Assignment 2: Sports Analytics using Shell Programming

CS253 - 2023

IIT Kanpur

For this assignment, you can use sed, grep, and awk commands in a shell script to do the desired tasks. You can use man pages to know more about these commands. You will also need to use output redirection (>, >>, or some equivalent method) to write to a file. You may use additional commands if needed.

[1] Write a script named **comm.sh,** which takes two files as input. Output error if any of the files does not exist. The script finds common lines between two files and prints th on the terminal. Do not use the comm command. You need to write logic yourself. Parse each file line by line and match with the other file. Do not use any automatic command for achieving this.

- Ignore empty lines
- **Useful constructs:** nested loop, if condition

[2] Write a script named **goals_above_30.sh** that takes two files as arguments.

- The first file is the input file. If the first file does not exist, then throw an error saying that the file does not exist.
- The second file is the output file which the script will create
- The input file is a CSV (Comma Separated Values) file representing a dataset of England Premier League Players from 2018 to 2019. The data is divided into different columns: full_name,age,birthday,birthday_GMT,league,season,position,Current Club,minutes_played_overall,minutes_played_home,minutes_played_away,nationality,appearances_overall,appearances_home,appearances_away,goals_overall,goals_home,goals_away,assists_overall,assists_home,assists_away,penalty_goals,penalty_misses,clean_sheets_overall,clean_sheets_home,clean_sheets_away,conceded_overall,conceded_home,conceded_away,yellow_cards_overall,red_cards_overall,goals_involved_per_90_overall,assists_per_90_overall,goals_per_90_overall,goals_per_90_home,goals_per_90_away,min_per_goal_overall,conceded_per_90_overall,min_per_conceded_overall,min_per_match,min_per_card_overall,min_per_assist_overall,cards_per_90_overall,rank_in_league_top_attackers,rank_in_league_top_midfielders,rank_in_league_top_defenders,rank_in_club_top_scorer.
- We have provided the input file: **england-premier-league-players-2018-to-2019-stats.csv** for use. You can assume that there are NO errors in the input file if it exists. The input file name can be other than england_league.csv, actual data can be different, but the data format (columns) will remain the same.

- Now, your script should find the players whose age is more than 30 and who scored at least 1 goal (column: **goals_overall**) and write it to the second file. Overwrite the second file if it exists.
- Print all fields along with column names
- **Useful Commands:** cat, awk

[3] Write a script named **top_goals_scorer.sh** that takes two files as arguments

- It should print all columns of the top 10 players with the highest goals_overall.
- If there is a tie among players, for example, A scored 20 goals and B scored 20 goals, then print both and consider it as count 1. Basically, the number of players can be more than 10, but the set of goals scored will have unique 10 values.
- The output will contain all fields
- **Useful commands and constructs:** cat, awk, sort, uniq, nested loop, if condition

[4] Write a script named **players.sh** that prints all unique players. Fields: full_name, nationality, position,goals_overall.

- Sort the players in descending manner by using the column goals_overall.
- Also, players with the same position should appear first. That means first print players that play as Goalkeeper, after that Defender, and so on.
- **Useful commands and constructs:** awk, sort, nested loop, if condition

For the questions 2, 3, 4:

- Take two files as arguments
- The first file is the input CSV file (**england-premier-league-players-2018-to-2019-stats.csv**), and the second one is the output file.
- Print an error message if the input csv does not exist.
- If it exists, then assume that format is correct.
- The second file is the output file which will be created or overwritten by the script
- **You may not use all the recommended shell script constructs and commands and may use other shell script constructs and commands that are not suggested.**

You can create one or more awk script files, or any other auxiliary file if needed. Submit all the source files in zipped format. There is no need to submit a CSV file or any generated output file. The directory must contain 4 scripts with filenames as given in the questions (please provide exact filenames as given; otherwise, we won't evaluate the submission), and any auxiliary awk or other script files you have used.