Project Report – UCDPA_Mark_Dillon

GitHub URL

https://github.com/darkmillon/UCDPA_Mark_Dillon

Abstract

Project assignment for UCD PA was to complete analysis on real world dataset using the modules learned during our 8 weeks, implement these learnings to provide relevant and tailored insights, visualizing the results. All work was carried out using Jupyter notebooks via the Anaconda platform. Python was our base coding language with use of several packages such as NumPy, Pandas and visualization packages such as Matplotlib and Seaborn.

Introduction

For this project I selected 2 sports data sets based on golf statistics for the PGA Tour of America.

Dataset 1 included all players and tournaments completed from 2015 to 2022 on the PGA Tour. Supporting this was Dataset 2 which was World Golf Rankings file, this was selected as a good support to allow merging two datasets of relevant data.

I selected both datasets for the project as sports analytics is the area I am keen to explore as I continue to upskill my analytics coding skillset, sports analytics is now a massive area within any sport. It is an industry that has really adopted digital analytics and is being utilized daily to inform and improve performance.

The particular dataset of PGA Golf stats was useful as golf measure a large amount of simple but SMART metrics.

I am also a keen golfer with local club and all our own amateur statistics are being tracked via online platform Howdidido and Golf Ireland, the long term aim is to improve my skillset to use the API and Web scraping tools on these sites for my own information.

Dataset

Dataset sources:

https://www.kaggle.com/datasets/robikscube/pga-tour-golf-data-20152022

https://www.kaggle.com/datasets/bradklassen/official-world-golf-ranking-data

Two datasets were chosen from Kaggle after researching various online datasets for analysis, Kaggle provided the best information on what was available. The Kaggle repository is extremely user friendly and simple to access.

The main dataset was PGA Tour data 2015 – 2022, this gave the most complete statistics for a large timeframe on golf tour that I could find, with the PGA Tour being the biggest and most well-known tour within golf. I also fact checked the stats against the PGA Tour website to confirm accuracy, spot-checking several statistics on winners and strokes score.

The dataset provided me with key variables that could be used to analyze on such as *player*, *tournament name, course, finishing position, strokes gained.*

From early analysis of the dataset also their was options available for indexes with fields like *player_id* and *tournament_id* being unique.

If needed we could build separate tables and build relationship tables based of these key fields.

To support this and give me option to merge datasets with relevant data I selected a database containing the Official world Golf rankings. This data was only up to 2019 but I felt it would allow me to demonstrate the skillset effectively.

For the purposes of demonstrating the SQL and API tasks for this assignment, I also choose some open source available:

Chinook database was download and used to demonstrate for SQL queries using sqlite3.

The Chinook database represents virtual media store, including several tables. It is a fully functional relationship database.

For the API calls, rather than using one that was in our training, I choose the Chuck Norris jokes API: https://api.chucknorris.io/ - this is a small humorous API call that allows you to pull random jokes, I added this as last piece of my notebook to finish on a lighter note!

Implementation Process

The following is walk through process of the tasks carried out on the Jupyter notebook ***UCDPA_MarkDillon.ipynb*** with notes and explanations added:

- Firstly logged into Git Hub and created my repository with just a main branch, left it public to allow easy accessibility.
- Opened Anaconda navigator and launched Jupyter notebooks, I created a new notebook to start my work, I had set local directory as folder on my desktop where all project work was saved.
- Imported all the necessary packages I knew would be needed as we worked through the project requirements
- The dataset (which was downloaded to my local) was then read into pandas dataframe using command **`pd.read_csv`** while printing the head of the dataframe to check the layout
- Next I used a series of commands to check the database, checking the datatypes, if there is any null values and if so then how many of each, results here dictated what cleanups were needed to make this dataset better.

```
      .info() , .dtypes, .isna().any() , .isnull().sum() ,
.describe()
```

- I then cleaned the data removing unnecessary columns where possible and filling in values where appropriate.
- 3 columns were removed ['Unnamed: 2', 'Unnamed: 3', 'Unnamed: 4'], as they did not hold any values across the entire dataset.
- The argument `inplace=True` was one I had previously omitted and became useful as I worked through to make sure the changes to dataset were maintained.
- Two commands were ran to show various options to fill null values with a different value, firstly by column specific and the for entire dataset (need to be careful with this for large datasets)

```
golf_stats['pos'] = golf_stats['pos'].fillna(0)
golf_stats.fillna(value = 0,
                  inplace = True)
```

- For next step I needed to convert the datatype of certain columns to string format, this was completed using the `.astype` extension. Once ran then I checked to confirm using the `.dtype` call.

- Next step involved the index and manipulating with different indices, researched on different options here as I wasn't certain what would be the best fit. Finally I did decide to set an index on multiple fields here to try give me unique entry for each row.

- Using the `.set_index()` call we combined 3 columns to set the index.

- We then sorted the index alphabetically using the `.sort_values()` call, on player column.

- Lastly on the index we checked for duplicated and dropped the duplicate indexes except the first occurrence: `.drop_duplicates(keep='first')`

- Next steps was to start analyzing and visualizing the data to produce some insights, several graphs and tables were produced to achieve this result, graphs will be detailed in the results section, here is high level overview:

    - Graph 1: Horizontal barchart using matplotlib using the `.loc` call to find winners and count them using `value_counts`
    - Graph 2 is similar but vertical version with 2 dimensions, wins and cuts made, to illustrate the correlation between them.
    - Graph 3: Again a horizontal barchart using matplotlib using the `.loc` call to find players with most missed cuts and count them using `value_counts`

- I then sliced the main dataframe into an easier to use subset with the columns I needed

- We used the `groupby()` function to analyze the tournaments and combined it with the `.agg` call to get summary statistics on each tournament, which also sorting on standard deviation

- I then attempted to use a seaborn scatterplot to visualize the difference by tournament in standard deviation but found too many tournaments to display effectively for the user

- I then used the `groupby()` function again to get the mean purse by Tournament and sort in descending order.

- Next we looked at player appearances over each season and see the path, so I mapped a line plot to count number of player appearances v season, adding some extra commands to modify the output such as `marker, linestyle and xticks`

- Another plot showed us the correlation between par for course and actual score of player and large groupings with very little outliers

- We used a simple code call next to add a column `"avg single round"` based off dividing 2 other columns making sure we had correct datatypes to do so.

- Another step next was to use `pivot table` option to view 2 different options, pivoting on strokes as value for both season column and tournament column, giving us some insight to where years or tournaments had higher or lower mean strokes.

- Next plot showed us the grouping of players into where the majority are finishing strokes throughout the tournaments, using seaborn kernel density plot to visualize.

- Now I wanted to see the winners and tournaments they won, so ran simple script to print off various winners(pos==1) and send output to print. Allowing us to determine 327 winners over the dataset.

- Then followed to check who had won tournament more than 1, again using the `groupby()` and `count()` functions

- Second last graph we used scatter plot to analyze the correlation between finishing position `(pos)` and strokes gained `(sg_total)` to see as we would expect lower the finishing position the higher the strokes gained.

- Our final analysis on the dataset was on the `purse` field, to sum the total purses by winning players and then visualize these with barchart for any player over certain value, using seaborn bar chart for nicer visualization

- .Our next task included adding a new column based off a `defined custom function`, I simply added a column `message` to loop through using an if statement for all players and add message: Winner or Loser based on their finishing position in that tournament.

- I also used the `.loc and .iterrows()` functions to create a new column based of a calculation and a for loop

- Next major step was to load another csv file and merge the 2 together to produce another dataframe, we performed similar steps here to check the data as previously outlined above.

- I subsetted my main dataset and put only columns I wanted into new dataframe, once I had my 2 dataframes I merged with `left join` to keep all records in my first table but merge the ranking column into this, matching on player v name columns from each.

- Tidy ups were then completed with the new merged dataframe, to fill values needed, first copying from one column to another and 2nd column filling with large value

- The data was sorted on ranking to give us the correct order and then exported to new CSV using `.to_csv` command

- We now finished work on main datasets and used the chinook db to test our SQL knowledge, it was downloaded to my local and I connected using sqlite3.

- I found sqlite3 easier to use the mysql or sqlalchemy, it connected easier without errors

- We ran 2 main queries:
  - One to simply select all from the genres tables and print the output on a loop
  - Second was a inner join on the customer and invoice table and send output to a pandas dataframe

- More test SQL queries was completed on sqlite online and the log has been included using the GROUP BY, ORDER BY and COUNT functions

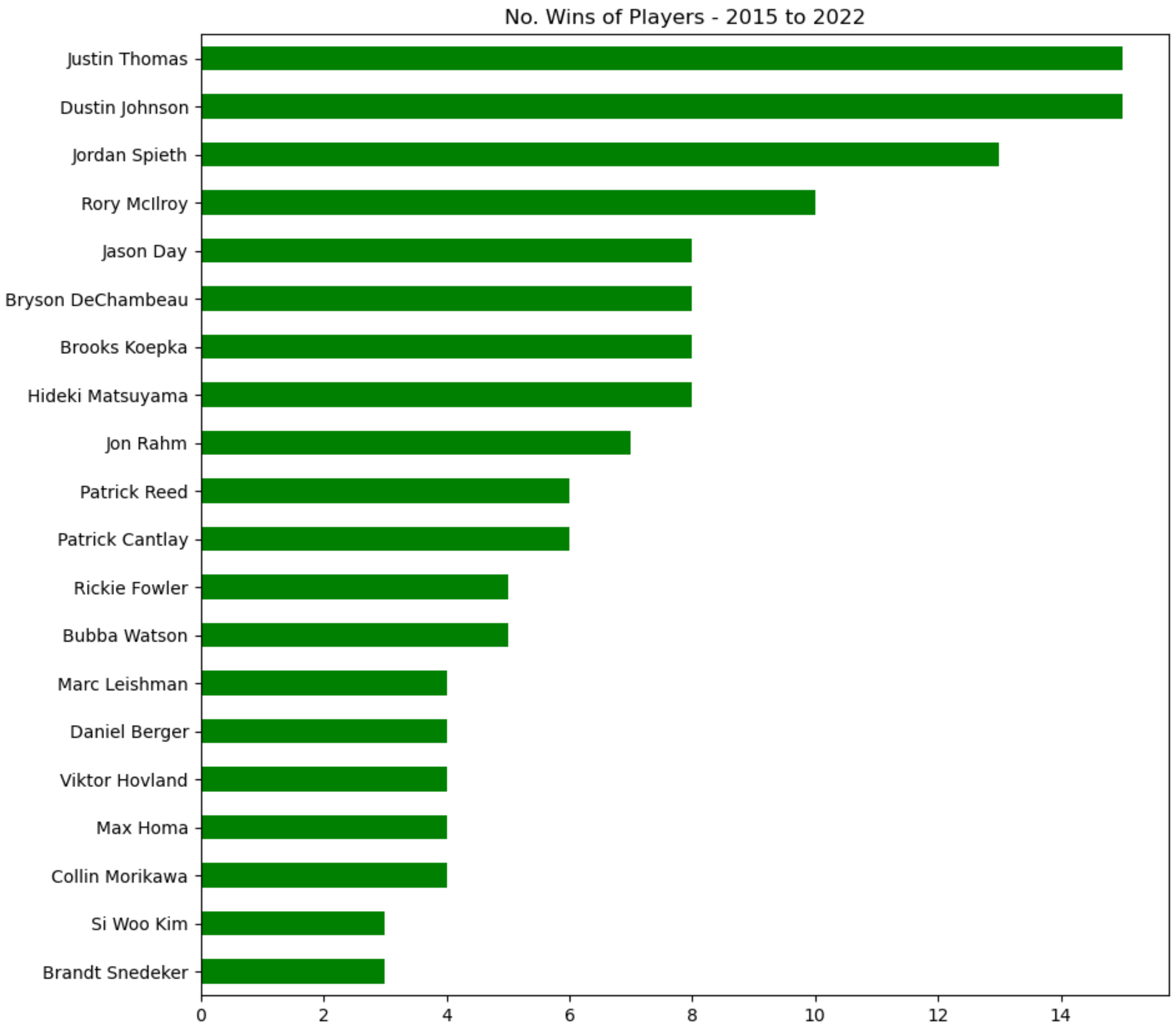
- Lastly I wanted to demonstrate API connection skills learnt over this course, we found a chuck Norris api from list of open api's on github

- We used this to pull back all the categories of jokes into a json and then display in a pandas data frame

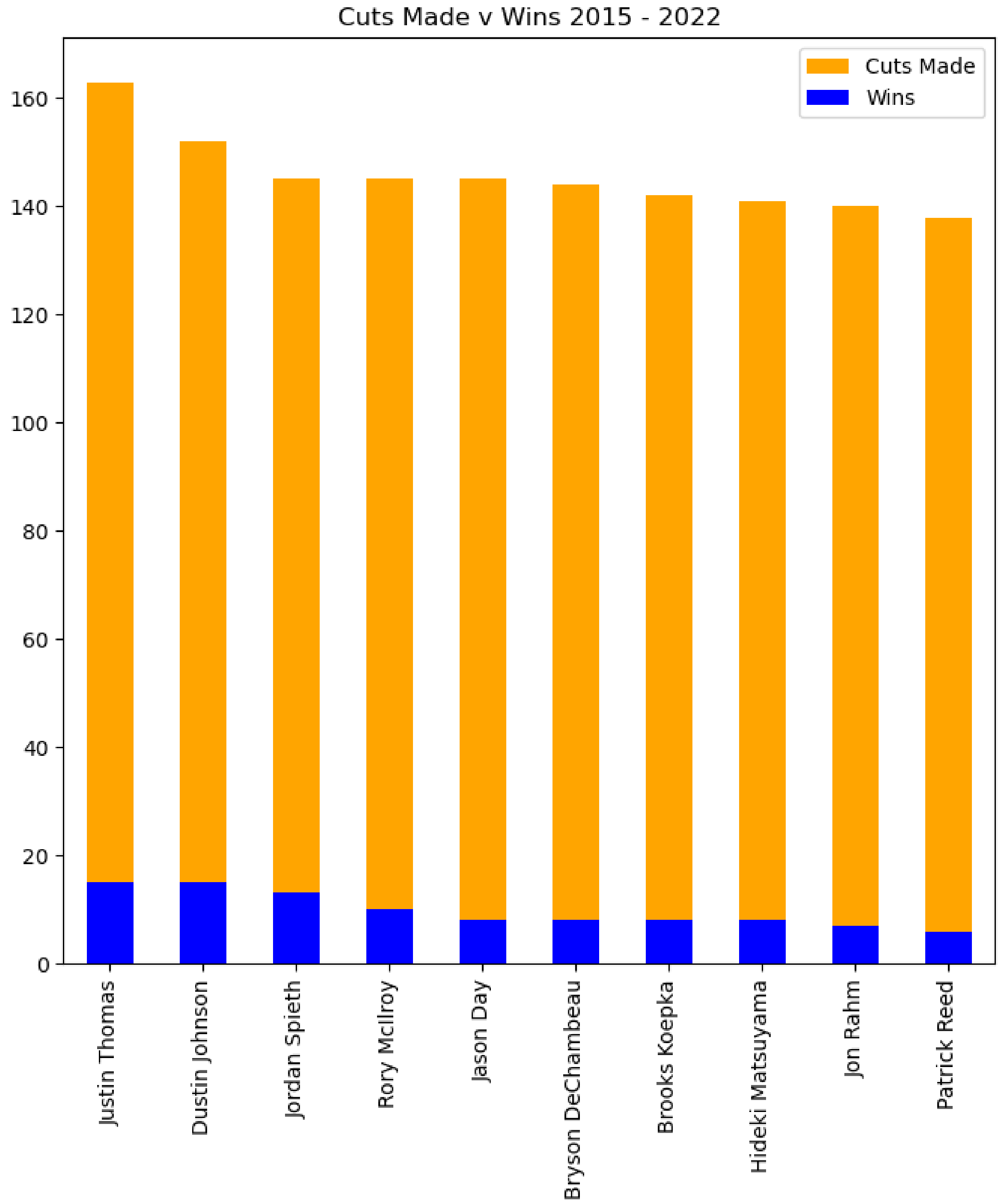
- For the last example of connecting we connected to random joke url, which generates a random joke each time its run, again we converted to json and printed the value field which contains the joke itself.

Results

***Graph 1** : Horizontal bar chart sort in descending order with player with most wins*



***Graph 2** : Grouped bar chart with 2 dimensions linked, wins and cuts made limited to top 10*



Graph 3 *: Horizontal bar chart with sorted in descending order with players with most missed cuts*

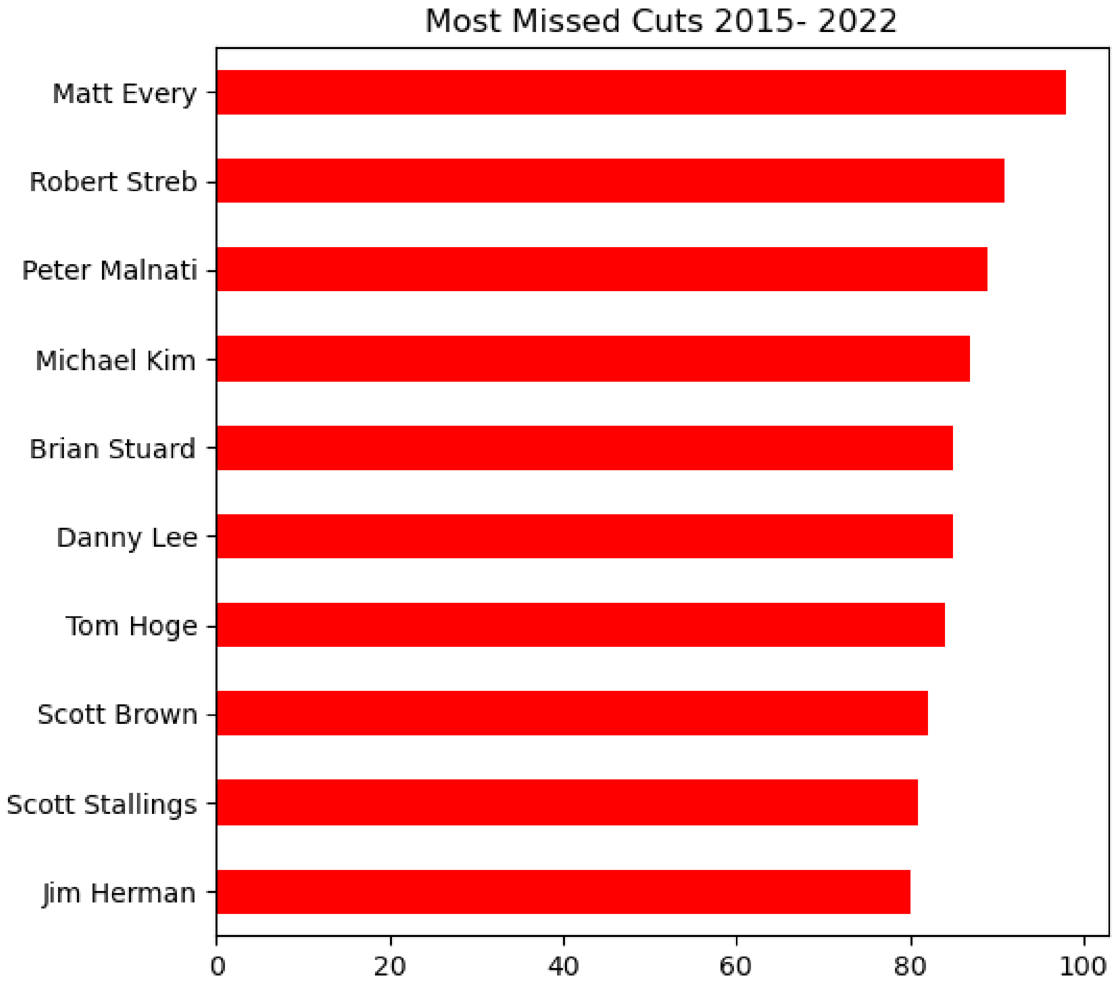
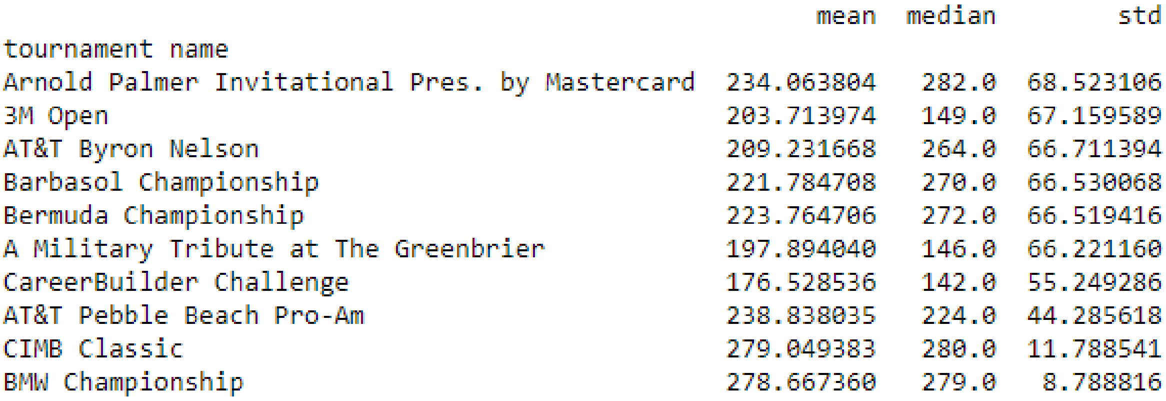
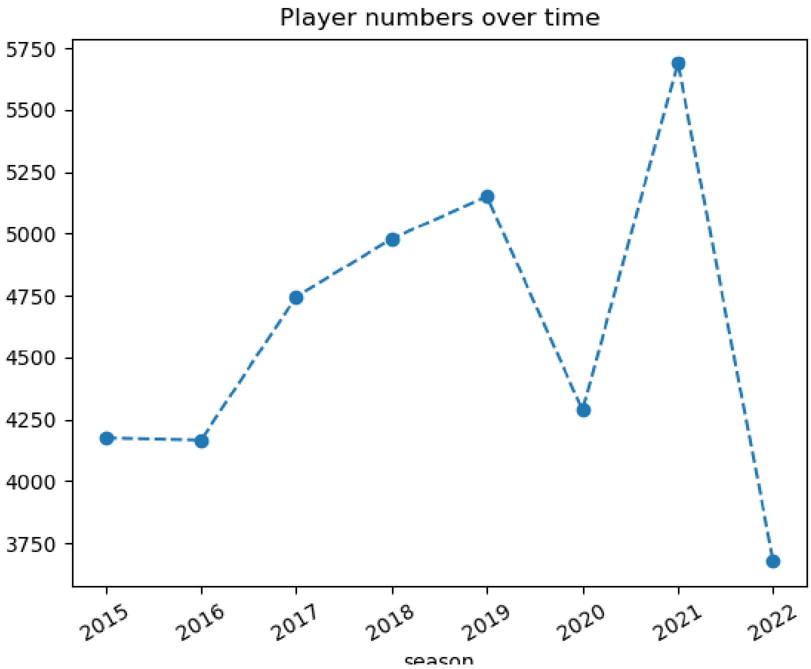


Table 1 *: Table for top 10 tournaments with aggregated stats, sorted by standard deviation*

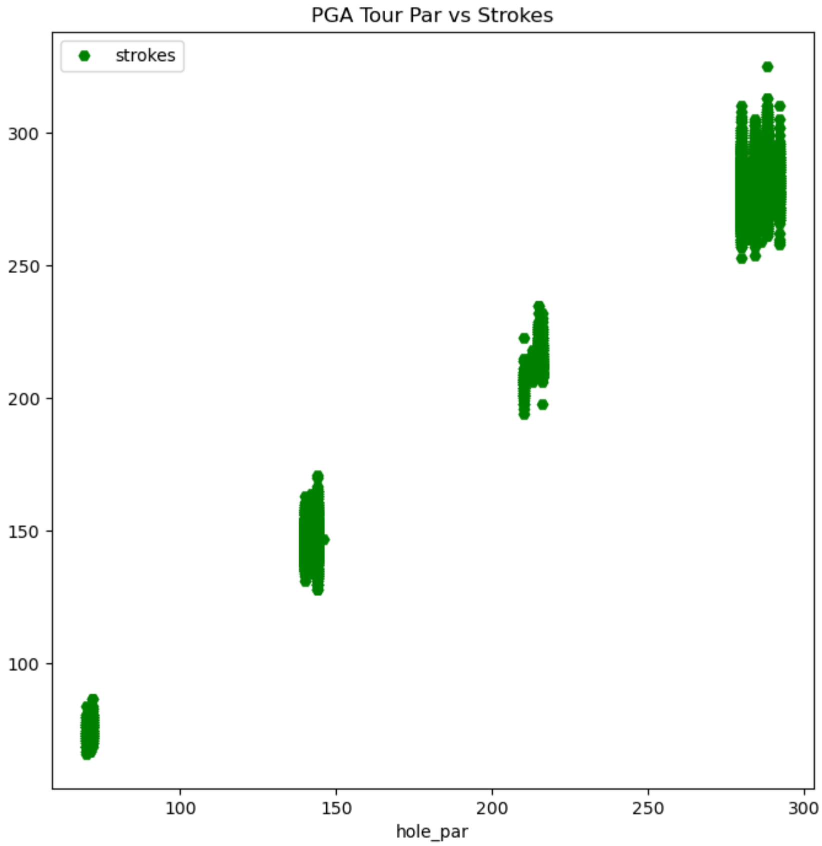


```
                                                  mean  median        std
tournament name
Arnold Palmer Invitational Pres. by Mastercard  234.063804   282.0  68.523106
3M Open                                         203.713974   149.0  67.159589
AT&T Byron Nelson                               209.231668   264.0  66.711394
Barbasol Championship                           221.784708   270.0  66.530068
Bermuda Championship                            223.764706   272.0  66.519416
A Military Tribute at The Greenbrier            197.894040   146.0  66.221160
CareerBuilder Challenge                         176.528536   142.0  55.249286
AT&T Pebble Beach Pro-Am                        238.838035   224.0  44.285618
CIMB Classic                                    279.049383   280.0  11.788541
BMW Championship                                278.667360   279.0   8.788816
```

***Graph 4** : Line plot for player appearances in tournaments v season, marker and linestyle changes, xticks rotated*



***Graph 5** : scatter plot for par strokes per player v actual strokes taken*



***Graph 6** : Kernel density plot for strokes against players to see where majority of players fall into*

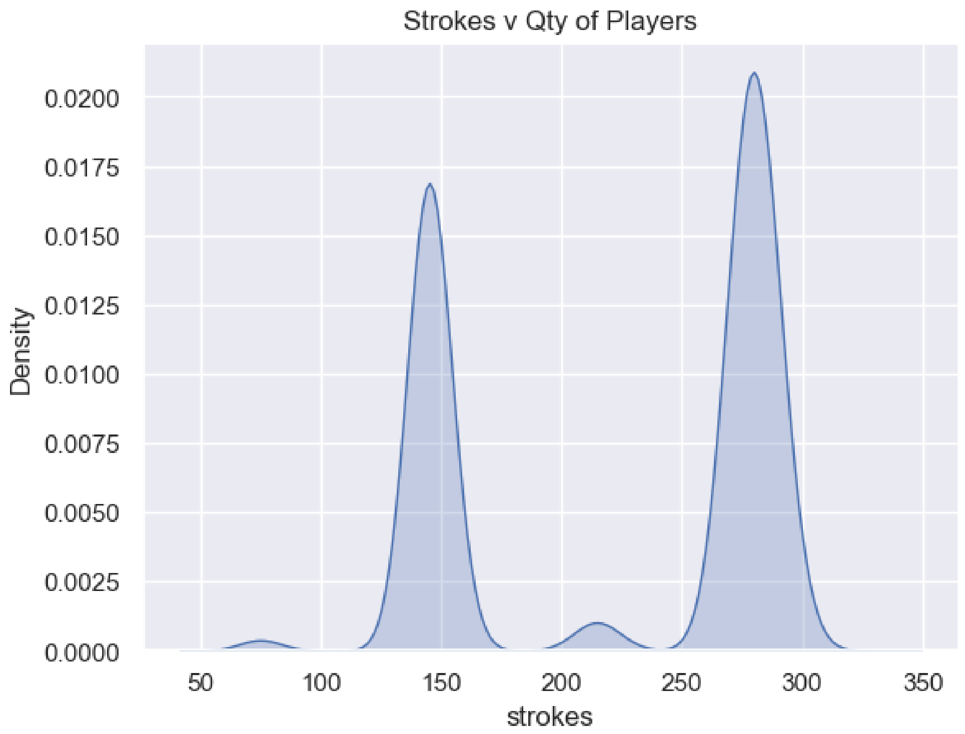
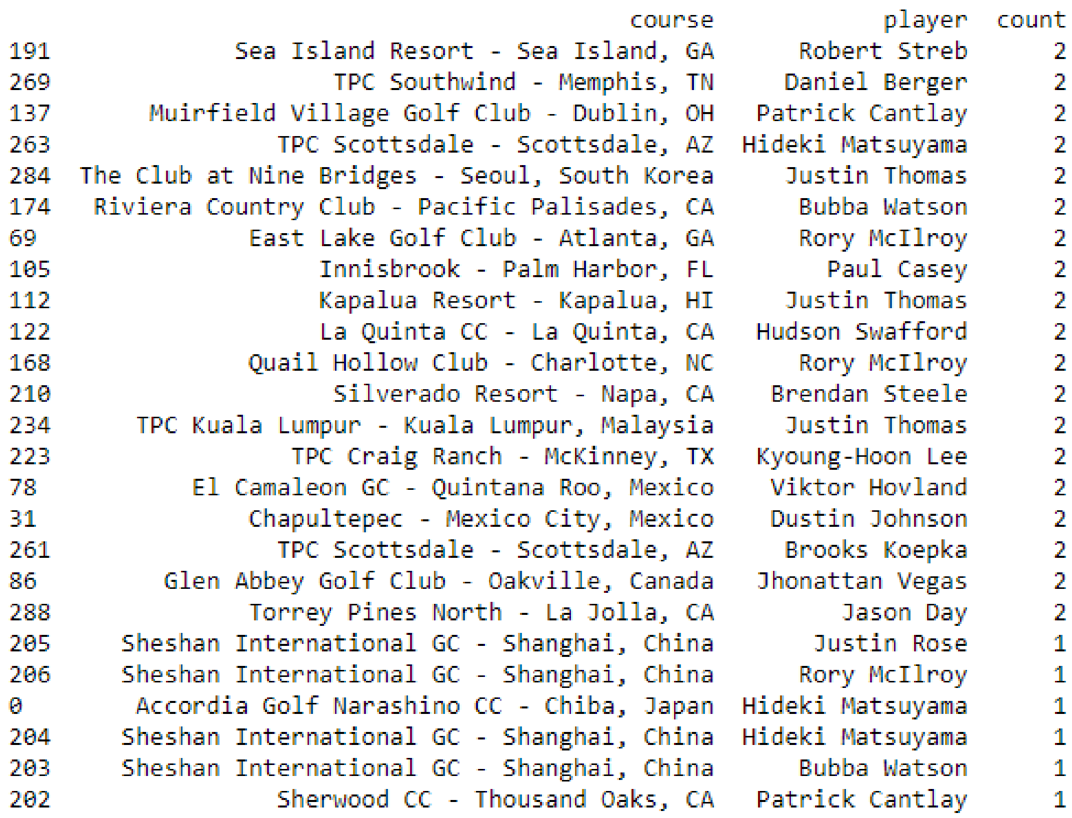
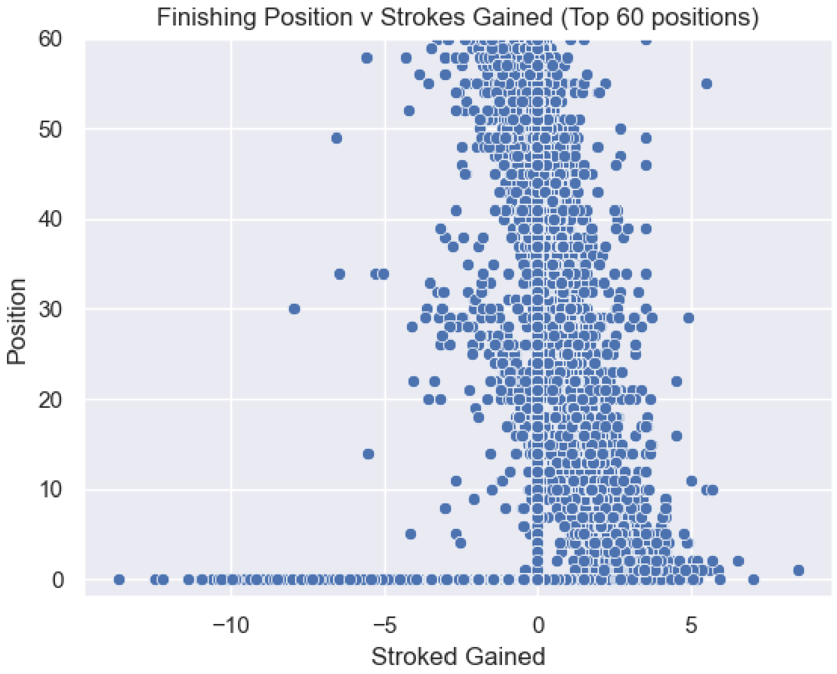


***Table 2** : Table for top 25 players and course and numbs of wins – sorted in descending order*

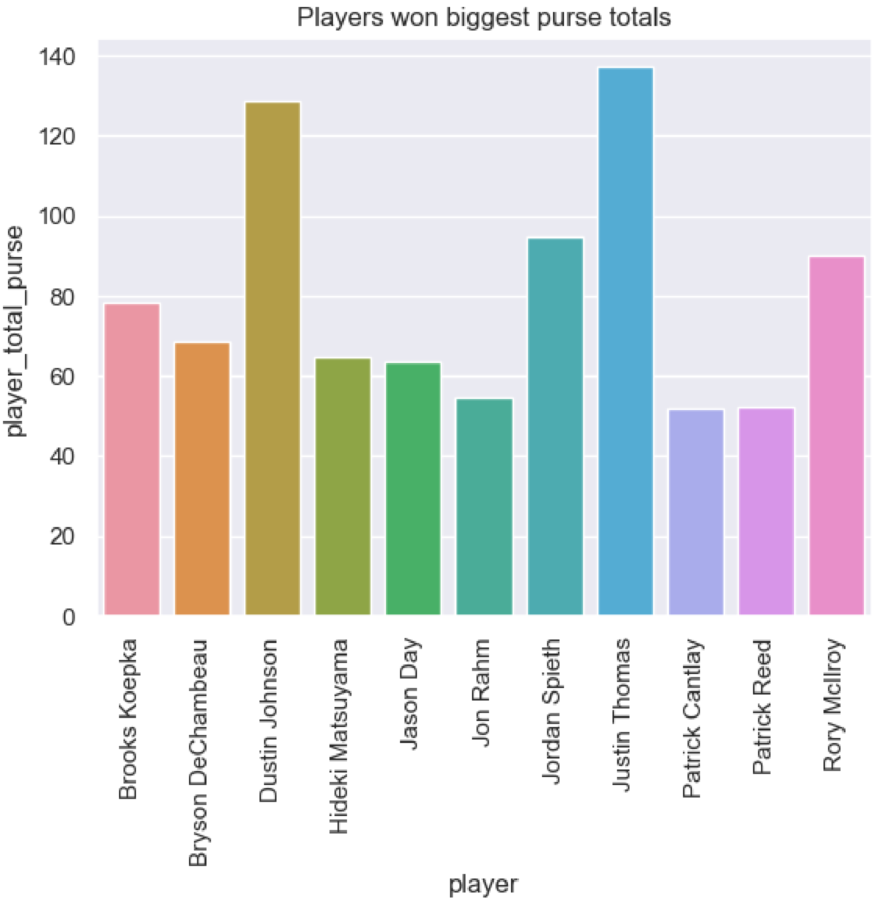


```
                                               course           player  count
191              Sea Island Resort - Sea Island, GA     Robert Streb      2
269                     TPC Southwind - Memphis, TN    Daniel Berger      2
137        Muirfield Village Golf Club - Dublin, OH  Patrick Cantlay      2
263                 TPC Scottsdale - Scottsdale, AZ  Hideki Matsuyama      2
284  The Club at Nine Bridges - Seoul, South Korea    Justin Thomas      2
174   Riviera Country Club - Pacific Palisades, CA     Bubba Watson      2
69                East Lake Golf Club - Atlanta, GA     Rory McIlroy      2
105                    Innisbrook - Palm Harbor, FL       Paul Casey      2
112                    Kapalua Resort - Kapalua, HI    Justin Thomas      2
122                    La Quinta CC - La Quinta, CA  Hudson Swafford      2
168               Quail Hollow Club - Charlotte, NC     Rory McIlroy      2
210                    Silverado Resort - Napa, CA   Brendan Steele      2
234        TPC Kuala Lumpur - Kuala Lumpur, Malaysia    Justin Thomas      2
223                 TPC Craig Ranch - McKinney, TX   Kyoung-Hoon Lee      2
78           El Camaleon GC - Quintana Roo, Mexico   Viktor Hovland      2
31                Chapultepec - Mexico City, Mexico   Dustin Johnson      2
261                 TPC Scottsdale - Scottsdale, AZ    Brooks Koepka      2
86          Glen Abbey Golf Club - Oakville, Canada  Jhonattan Vegas      2
288               Torrey Pines North - La Jolla, CA        Jason Day      2
205       Sheshan International GC - Shanghai, China      Justin Rose      1
206       Sheshan International GC - Shanghai, China     Rory McIlroy      1
0          Accordia Golf Narashino CC - Chiba, Japan  Hideki Matsuyama      1
204       Sheshan International GC - Shanghai, China  Hideki Matsuyama      1
203       Sheshan International GC - Shanghai, China     Bubba Watson      1
202                 Sherwood CC - Thousand Oaks, CA  Patrick Cantlay      1
```

***Graph 7** : Scatter plot on players for position finished v strokes gained, lower the position the more in the positive strokes gained*



***Graph 8** : Seaborn barchart for all players with purse winnings over 50*



Insights

Derived insights from this analysis:

- Justin Thomas has the most wins, largest purse winnings and most cuts made, while Dustin Johnson has joint most wins but 2nd in purse winnings
- Matt Every has missed the most cuts in tournaments over the time period
- Arnold Palmer Invitational tournament has the highest standard deviation at 68.52
- The Players Championship has the highest average purse over the period at 12.8m
- The highest number of player appearances at tournaments was in 2021 season
- We had 327 winners across the time period, with 19 players winning twice at the same course.
- Parker McLachlin has the highest average single round score of 87

References

https://www.kaggle.com/datasets/robikscube/pga-tour-golf-data-20152022

https://www.kaggle.com/datasets/bradklassen/official-world-golf-ranking-data

https://www.sqlitetutorial.net/sqlite-sample-database/

https://api.chucknorris.io/

www.kaggle.com. (n.d.). *Official World Golf Ranking Data*. [online] Available at: https://www.kaggle.com/datasets/bradklassen/official-world-golf-ranking-data [Accessed 1 Feb. 2023]

https://www.kaggle.com/datasets/robikscube/pga-tour-golf-data-20152022

www.kaggle.com. (n.d.). *PGA Tour Golf Data - (2015-2022)*. [online] Available at: https://www.kaggle.com/datasets/robikscube/pga-tour-golf-data-20152022 [Accessed 1 Feb. 2023].

https://www.kaggle.com/datasets/bradklassen/official-world-golf-ranking-data

SQLite Tutorial. (n.d.). *SQLite Sample Database And Its Diagram (in PDF format)*. [online] Available at: https://www.sqlitetutorial.net/sqlite-sample-database/.

Howdidido.com. (2019). *HowDidiDo*. [online] Available at: https://www.howdidido.com/ [Accessed 22 Apr. 2019].

www.golfireland.ie. (n.d.). *Removing barriers for women*. [online] Available at: https://www.golfireland.ie/.

https://www.facebook.com/PGATour (2019). *PGATOUR.COM - Official Home of Golf and the FedExCup*. [online] PGATour. Available at: https://www.pgatour.com/.

sqliteonline.com. (n.d.). *SQL Online Compiler - for Data Science*. [online] Available at: https://sqliteonline.com/