DESCRIPTION

Table: Activity

```
+--------------+---------+
| Column Name  | Type    |
+--------------+---------+
| player_id    | int     |
| device_id    | int     |
| event_date   | date    |
| games_played | int     |
+--------------+---------+
```

(player_id, event_date) is the primary key (combination of columns with unique values) of this table.

This table shows the activity of players of some games.

Each row is a record of a player who logged in and played a number of games (possibly 0) before logging out on someday using some device.

Write a solution to report the **fraction** of players that logged in again on the day after the day they first logged in, **rounded to 2 decimal places**. In other words, you need to determine the number of players who logged in on the day immediately following their initial login, and divide it by the number of total players.

The result format is in the following example.

Example 1:

Input:

Activity table:

```
+-----------+-----------+------------+--------------+
| player_id | device_id | event_date | games_played |
+-----------+-----------+------------+--------------+
| 1         | 2         | 2016-03-01 | 5            |
| 1         | 2         | 2016-03-02 | 6            |
| 2         | 3         | 2017-06-25 | 1            |
| 3         | 1         | 2016-03-02 | 0            |
| 3         | 4         | 2018-07-03 | 5           |
+-----------+-----------+------------+--------------+
```

Output:

```
+-----------+
| fraction  |
+-----------+
| 0.33      |
+-----------+
```

Explanation:

Only the player with id 1 logged back in after the first day he had logged in so the answer is 1/3 = 0.33

SOLUTION

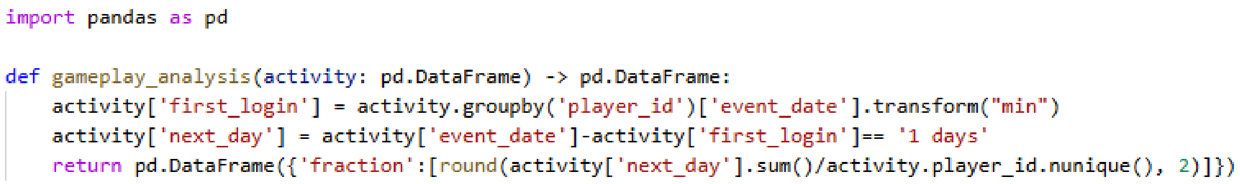
Option 1:

- Add ‘first_login’ column into ‘activity’ with minimum event_date using groupby(), and transform() in which ‘min’ is used to select first ‘event_date’ for each player_id
- Add ‘next_day’ column with boolean values where event_date – ‘first_login’ equals one day
- Calculate fraction by finding the total of ‘next_day’ column divided by number of unique players, and round it to two decimals

```
import pandas as pd

def gameplay_analysis(activity: pd.DataFrame) -> pd.DataFrame:
    activity['first_login'] = activity.groupby('player_id')['event_date'].transform("min")
    activity['next_day'] = activity['event_date']-activity['first_login']== '1 days'
    return pd.DataFrame({'fraction':[round(activity['next_day'].sum()/activity.player_id.nunique(), 2)]})
```

- Snapshot of the same code above for readability purposes



```
import pandas as pd

def gameplay_analysis(activity: pd.DataFrame) -> pd.DataFrame:
    activity['first_login'] = activity.groupby('player_id')['event_date'].transform("min")
    activity['next_day'] = activity['event_date']-activity['first_login']== '1 days'
    return pd.DataFrame({'fraction':[round(activity['next_day'].sum()/activity.player_id.nunique(), 2)]})
```