Nba Data Project

File Started 03JAN2023

* Goal **: Analyze if Denver has an above normal home court advantage, presumably due to altitude.**

# Completed:

* + Have 4 seasons worth of advanced data , 4 seasons worth of traditional game data
    - ✅ Going to expand traditional games to 20 years till ‘02 (22Jan2023)
      * Problem ✅:
        + 10-11, SQL table has 09-10 data, as does the raw data folder
        + 12-13, SQL table has 13-14 data
      * Plan to handle above:
        + Remove data from trad\_all\_years table
        + Remove data from the problem tables
        + Check the other tables
        + Rerun python code and reupload
  + Get the data into sql.
    - ✅ Data is now in sql ready to explore (18JAN2023)
    - What teams have won the most home games
      * ✅ Denver is 4th, Utah is 3rd (interesting cause also a higher elevation stadium)
    - What teams have the highest proportion of wins at home (HomeWinProp)
      * Using home wins over total wins
        + Don't love this, seems to have little correlation with actually being good
      * RESULT:After gathering more data we can see, when we rank by HWP, Utah and Denver stand out in terms of total wins.
        + i.e. the few teams that are better than them by HWP are just bad and can only win at home (not the same as a home court advantage
        + When ranked by homewins, Utah and Denver appear to be outliers
      * **UPDATE**: Switched to difference between home and total win pct.
        + Have a plot in vizzies file
        + Might try to update to include the logos
        + This feels like the best update so far
  + Data verification : good, 1 game cancelled, other game was wizards hornets
* Whats next:

1. Incorporating advanced data
   * Have to find a way to truly, join data, right now it has an extra duplicate ✅
   * Need to add a column to each row : opponent, should be possible, using matchup column ✅
   * TOP PRIORITY : expanding advanced data to 02-03 if possible ✅
     1. Not going well erroring out way more frequently than last time ✅
     2. Need to look into errors ✅
   * Ranking net rating difference home v away
   * Difference in pace against DEN, UTA ?
2. Team by team record on the road v @ DEN, UTA ??? Still considering this
   * Pretty easy to get the numbers
   * Need to use pivot table
     1. Need to download tablefunc
     2. Or do it in python
   * Wondering how useful/trackable this actually is
     1. Would be a decent summary statistic, but all its really showing that theses teams (which we know are good) are harder to play @ than the average.
     2. Still think it could be useful but going to put it off for now
     3. Best way would be a 30 by 30 pivot table with the teams win percent @ each location
3. Using Hypothesis testing
4. Advanced data analysis
   * Net rating based:
     1. What team has the best net rating at home ✅
     2. Best net rating at home vs overall net rating, home vs away ✅
     3. Opponent net rating v opponent avg net rating for home teams
        1. Probably skip, would need to calc net rating by season, then add that to each team calc the difference
   * Pace per 40 analysis
     1. Do opponents play slower in Den and Utah
        1. Not looking good for opponents playing slower in denver
     2. Do Den and Utah play fast on the road

* Findings:

1. DEN and UTA rank very high by total home wins / home win %
   * Only teams above them in total wins/hw% are the teams that are just better overall (SAS , DAL ; 1 and 2 respectively in homewins and total wins)
   * HWP (the proportion of wins at home / total wins) shows DEN and UTA as pretty stark outliers to the trend. Trend is best HWP teams tend to be just bad teams who don’t win a lot but when they do it happens to be at home.

***#### Useful SQL Queries ####***

**To get number of home wins**

SELECT team\_abbreviation AS TEAM, count(\*) AS HomeWins

FROM trad\_all\_years

WHERE home\_away = 'HOME' AND wl = 'W'

GROUP BY team\_abbreviation

ORDER BY 2 DESC, team\_abbreviation;

**Total wins, home wins, away wins, proportion of wins at home / totalwins**

WITH homewins AS (

SELECT team\_abbreviation, count(\*) AS h

FROM trad\_all\_years

WHERE home\_away = 'HOME' AND wl = 'W'

GROUP BY team\_abbreviation),

totalwins AS(

SELECT team\_abbreviation, count(\*) AS t

FROM trad\_all\_years

WHERE wl = 'W'

GROUP BY team\_abbreviation

)

SELECT totalwins.team\_abbreviation, homewins.h AS homewins, totalwins.t - homewins.h as awaywins ,

totalwins.t AS totalwins, ROUND((CAST(homewins.h as decimal)/ totalwins.t),3) AS proportion

FROM totalwins

LEFT JOIN homewins

ON homewins.team\_abbreviation = totalwins.team\_abbreviation

GROUP BY totalwins.team\_abbreviation, homewins.h, totalwins.t

ORDER BY 5 DESC, team\_abbreviation;

**Copying into a already made table :**

COPY advanced\_18\_19

FROM '/Volumes/GoogleDrive/My Drive/updated\_data/raw\_data/advanced\_18\_19.csv'

DELIMITER ','

CSV HEADER;

**Creating a table:**

CREATE TABLE traditional\_17\_18 (

SEASON\_ID VARCHAR(10),

TEAM\_ID VARCHAR(12),

TEAM\_ABBREVIATION VARCHAR(3),

TEAM\_NAME VARCHAR(50),

GAME\_ID INT,

GAME\_DATE DATE,

MATCHUP VARCHAR(15),

WL CHAR(1),

MIN INT,

FGM INT,

FGA INT,

FG\_PCT NUMERIC,

FG3M INT,

FG3A INT,

FG3\_PCT NUMERIC,

FTM INT,

FTA INT,

FT\_PCT NUMERIC,

OREB INT,

DREB INT,

REB INT,

AST INT,

STL INT,

BLK INT,

TOV INT,

PF INT,

PTS INT,

PLUS\_MINUS INT,

VIDEO\_AVAILABLE INT,

HOME\_AWAY VARCHAR(4)

)

WITH homewins AS (

SELECT team\_abbreviation, count(\*) AS h

FROM trad\_all\_years

WHERE home\_away = 'HOME' AND wl = 'W'

GROUP BY team\_abbreviation),

totalwins AS(

SELECT team\_abbreviation, count(\*) AS t

FROM trad\_all\_years

WHERE wl = 'W'

GROUP BY team\_abbreviation

),

homelosses AS (

SELECT team\_abbreviation, count(\*) AS l

FROM trad\_all\_years

WHERE home\_away = 'HOME' AND wl = 'L'

GROUP BY team\_abbreviation),

totallosses AS(

SELECT team\_abbreviation, count(\*) AS tl

FROM trad\_all\_years

WHERE wl = 'L'

GROUP BY team\_abbreviation

)

SELECT totalwins.team\_abbreviation, homewins.h AS homewins, totalwins.t - homewins.h as awaywins ,

totalwins.t AS totalwins, ROUND((CAST(homewins.h as decimal)/ totalwins.t),3) AS proportion,

RANK() OVER(order by ROUND((CAST(homewins.h as decimal)/ totalwins.t),3) DESC) AS HWP\_Rank,

homelosses.l AS home\_losses, totallosses.tl - homelosses.l as away\_losses , totallosses.tl AS total\_losses,

RANK() OVER(order by totallosses.tl DESC) AS loss\_rank

FROM totalwins

LEFT JOIN homewins

USING(team\_abbreviation)

LEFT JOIN homelosses

USING(team\_abbreviation)

LEFT JOIN totallosses

USING(team\_abbreviation)

WHERE totalwins.t + totallosses.tl > 600

**Join between trad and adv, limits double entries:**

SELECT \*

FROM traditional\_18\_19 AS t

LEFT JOIN advanced\_18\_19 AS a

ON t.game\_id = a.game\_id AND t.team\_abbreviation = a.team\_abbreviation

ORDER BY t.game\_id, home\_away

LIMIT 10 ;

**Difference between Home and total:**

WITH homewins AS (

SELECT team\_abbreviation, count(\*) AS h

FROM trad\_all\_years

WHERE home\_away = 'HOME' AND wl = 'W'

GROUP BY team\_abbreviation),

totalwins AS(

SELECT team\_abbreviation, count(\*) AS t

FROM trad\_all\_years

WHERE wl = 'W'

GROUP BY team\_abbreviation

),

homelosses AS (

SELECT team\_abbreviation, count(\*) AS l

FROM trad\_all\_years

WHERE home\_away = 'HOME' AND wl = 'L'

GROUP BY team\_abbreviation),

totallosses AS(

SELECT team\_abbreviation, count(\*) AS tl

FROM trad\_all\_years

WHERE wl = 'L'

GROUP BY team\_abbreviation

)

SELECT totalwins.team\_abbreviation, ROUND((CAST(homewins.h AS decimal)/(homewins.h +homelosses.l)),3) AS homewinpct,

ROUND((CAST(totalwins.t AS decimal)/(totalwins.t +totallosses.tl)),3) AS totalwinpct,

(ROUND((CAST(homewins.h AS decimal)/(homewins.h +homelosses.l)),3) -

ROUND((CAST(totalwins.t AS decimal)/(totalwins.t +totallosses.tl)),3)) AS diff,

RANK() OVER(ORDER BY ROUND((CAST(totalwins.t AS decimal)/(totalwins.t +totallosses.tl)),3) DESC) AS winpct\_rank

FROM totalwins

LEFT JOIN homewins

USING(team\_abbreviation)

LEFT JOIN homelosses

USING(team\_abbreviation)

LEFT JOIN totallosses

USING(team\_abbreviation)

WHERE totalwins.t + totallosses.tl > 600

ORDER BY 4 DESC

**GET NET RATING DIFFERENTIAL HOME AWAY:**

WITH home\_net as (

SELECT team\_abbreviation, ROUND(SUM(net\_rating)/count(\*),3) as home\_net

FROM master\_table

WHERE home\_away = 'HOME'

GROUP BY team\_abbreviation

ORDER BY 2 DESC),

away\_net AS(

SELECT team\_abbreviation, ROUND(SUM(net\_rating)/count(\*),3) as away\_net

FROM master\_table

WHERE home\_away = 'AWAY'

GROUP BY team\_abbreviation

ORDER BY 2 DESC

),

overall\_net AS(

SELECT team\_abbreviation, ROUND(SUM(net\_rating)/count(\*),3) as overall\_net

FROM master\_table

GROUP BY team\_abbreviation

ORDER BY 2 DESC

)

SELECT home\_net.team\_abbreviation, home\_net- away\_net as h\_a\_diff,

home\_net - overall\_net AS h\_o\_diff

FROM home\_net

LEFT JOIN away\_net

USING (team\_abbreviation)

LEFT JOIN overall\_net

USING (team\_abbreviation)

ORDER BY 3 DESC

\*\*If running on the Windows\*\*: path type is pd.read\_csv(r"C:\Users\Jo\Google Drive\updated\_data\raw\_data/hometotaldiff.csv")

\*\*If running on the Mac\*\*: all\_years.to\_csv(r"/Volumes/GoogleDrive/My Drive/updated\_data/raw\_data/trad\_all\_years\_opp.csv", index=False)

# Initializing our all\_games variable

all\_games = []

# This loop

for count ,i in enumerate(games\_list):

print(count,'game\_id : ', i, ', Percent Done: ', round(((count+1)/len(games\_list))\*100,2) , '%')

player\_stats\_data = boxscoreadvancedv2.BoxScoreAdvancedV2(game\_id= i, headers=headers, timeout=100)

stats\_df = player\_stats\_data.get\_data\_frames()[1]

all\_games.append(stats\_df)

time.sleep(.600)

#run after above loop completes

all\_games=pd.concat(all\_games)