Sophia Ibarrola DS210 Project

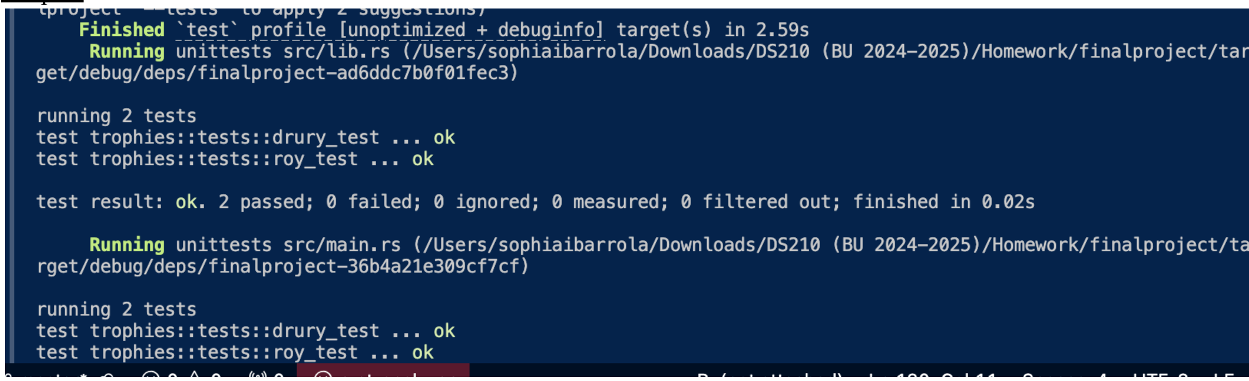
1. **Project Overview & Data Processing**
    a. Goal: My goal for this project was to host a College Hockey Award show that follows the same pattern as the NHL awards given out each season. These awards include the Travis Roy (NHL: Hart, MVP), the John Cullen (NHL: Ross, most points), the Chris Drury (NHL: Richard, most goals), the Poulin (NHL: Selke, best defensive forward), the Hutson (NHL: Norris, best defenseman), and the Yegorov (NHL: Lady Byng, most sportsmanlike).
    b. Dataset: This dataset comes from using the Pivot Stats tab of the United States College Hockey Organization website.
    c. Data Processing: The first thing I did for my project was to clean up some of the data and add new stats to the DataFrame. I found this easiest to do in Python through Pandas because it's what I'm most familiar with, and I added the .ipynb file in the finalproject folder. The output of this file is the final dataset I'm using, with all the college hockey players and their corresponding statistics. I will specify how I loaded it into Rust in my Code Structure; it should be noted that I used Polars in order to accomplish my calculations.
2. **Code Structure**
    a. Dataframe module: I created the module dataframe.rs to store some base functions for making dataframes to allow for the calculations needed later.
        i. Read_csv: This function was created so that I would have a more generic way to read a csv through Polars. I made this generic so that I could run tests on a smaller test csv. The input is the path, and the output is a Polars LazyFrame, a struct which allows me to apply pandas-esque functions to the dataframe.
        ii. Team_df: This function was made to create a dataframe with just the data for the specified team. The inputs are the path to the csv file (so I can run read_csv) and the team to grab the data.
    b. Metrics module: I created the module metrics.rs to calculate the different offensive and defensive metrics that would be needed for the more "subjective" awards.
        i. Team_add_col_df: This function calculates the base columns that are needed to calculate the different metrics. This includes blocks/game and plus-minus/game which were needed for the defensive metric; and goals/game, points/game, and special teams/game (power play goals/game + short handed goals/game) which were needed for the offensive metric. The inputs are the path and team so that team_df can be run, and the output is a LazyFrame with the new columns.
        ii. Team_metric_col_df: This function calculates the offensive and defensive metrics based on a sum of the criteria laid out in team_add_col_df. The inputs are the path and team so that team_add_col_df can be run and the output is a LazyFrame with the new columns.
        iii. Team_final_metric_df: This function calculates the sum of the two metrics (for the MVP award). The inputs are the path and team so that team_metric_col_df can be run. The output is a LazyFrame with the new column.
        iv. Defensemen_df: This function creates a dataframe with the new columns (the base columns, the offensive metric, the defensive metric, and the cumulative metric) for just the defensemen on the team. The inputs are the path and team so that team_final_metric_df can be run, and the output is a LazyFrame with only the statistics for the defensemen.
        v. Forward_df: This function does the same as defensemen_df, but for the forwards on the team instead. The inputs are the path and team so that team_final_metric_df can be run, and the output is a LazyFrame with only the statistics for the forwards.
    c. Trophies Module: I created the module trophies.rs to return the winners of each trophy based on the metrics calculated in metrics.rs. Unless otherwise stated, I used team_final_metric_df to sort the metric columns, meaning the input would be the path and team. After sorting, I isolated the first row (the winner) and returned the player name as a string.
        i. Roy: This award is the equivalent to the Hart trophy, which is awarded to the league MVP. This award was filtered based on the cumulative metric.
        ii. Cullen: This award is the equivalent to the Art Ross, which is awarded to the points leader. This award was filtered based on points in descending order.
        iii. Drury: This award is the equivalent to the Rocket Richard, which is awarded to the goals leader. This award was filtered based on goals in descending order.

   iv. Poulin: This award is the equivalent to the Selke, which is awarded to the best defensive forward. Instead of using team_final_metric_df, I used forward_df, since this only deals with forwards, and filtered based on defensive_metric in descending order.
   v. Hutson: This award is the equivalent to the Norris, which is awarded to the best defenseman. Instead of using team_final_metric_df, I used defensemen_df, since this only deals with defensemen, and filtered based on defensive_metric in descending order.
   vi. Yegorov: This award is the equivalent to the Lady Byng, which is awarded to the most sportsmanlike. This award was filtered on penalty minutes in ascending order.

d. main.rs: In main.rs, I first created an input option to put in the team desired. After this, I created print lines to announce the winners of each trophy.

3. Tests

a. Output:

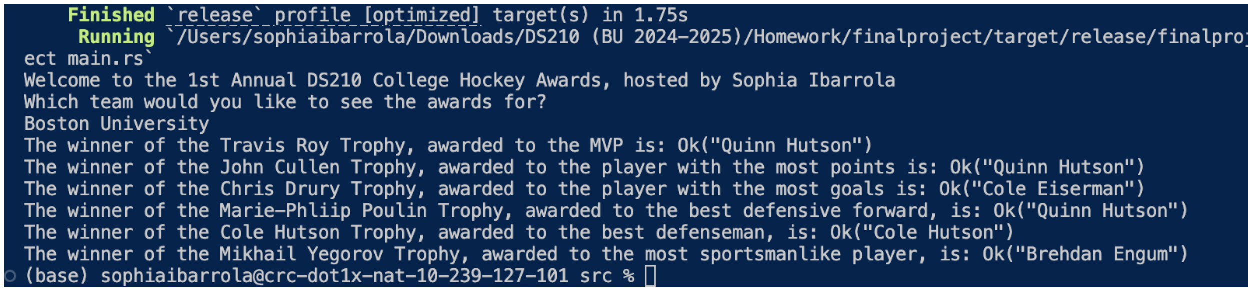


```
(project --tests` to apply 2 suggestions)
    Finished `test` profile [unoptimized + debuginfo] target(s) in 2.59s
     Running unittests src/lib.rs (/Users/sophiaibarrola/Downloads/DS210 (BU 2024-2025)/Homework/finalproject/tar
get/debug/deps/finalproject-ad6ddc7b0f01fec3)

running 2 tests
test trophies::tests::drury_test ... ok
test trophies::tests::roy_test ... ok

test result: ok. 2 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.02s

     Running unittests src/main.rs (/Users/sophiaibarrola/Downloads/DS210 (BU 2024-2025)/Homework/finalproject/ta
rget/debug/deps/finalproject-36b4a21e309cf7cf)

running 2 tests
test trophies::tests::drury_test ... ok
test trophies::tests::roy_test ... ok
```

b. Reasoning: I tested one of the subjective awards (Roy/MVP) and one of the objective awards (Drury/goals). I found that this was a good test because the trophy functions are the culmination of the rest of the code, and testing them means testing the breadth of my code.

4. Results

a. Outputs:



```
    Finished `release` profile [optimized] target(s) in 1.75s
     Running `/Users/sophiaibarrola/Downloads/DS210 (BU 2024-2025)/Homework/finalproject/target/release/finalproj
ect main.rs`
Welcome to the 1st Annual DS210 College Hockey Awards, hosted by Sophia Ibarrola
Which team would you like to see the awards for?
Boston University
The winner of the Travis Roy Trophy, awarded to the MVP is: Ok("Quinn Hutson")
The winner of the John Cullen Trophy, awarded to the player with the most points is: Ok("Quinn Hutson")
The winner of the Chris Drury Trophy, awarded to the player with the most goals is: Ok("Cole Eiserman")
The winner of the Marie-Phliip Poulin Trophy, awarded to the best defensive forward, is: Ok("Quinn Hutson")
The winner of the Cole Hutson Trophy, awarded to the best defenseman, is: Ok("Cole Hutson")
The winner of the Mikhail Yegorov Trophy, awarded to the most sportsmanlike player, is: Ok("Brehdan Engum")
○ (base) sophiaibarrola@crc-dot1x-nat-10-239-127-101 src % ▯
```

b. Interpretation: I'm very happy with these results, because the winners more or less align with who I would expect to have won them. With the Cullen, Drury, and Yegorov trophies, the winners were pretty cut and dry. However, for the subjective trophies (Roy, Poulin, and Huston), I'm pleased that the results were reflective of what I would expect voting would be, if there were to be a board of judges that voted on this as there is in NHL hockey.

5. Usage Instructions

a. Running code: Input the name of a college hockey team, ex: "Boston University". It should also be noted that Polars must be installed in order to run this.

****NOTE ABOUT THE GIT COMMIT HISTORY:** I unfortunately didn't initialize the Git repository early enough in the process, so the commit history is from basically when I was entirely done with the process. I can't find a way to transfer the local history (I did the project entirely on the Visual Studio Code app on my laptop). I'm sorry that there is not an entire history of the program.