Observational Units:

I got all the data for my project from FBREF. The observational units for the data are individual players who participated in any of the top 5 leagues for the 2021-2022 season. The top 5 leagues are the Premier League (England), La Liga (Spain), Serie A (Italy), Bundesliga (Germany), and Ligue 1 (France). The variables for each player are their observed statistics for the 2021-2022 season. These observed statistics span many different facets of their game. The categories I decided to include were shooting, passing, defensive actions, possession, and creativity metrics. I joined this data to the basic information for each player such as their age, team, position, and minutes played. These variables are measured manually by observing each individual game and recording every action throughout the 90 minutes for each player. These are then compiled over the season for each player to provide a summary of the season they had.

Data Cleaning:

Since all the data for the different categories and leagues were stored in separate tables, I had to clean each table individually before merging and concatenating the data for the final table. Since I wanted to cover 5 leagues, I ended up using 30 separate tables to make the final dataset.

Standard Table Cleaning:

The first part of cleaning the standard tables included removing the unnecessary columns that were part of the pd.read_csv() statement. Next, I removed the per90 statistics since they all shared a name with the base stat counterpart. For example, the variable for goals was Gls, and goals per 90 was Gls.1. So I removed these since I could make them if I wanted by using the Gls and the 90s variable.

Shooting Cleaning:

I cleaned the shooting table by removing unnecessary columns when reading the data. I then wanted to have a way to specify that these variables all were related to shooting so I created a dictionary to rename the columns to shoot.var_name instead of just var_name. The for loop I made to do this only adds a shoot prefix if the column is not being used to join the data. This means I won't have to specify right_on and left_on and can still use the on argument when joining the data.

Passing Cleaning:

The passing table cleaning is almost the same as the shooting table. I wanted to add a pass prefix to the unique variables. The only difference was that I also had to change some duplicate variable names that were for short, medium, and long passing. For these variables I added pass.short., pass.med, pass.long respectively.

Defense Cleaning:

Cleaning the defense table was the same as the other tables. I dropped the unwanted variables and renamed the variables to have the defense prefix if not being used in the join. For the defense table, I also had to adjust a duplicated variable name to be more specific.

Possession Cleaning:

Possession cleaning had nothing special and was a repeat of the processes used on all the above tables.

Shot Creating Actions Cleaning:

I did the same process again. The only difference was having to specify a sca.G and sca.S to differentiate goal and shot-creating actions respectively.

Joining:

For each league, I joined the six tables to create a full table for each league. The full table included the standard, passing, shooting, defense, possession, and shot creation tables for each league. I then concatenated the full tables for each league vertically to obtain the final table with all categories and all leagues in the same table.

Model Filtering:

For my model, I decided to only look at non-Goalkeepers. Since goalkeepers need to be evaluated entirely on their own set of statistics, it did not make sense to include them in the model. I also limited the model to only having players who played at least 1000 minutes throughout the season.