.head()
```

[ ]:
```
# examine the number of rows and columns
df.shape
```

Goal: Filter the DataFrame rows to only show movies with a 'duration' of at least 200 minutes.

[ ]:
```
type(df.duration)
```

[ ]:
```
# create a list in which each element refers to a DataFrame row: True if the␣
 ↪row satisfies the condition, False otherwise
booleans = []
for length in df.duration:
    if length >= 200:
        booleans.append(True)
    else:
        booleans.append(False)
```

[ ]:
```
# confirm that the list has the same length as the DataFrame
len(booleans)
```

[ ]:
```
booleans[:10]
```

[ ]:
```
# use bracket notation with the boolean Series to tell the DataFrame which rows␣
 ↪to display
df[booleans]
```

[ ]:
```
# simplify the steps above: no need to write a for loop to create 'is_long'
# pandas will broadcast the comparison
is_long = df.duration >= 200
```

[ ]:
```
df[is_long]
```

[ ]:
```
# or equivalently, write it in one line (no need to create the 'is_long' object)
df[df.duration >= 200]
```

[ ]:
```
# select the 'genre' Series from the filtered DataFrame
df[df.duration >= 200].genre
```

[ ]:
```
df["len_title"] = df.title.str.len()
```

[ ]:
```
df
```

[ ]:
```
df.sort_values(["len_title","title"]).head(20)
```

3 How to find Unique values from a Series

[ ]:
```
df.genre.unique()
```

[ ]:
```
df.genre.nunique()
```

4 How to apply multiple filter criteria to a pandas DataFrame?

[ ]:
```
df.head()
```

[ ]:
```
# filter the DataFrame to only show movies with a 'duration' of at least 200␣
↪minutes
df[df.duration >= 200]
```

logical operators:

- **and**: True only if **both sides** of the operator are True
- **or**: True if **either side** of the operator is True

Rules for specifying **multiple filter criteria** in pandas:

- use **&** instead of **and**
- use **|** instead of **or**
- add **parentheses** around each condition to specify evaluation order

Q. Filter the DataFrame of long movies (duration >= 200) to only show movies which also have a 'genre' of 'Drama'

[ ]:
```
# use the '&' operator to specify that both conditions are required
df[(df.duration >=200) & (df.genre == 'Drama')]
```

Q. Filter the original DataFrame to show movies with a 'genre' of 'Crime' or 'Drama' or 'Action'

[ ]:
```
# use the '|' operator to specify that a row can match any of the three criteria
df[(df.genre == 'Crime') | (df.genre == 'Drama') | (df.genre == 'Action')].
↪head(10)
```

[ ]:
```
# or equivalently, use the 'isin' method
df[df.genre.isin(['Crime', 'Drama', 'Action'])]
```