What kind of cleaning steps did you perform?

1. 3 sets of data was acquired: NBA Salary (1985-2017) data from data.world, Inflation data from Federal Reserve Economic Data (FRED), and NBA Advanced Stats (1997-2017) data that was scraped from www.basketball-reference.com
    a. NBA Salary: This data set contained a unique identifier, player_id, that was associated to a salary and year. Not much cleaning was done on this data set.
    b. Inflation data: In this dataset, I wanted to get the average inflation value for the year instead of by day.
        i. I converted date to datetime in order to extract the year. Then the dataset was narrowed down to just Year and Inflation Unit (CPIAUCNS).
        ii. Next, I grouped the data by year to get a mean of CPIAUCNS.
        iii. After resetting the index, I created a value to multiply the salary dataset with to adjust for inflation.
        iv. I divided the CPIAUCNS for 2019 (inflation.iloc[-1,1]) by the CPIAUCNS for that year (inflation['CPIAUCNS']).
    c. Adjusting Salary for Inflation: I did a merge on salary and inflation dataset on salary season_end column to inflation year column. Salary was then multiplied by the inflation multiplier and rounded. This new dataset was then narrowed down to player_id, adj_salary, year, season and team.
    d. NBA Advanced Stats: BeautifulSoup was used to extract data from www.basketball-reference.com.
        i. To get a list of the urls to be scraped, I first identified that each url is the same with the exception of the year.
        ii. A list of years from 1997 to 2017 was created.
        iii. Then the URL was iterated with each year to get a list of URLs.
        iv. A function was defined for the scraping (urlScraping).
            1. Within this function, BeautifulSoup is used to open the url.
            2. Column headers, player_stats, player_id, and year were extracted.
            3. A DataFrame is created using headers as columns and player_stats as rows.
            4. There were blank columns in the DataFrame that were taken out using isspace().
            5. There were also rows with all NaN values that were removed as well.
            6. Extracted data player_id and year were added as columns.

7. DataFrame index was reset, then the index column was dropped.
8. Certain columns were converted from Object to Float/Int.

v. A for loop was created to run all url through urlScraping function to save as a csv with matching year.

vi. Another for loop was used to read_csv and append all data into an empty list where that list was concatenated to create one list of advanced stats for players from 1997-2017.

e. Advanced Stats joined with Adjusted Salary: A left join on advanced stats was done with adjusted salary on the player_id and year. Left join was used because I wanted to see how many players didn't have any salary in the salary dataset. Players without salary accounted for less than 5% of the total data, so they were excluded.

How did you deal with missing values, if any?

1. Missing values in the scraped datasets for Advanced Stats were removed becuse they were acting as spacers in html table.
2. In the merged advanced stats and adjusted salary, players with missing salaries were taken out from data set since they accounted for less than 5% of data.

Were there outliers, and how did you handle them?
There were no significant outliers.