MarchMadness

```
library(readr)
library(dplyr)
```

```
Attaching package: 'dplyr'

The following objects are masked from 'package:stats':

    filter, lag

The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union
```

```
library(tidyr)
```

```
files <- list.files("March Madness Regular Season Data", pattern = "\\.csv$", full.names = TR


regular_season_df <- files |>
  lapply(function(f) read_csv(f, col_types = cols(.default = "c"))) |>
  bind_rows()
```

```
getwd()
```

```
[1] "/Users/ananyadas/sta345-final"
```

```
list.files()
```

```
[1] "March Madness Regular Season Data"
[2] "MarchMadness.pdf"
[3] "MarchMadness.qmd"
[4] "MarchMadness.rmarkdown"
[5] "matchup_statistics_all.csv"
[6] "matchup_statistics_complete.csv"
[7] "matchup_statistics.csv"
[8] "sta345-final.Rproj"
[9] "tourney_results_mod_through_2025.csv"
```

```
list.files("March Madness Regular Season Data")
```

```
 [1] "ncaa_tr_regular_2000_2001.csv" "ncaa_tr_regular_2002_2003.csv"
 [3] "ncaa_tr_regular_2004_2005.csv" "ncaa_tr_regular_2006_2007.csv"
 [5] "ncaa_tr_regular_2008_2009.csv" "ncaa_tr_regular_2010_2011.csv"
 [7] "ncaa_tr_regular_2010_2025.csv" "ncaa_tr_regular_2012_2013.csv"
 [9] "ncaa_tr_regular_2014_2015.csv" "ncaa_tr_regular_2016_2017.csv"
[11] "ncaa_tr_regular_2018_2019.csv" "ncaa_tr_regular_2020_2021.csv"
[13] "ncaa_tr_regular_2022_2023.csv" "ncaa_tr_regular_2024_2025.csv"
```

```
tourney_result <- read_csv("tourney_results_mod_through_2025.csv")
```

```
Rows: 2585 Columns: 15
-- Column specification --------------------------------------------------------
Delimiter: ","
chr  (5): wloc, wteam_school, lteam_school, wteam_region, lteam_region
dbl (10): season, daynum, wteam, wscore, lteam, lscore, numot, wteam_seed, l...

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
lower_better <- c(
  # Negative outcomes (your team)
  "Turnovers per Game",
  "Turnovers per Possession",
  "Turnovers per Offensive Play",
  "Personal Fouls per Game",
```

```
  "Personal Fouls per Possession",
  "Personal Fouls per Defensive Play",

  # Opponent stats - you want *less* of these
  "Opponent Points per Game",
  "Opponent Average Scoring Margin",
  "Opponent Floor %",
  "Opponent 1st Half Points per Game",
  "Opponent 2nd Half Points per Game",
  "Opponent Overtime Points per Game",
  "Opponent Points from 2 pointers",
  "Opponent Points from 3 pointers",
  "Opponent Shooting %",
  "Opponent Effective Field Goal %",
  "Opponent Three Point %",
  "Opponent Two Point %",
  "Opponent Free Throw %",
  "Opponent True Shooting %",
  "Opponent Field Goals Made per Game",
  "Opponent Three Pointers Made per Game",
  "Opponent Free Throws Made per Game",
  "Opponent Non-blocked 2 Pt %",
  "Opponent Offensive Rebounds per Game",
  "Opponent Offensive Rebounding %",
  "Opponent Assists per Game",
  "Opponent Assists per FGM",
  "Opponent Assists per Possession",
  "Opponent Assist / Turnover Ratio",
  "Opponent Win % - All Games",
  "Opponent Win % - Close Games",
  "Opponent Effective Possession Ratio",

  # Defensive efficiency (points allowed per 100 poss)
  "Defensive Efficiency",

  # Mixed/negative:
  # Opponent Defensive Rebounding % is bad for you
  "Opponent Defensive Rebounding %"
)
lower_better[!lower_better %in% names(regular_season_df)]
```

```
character(0)
```

```r
regular_season_unique <- regular_season_df |>
  distinct(school, year, .keep_all = TRUE)

stat_cols <- regular_season_unique |>
  select(-school, -year) |>
  names()
```

```r
team_season_ranks <- regular_season_unique |>
  group_by(year) |>
  mutate(
    # Stats where higher is better
    across(
      all_of(setdiff(stat_cols, lower_better)),
      ~ {
        x_num <- parse_number(as.character(.))
        rank(-x_num, ties.method = "average")
      }
    ),

    # Stats where lower is better
    across(
      all_of(lower_better),
      ~ {
        x_num <- parse_number(as.character(.))
        rank(x_num, ties.method = "average")
      }
    )
  ) |>
  ungroup()
```

```
Warning: There were 470 warnings in `mutate()`.
The first warning was:
i In argument: `across(...)`.
i In group 22: `year = "2021"`.
Caused by warning:
! 10 parsing failures.
row col expected actual
348  -- a number     --
349  -- a number     --
350  -- a number     --
351  -- a number     --
352  -- a number     --
```

```
... ... ........ ......
See problems(...) for more details.
i Run `dplyr::last_dplyr_warnings()` to see the 469 remaining warnings.
```

```
regular_season_unique %>%
  count(school, year) %>%
  filter(n > 1) %>%
  arrange(desc(n))
```

```
# A tibble: 0 x 3
# i 3 variables: school <chr>, year <chr>, n <int>
```

round of 64 games we are looking at

```
top_vs_bottom_round64 <- tourney_result |>
  mutate(
    low_seed  = pmin(wteam_seed, lteam_seed),
    high_seed = pmax(wteam_seed, lteam_seed)
  ) |>
  filter(
    round == 64,
    (low_seed == 1 & high_seed == 16) |
    (low_seed == 2 & high_seed == 15) |
    (low_seed == 3 & high_seed == 14) |
    (low_seed == 4 & high_seed == 13)
  )
```

```
library(tidyr)
# Check team name matches
teams_in_tourney <- top_vs_bottom_round64 |>
  select(wteam_school, lteam_school) |>
  pivot_longer(everything(), values_to = "team") |>
  distinct(team) |>
  pull(team)

teams_in_rankings <- team_season_ranks |>
  distinct(school) |>
  pull(school)

# Find tournament teams not in rankings
missing_from_rankings <- setdiff(teams_in_tourney, teams_in_rankings)
print("Teams in tournament but not in rankings:")
```

```
[1] "Teams in tournament but not in rankings:"
```

```
print(missing_from_rankings)
```

```
 [1] "Loyola-Chi"     "N Carolina"     "Ohio State"     "Iowa State"
 [5] "St Johns"       "GA Tech"        "Northeastrn"    "U Penn"
 [9] "Miss Val St"    "Ball State"     "AR Lit Rock"    "LA Monroe"
[13] "TX Christian"   "Central Mich"   "Idaho State"    "LA Tech"
[17] "GA Southern"    "Boise State"    "North Texas"    "TX-San Ant"
[21] "Rob Morris"     "S Car State"    "McNeese St"     "E Tenn St"
[25] "Geo Mason"      "Connecticut"    "S Mississippi"  "TX Southern"
[29] "Coppin State"   "St Fran (PA)"   "Penn State"     "Miami (OH)"
[33] "U Mass"         "LA Lafayette"   "TN State"       "Wright State"
[37] "James Mad"      "Texas State"    "Central FL"     "Hawaii"
[41] "Loyola-MD"      "N Mex State"    "Mt St Marys"    "Nicholls St"
[45] "St Peters"      "WI-Grn Bay"     "Lg Beach St"    "Weber State"
[49] "NC-Grnsboro"    "St Josephs"     "Charl South"    "S Carolina"
[53] "St Marys"       "Utah State"     "San Fransco"    "Col Charlestn"
[57] "Alcorn State"   "Miami (FL)"     "Central Conn"   "SE Missouri"
[61] "NC-Wilmgton"    "Boston Col"     "Kent State"     "Cal St Nrdge"
[65] "Fla Atlantic"   "Miss State"     "IL-Chicago"     "Sam Hous St"
[69] "IUPUI"          "NC-Asheville"   "TX El Paso"     "E Washingtn"
[73] "Wash State"     "TX A&M-CC"      "Maryland BC"    "TX-Arlington"
[77] "Ste F Austin"   "Ark Pine Bl"    "W Virginia"     "St Bonavent"
[81] "Detroit"        "Fla Gulf Cst"   "WI-Milwkee"     "Abilene Chr"
[85] "Gardner Webb"   "VA Tech"        "Kennesaw"       "UNC Wilmington"
```

```
# Find which years these teams appear in
top_vs_bottom_round64 |>
  filter(wteam_school %in% missing_from_rankings | lteam_school %in% missing_from_rankings) |>
  select(season, wteam_school, lteam_school) |>
  arrange(season)
```

```
# A tibble: 278 x 3
   season wteam_school lteam_school
    <dbl> <chr>        <chr>
 1   1985 Loyola-Chi   Iona
 2   1985 N Carolina   Middle Tenn
 3   1985 Ohio State   Iowa State
 4   1985 St Johns     Southern
 5   1985 GA Tech      Mercer
```

```
 6   1985 Illinois     Northeastrn
 7   1985 Memphis      U Penn
 8   1986 Duke         Miss Val St
 9   1986 GA Tech      Marist
10   1986 Memphis      Ball State
# i 268 more rows
```

```r
library(tidyr)

# Updated name mapping with correct ranking names
name_fixes <- tribble(
  ~tourney_name, ~ranking_name,
  "Loyola-Chi", "Loyola Chi",
  "N Carolina", "North Carolina",
  "Ohio State", "Ohio St",
  "Iowa State", "Iowa St",
  "St Johns", "St John's",
  "GA Tech", "Georgia Tech",
  "Northeastrn", "Northeastern",
  "U Penn", "Penn",
  "Miss Val St", "Miss Valley St",
  "Ball State", "Ball St",
  "AR Lit Rock", "Little Rock",
  "LA Monroe", "UL Monroe",
  "TX Christian", "TCU",
  "Central Mich", "C Michigan",
  "Idaho State", "Idaho St",
  "LA Tech", "Louisiana Tech",
  "GA Southern", "Georgia So",
  "Boise State", "Boise St",
  "North Texas", "N Texas",
  "TX-San Ant", "UTSA",
  "Rob Morris", "Robert Morris",
  "S Car State", "S Carolina St",
  "McNeese St", "McNeese",
  "E Tenn St", "E Tennessee St",
  "Geo Mason", "George Mason",
  "Connecticut", "UConn",
  "S Mississippi", "Southern Miss",
  "TX Southern", "Texas So",
  "Coppin State", "Coppin St",
  "St Fran (PA)", "St Francis PA",
```

```
"Penn State", "Penn St",
"Miami (OH)", "Miami OH",
"U Mass", "UMass",
"LA Lafayette", "Louisiana",
"TN State", "Tennessee St",
"Wright State", "Wright St",
"James Mad", "J Madison",
"Texas State", "Texas St",
"Central FL", "UCF",
"Hawaii", "Hawai'i",
"Loyola-MD", "Loyola MD",
"N Mex State", "New Mexico St",
"Mt St Marys", "Mt St Mary's",
"Nicholls St", "Nicholls",
"St Peters", "Saint Peter's",
"WI-Grn Bay", "Green Bay",
"Lg Beach St", "Long Beach St",
"Weber State", "Weber St",
"NC-Grnsboro", "NC Greensboro",
"St Josephs", "Saint Joseph's",
"Charl South", "Charleston So",
"S Carolina", "South Carolina",
"St Marys", "Saint Mary's",
"Utah State", "Utah St",
"San Fransco", "San Francisco",
"Col Charlestn", "Charleston",
"Alcorn State", "Alcorn St",
"Miami (FL)", "Miami",
"Central Conn", "C Connecticut",
"SE Missouri", "SE Missouri St",
"NC-Wilmgton", "NC Wilmington",
"Boston Col", "Boston College",
"Kent State", "Kent St",
"Cal St Nrdge", "CS Northridge",
"Fla Atlantic", "Florida Atlantic",
"Miss State", "Mississippi St",
"IL-Chicago", "Illinois Chicago",
"Sam Hous St", "Sam Houston",
"IUPUI", "IU Indy",
"NC-Asheville", "NC Asheville",
"TX El Paso", "UTEP",
"E Washingtn", "E Washington",
```

```
  "Wash State", "Washington St",
  "TX A&M-CC", "Texas A&M-CC",
  "Maryland BC", "UMBC",
  "TX-Arlington", "UT Arlington",
  "Ste F Austin", "SF Austin",
  "Ark Pine Bl", "AR-Pine Bluff",
  "W Virginia", "West Virginia",
  "St Bonavent", "St Bonaventure",
  "Detroit", "Detroit Mercy",
  "Fla Gulf Cst", "FGCU",
  "WI-Milwkee", "Milwaukee",
  "Abilene Chr", "Abl Christian",
  "Gardner Webb", "Gardner-Webb",
  "VA Tech", "Virginia Tech",
  "Kennesaw", "Kennesaw St",
  "UNC Wilmington", "NC Wilmington"
)

# Apply the fixes
top_vs_bottom_round64_fixed <- top_vs_bottom_round64 |>
  left_join(name_fixes, by = c("wteam_school" = "tourney_name")) |>
  mutate(wteam_school = coalesce(ranking_name, wteam_school)) |>
  select(-ranking_name) |>
  left_join(name_fixes, by = c("lteam_school" = "tourney_name")) |>
  mutate(lteam_school = coalesce(ranking_name, lteam_school)) |>
  select(-ranking_name)

# Verify the fixes worked
teams_in_tourney_fixed <- unique(c(
  top_vs_bottom_round64_fixed$wteam_school,
  top_vs_bottom_round64_fixed$lteam_school
))

missing_after_fix <- setdiff(teams_in_tourney_fixed, teams_in_rankings)
print("Teams still missing after fixes:")
```

```
[1] "Teams still missing after fixes:"
```

```
print(missing_after_fix)
```

```
character(0)
```

```r
# Also check if there are still mismatches
print(paste("Number of mismatches remaining:", length(missing_after_fix)))
```

```
[1] "Number of mismatches remaining: 0"
```

```r
# Filter to only years with available data (2000+)
top_vs_bottom_round64_filtered <- top_vs_bottom_round64_fixed |>
  filter(season >= 2000)

# Convert year to numeric in team_season_ranks
team_season_ranks <- team_season_ranks |>
  mutate(year = as.numeric(year))

# Now create matchup statistics with the filtered data
matchup_stats <- top_vs_bottom_round64_filtered |>
  left_join(
    team_season_ranks,
    by = c("wteam_school" = "school", "season" = "year")
  ) |>
  left_join(
    team_season_ranks,
    by = c("lteam_school" = "school", "season" = "year"),
    suffix = c("_winner", "_loser")
  )

# Identify which team is lower-seeded and which is higher-seeded
matchup_stats <- matchup_stats |>
  mutate(
    is_upset = (wteam_seed > lteam_seed),
    lower_seed_team = if_else(wteam_seed > lteam_seed, "winner", "loser"),
    higher_seed_team = if_else(wteam_seed < lteam_seed, "winner", "loser")
  )

# Calculate matchup statistics (lower seed rank - higher seed rank)
# Get all stat column names
stat_cols_winner <- grep("_winner$", names(matchup_stats), value = TRUE)

# Remove the suffixes to get base stat names
base_stat_names <- sub("_winner$", "", stat_cols_winner)

# Create matchup stat columns
```

```r
for (stat in base_stat_names) {
  winner_col <- paste0(stat, "_winner")
  loser_col <- paste0(stat, "_loser")
  matchup_col <- paste0("matchup_", stat)

  matchup_stats <- matchup_stats |>
    mutate(
      !!matchup_col := if_else(
        lower_seed_team == "winner",
        as.numeric(!!sym(winner_col)) - as.numeric(!!sym(loser_col)),
        as.numeric(!!sym(loser_col)) - as.numeric(!!sym(winner_col))
      )
    )
}

# Select relevant columns for analysis
matchup_stats_clean <- matchup_stats |>
  select(
    season,
    wteam_school,
    lteam_school,
    wteam_seed,
    lteam_seed,
    is_upset,
    round,
    starts_with("matchup_")
  )

# View the result
head(matchup_stats_clean)
```

```
# A tibble: 6 x 122
  season wteam_school lteam_school   wteam_seed lteam_seed is_upset round
   <dbl> <chr>        <chr>               <dbl>      <dbl> <lgl>    <dbl>
1   2000 Arizona      Jackson St              1         16 FALSE       64
2   2000 Iowa St      C Connecticut           2         15 FALSE       64
3   2000 LSU          SE Missouri St          4         13 FALSE       64
4   2000 Maryland     Iona                    3         14 FALSE       64
5   2000 Michigan St  Valparaiso              1         16 FALSE       64
6   2000 Oklahoma     Winthrop                3         14 FALSE       64
# i 115 more variables: `matchup_Points per Game` <dbl>,
#   `matchup_Average Scoring Margin` <dbl>,
```

```
#   `matchup_Offensive Efficiency` <dbl>, `matchup_Floor %` <dbl>,
#   `matchup_1st Half Points per Game` <dbl>,
#   `matchup_2nd Half Points per Game` <dbl>,
#   `matchup_Overtime Points per Game` <dbl>,
#   `matchup_Average 1st Half Margin` <dbl>, ...
```

```
# Check for any missing values
missing_summary <- matchup_stats_clean |>
  summarise(across(starts_with("matchup_"), ~sum(is.na(.))))

print("Missing values per matchup statistic:")
```

```
[1] "Missing values per matchup statistic:"
```

```
print(t(missing_summary))
```

```
                                                    [,1]
matchup_Points per Game                                0
matchup_Average Scoring Margin                         0
matchup_Offensive Efficiency                           0
matchup_Floor %                                        0
matchup_1st Half Points per Game                       0
matchup_2nd Half Points per Game                       0
matchup_Overtime Points per Game                       0
matchup_Average 1st Half Margin                        0
matchup_Average 2nd Half Margin                        0
matchup_Average Overtime Margin                        0
matchup_Points from 2 pointers                         0
matchup_Points from 3 pointers                         0
matchup_Percent of Points from 2 Pointers              0
matchup_Percent of Points from 3 Pointers              0
matchup_Percent of Points from Free Throws             0
matchup_Shooting %                                     0
matchup_Effective Field Goal %                         0
matchup_Three Point %                                  0
matchup_Two Point %                                    0
matchup_Free Throw %                                   0
matchup_True Shooting %                                0
matchup_Field Goals Made per Game                      0
matchup_Field Goals Attempted per Game                 0
matchup_Three Pointers Made per Game                   0
```

```
matchup_Three Pointers Attempted per Game                    0
matchup_Free Throws Made per Game                            0
matchup_Free Throws Attempted per Game                       0
matchup_Three Point Rate                                     0
matchup_Two Point Rate                                       0
matchup_Free Throws Attempted per Field Goal Attempted       0
matchup_Free Throws Made per 100 Possessions                 0
matchup_Free Throws Attempted per Offensive Play             0
matchup_Non-blocked 2 Pt %                                   0
matchup_Offensive Rebounds per Game                          0
matchup_Defensive Rebounds per Game                          0
matchup_Team Rebounds per Game                               0
matchup_Total Rebounds per Game                              0
matchup_Offensive Rebounding %                               0
matchup_Defensive Rebounding %                               0
matchup_Total Rebounding % (Rebound Rate)                    0
matchup_Blocks per Game                                      0
matchup_Steals per Game                                      0
matchup_Block %                                              0
matchup_Steals per Possession                                0
matchup_Steals per Defensive Play                            0
matchup_Assists per Game                                     0
matchup_Turnovers per Game                                   0
matchup_Turnovers per Possession                             0
matchup_Assist / Turnover Ratio                              0
matchup_Assists per FGM                                      0
matchup_Assists per Possession                               0
matchup_Turnovers per Offensive Play                         0
matchup_Personal Fouls per Game                              0
matchup_Personal Fouls per Possession                        0
matchup_Personal Fouls per Defensive Play                    0
matchup_Opponent Points per Game                             0
matchup_Opponent Average Scoring Margin                      0
matchup_Defensive Efficiency                                 0
matchup_Opponent Floor %                                     0
matchup_Opponent 1st Half Points per Game                    0
matchup_Opponent 2nd Half Points per Game                    0
matchup_Opponent Overtime Points per Game                    0
matchup_Opponent Points from 2 pointers                      0
matchup_Opponent Points from 3 pointers                      0
matchup_Opponent Percent of Points from 2 Pointers           0
matchup_Opponent Percent of Points from 3 Pointers           0
matchup_Opponent Percent of Points from Free Throws          0
```

```
matchup_Opponent Shooting %                                  0
matchup_Opponent Effective Field Goal %                      0
matchup_Opponent Three Point %                               0
matchup_Opponent Two Point %                                 0
matchup_Opponent Free Throw %                                0
matchup_Opponent True Shooting %                             0
matchup_Opponent Field Goals Made per Game                   0
matchup_Opponent Field Goals Attempted per Game              0
matchup_Opponent Three Pointers Made per Game                0
matchup_Opponent Three Pointers Attempted per Game           0
matchup_Opponent Free Throws Made per Game                   0
matchup_Opponent Free Throws Attempted per Game              0
matchup_Opponent Three Point Rate                            0
matchup_Opponent Two Point Rate                              0
matchup_Opponent FTA per FGA                                 0
matchup_Opponent Free Throws Made per 100 Possessions        0
matchup_Opponent Free Throws Attempted per Offensive Play    0
matchup_Opponent Non-blocked 2 Pt %                          0
matchup_Opponent Offensive Rebounds per Game                 0
matchup_Opponent Defensive Rebounds per Game                 0
matchup_Opponent Team Rebounds per Game                      0
matchup_Opponent Total Rebounds per Game                     0
matchup_Opponent Offensive Rebounding %                      0
matchup_Opponent Defensive Rebounding %                      0
matchup_Opponent Blocks per Game                             0
matchup_Opponent Steals per Game                             0
matchup_Opponent Block %                                     0
matchup_Opponent Steals per Possession                       0
matchup_Opponent Steals per Defensive Play                   0
matchup_Opponent Assists per Game                            0
matchup_Opponent Turnovers per Game                          0
matchup_Opponent Assist / Turnover Ratio                     0
matchup_Opponent Assists per FGM                             0
matchup_Opponent Assists per Possession                      0
matchup_Opponent Turnovers per Possession                    0
matchup_Opponent Turnovers per Offensive Play                0
matchup_Opponent Personal Fouls per Game                     0
matchup_Opponent Personal Fouls per Possession               0
matchup_Opponent Personal Fouls per Defensive Play           0
matchup_Games Played                                         0
matchup_Possessions per Game                                 0
matchup_Extra Scoring Chances per Game                       0
matchup_Effective Possession Ratio                           0
```

```
matchup_Opponent Effective Possession Ratio                  0
matchup_Win % - All Games                                    0
matchup_Win % - Close Games                                  0
matchup_Opponent Win % - All Games                           0
matchup_Opponent Win % - Close Games                         0
```

```
# Save the dataset
write_csv(matchup_stats_clean, "matchup_statistics.csv")

print(paste("Total games:", nrow(matchup_stats_clean)))
```

```
[1] "Total games: 400"
```

```
print(paste("Total upsets:", sum(matchup_stats_clean$is_upset)))
```

```
[1] "Total upsets: 40"
```

```
print(paste("Upset percentage:", round(100 * mean(matchup_stats_clean$is_upset), 1), "%"))
```

```
[1] "Upset percentage: 10 %"
```