

Do Sequels Earn More? - Selecting Specific Columns

The screenshot shows a web browser window with a DataCamp exercise titled "Do sequels earn more?". The exercise is part of a course "Joining Data with pandas". The instructions state: "It is time to put together many of the aspects that you have learned in this chapter. In this exercise, you'll find out which movie sequels earned the most compared to the original movie. To answer this question, you will merge a modified version of the `sequels` and `financials` tables where their index is the movie ID. You will need to choose a merge type that will return all of the rows from the `sequels` table and not all the rows of `financials` table need to be included in the result. From there, you will join the resulting table to itself so that you can compare the revenue values of the original movie to the sequel. Next, you will calculate the difference between the two revenues and sort the resulting dataset. The `sequels` and `financials` tables have been provided."

The instructions also state: "Select the `title_org`, `title_seq`, and `diff` columns of `orig_seq` and save this as `titles_diff`."

The code editor shows the following Python code:

```
1 # Merge sequels and financials on index id
2 sequels_fin = sequels.merge(financials, on='id', how='left')
3
4 # Self merge with suffixes as inner join with left on sequel and right on id
5 orig_seq = sequels_fin.merge(sequels_fin, how='inner', left_on='sequel',
6                             right_on='id', right_index=True,
7                             suffixes=('_org', '_seq'))
8
9 # Add calculation to subtract revenue_org from revenue_seq
10 orig_seq['diff'] = orig_seq['revenue_seq'] - orig_seq['revenue_org']
11
12 # Select the title_org, title_seq, and diff
13 titles_diff = orig_seq[['title_org', 'title_seq', 'diff']]
```

Screenshot showing the exercise context for selecting specific columns after merging sequels and financials.

Code Answer:

```
# Merge sequels and financials on index id
sequels_fin = sequels.merge(financials, on='id', how='left')
```

```
# Self merge with suffixes as inner join with left on sequel and right on id
orig_seq = sequels_fin.merge(sequels_fin, how='inner', left_on='sequel',
                             right_on='id', right_index=True,
                             suffixes=('_org', '_seq'))
```

```
# Add calculation to subtract revenue_org from revenue_seq
orig_seq['diff'] = orig_seq['revenue_seq'] - orig_seq['revenue_org']
```

```
# Select the title_org, title_seq, and diff columns
titles_diff = orig_seq[['title_org', 'title_seq', 'diff']]
```

```
# Print the resulting DataFrame
print(titles_diff)
```

Explanation:

1. The first step merges the 'sequels' table with the 'financials' table on the 'id' column using a left join, ensuring all rows from the 'sequels' table are included.
2. A self join is then performed on the resulting DataFrame ('sequels_fin') using the 'sequel' column on the left and the 'id' column on the right. The `suffixes=('_org', '_seq')` parameter is used to differentiate columns for the original movie and its sequel.
3. The 'diff' column is created by calculating the difference in revenue between the sequel and the original movie.
4. Finally, the required columns ('title_org', 'title_seq', and 'diff') are selected and stored in the 'titles_diff' DataFrame. This subset shows the titles of the original movies and their sequels, along with the revenue difference.