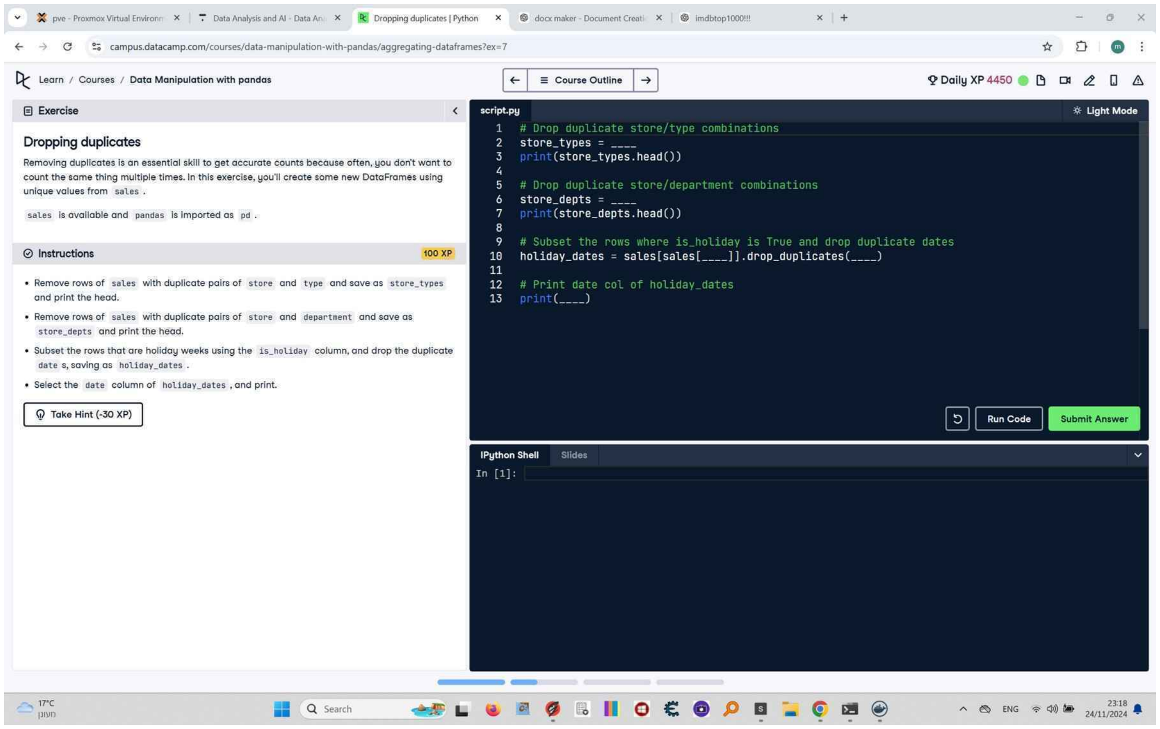
Dropping Duplicates (Updated Solution)

This document includes the question, the updated solution, and a breakdown of the code provided in the screenshot.

Uploaded Screenshot

Below is the updated screenshot of the task:



campus.datacamp.com/courses/data-manipulation-with-pandas/aggregating-dataframes?ex=7

Learn / Courses / Data Manipulation with pandas

Exercise

Dropping duplicates

Removing duplicates is an essential skill to get accurate counts because often, you don't want to count the same thing multiple times. In this exercise, you'll create some new DataFrames using unique values from `sales`.

`sales` is available and `pandas` is imported as `pd`.

Instructions 100 XP

- Remove rows of `sales` with duplicate pairs of `store` and `type` and save as `store_types` and print the head.
- Remove rows of `sales` with duplicate pairs of `store` and `department` and save as `store_depts` and print the head.
- Subset the rows that are holiday weeks using the `is_holiday` column, and drop the duplicate `date` s, saving as `holiday_dates`.
- Select the `date` column of `holiday_dates`, and print.

script.py

```python
# Drop duplicate store/type combinations
store_types = ____
print(store_types.head())

# Drop duplicate store/department combinations
store_depts = ____
print(store_depts.head())

# Subset the rows where is_holiday is True and drop duplicate dates
holiday_dates = sales[sales[____]].drop_duplicates(____)

# Print date col of holiday_dates
print(____)
```

Question

1. Remove rows of `sales` with duplicate pairs of `store` and `type` and save as `store_types`, then print the head.
2. Remove rows of `sales` with duplicate pairs of `store` and `department` and save as `store_depts`, then print the head.
3. Subset the rows that are holiday weeks using the `is_holiday` column and drop duplicate `date`s, saving as `holiday_dates`.
4. Select the `date` column of `holiday_dates` and print.

Updated Answer

```python
# Drop duplicate store/type combinations
store_types = sales.drop_duplicates(subset=['store', 'type'])
print(store_types.head())

# Drop duplicate store/department combinations
store_depts = sales.drop_duplicates(subset=['store', 'department'])
```

```
print(store_depts.head())

# Subset rows where is_holiday is True and drop duplicate dates
holiday_dates = sales[sales['is_holiday']].drop_duplicates(subset='date')

# Print date column of holiday_dates
print(holiday_dates['date'])
```

Code Explanation

# Explanation of the code:

1. `sales.drop_duplicates(subset=['store', 'type'])`: Removes duplicate rows based on the combination of `store` and `type` columns.

2. `sales.drop_duplicates(subset=['store', 'department'])`: Removes duplicate rows based on the combination of `store` and `department` columns.

3. `sales[sales['is_holiday']]`: Filters the rows where the `is_holiday` column is `True`.

4. `.drop_duplicates(subset='date')`: Removes duplicate rows based on the `date` column within the filtered DataFrame.

5. `holiday_dates['date']`: Selects the `date` column from the `holiday_dates` DataFrame.