NBA Advanced Stats Analysis

Frank Chen: 400238573

09/03/2020

Introduction

For this final project capitulating the HTHSCI 1M03 course, I wanted to investigate a long-time passion of mine: the National Basketball Association.

Data

The data was obtained from Kaggle, found here:

https://www.kaggle.com/pablote/nba-enhanced-stats#2012-18_officialBoxScore.csv

It includes a summary of basic and advanced NBA stats from 2012-2018 for each regular season game in that time.

This data is interesting because, in basketball, there's hot debate surrounding the shot selection and the importance of some stats over others towards winning. Different teams prioritze different actions, and the effectiveness of such actions is reflected statisically.

Of course, you can't do it all – to find these metrics, this data analysis report looka at statistics that the winningest teams tend to share and those they sacrifice. We also hope to see how the game has changed over time and how teams and organizations emphasize specific stats to optimize winning.

Ultimately, finding trends to optimize success helps NBA teams and the cities that root for them.

Research Questions:

- What statistics do winning teams share that contributes to their success the most?
- How has the understanding of these winning stats changed as coaches game plan differently based on the knowledge of mistakes in previous years? (ex., do teams shoot more threes now?)

Glossary of NBA Stats

In case you don't know these stats, here is a summary, obtained from https://www.basketball-reference.com/about/glossary.html and personal knowledge.

Basics:

- PTS: points
- AST: assists
- (O/D/T)RB: (Offensive/Defensive/Total) rebounds

- TO: turnovers
- STL: steals
- BLK: blocks
- PF: personal fouls
- FGA: field goals attempted
- FGM: field goals made
- FG%: field goal percentage ($\frac{FGA}{FGM} * 100\%$)
- 2P: two-pointers
- 3P: three-pointers
- FT: free throws

Advanced:

- TS: true shooting - intended as more accurate measure of team shooting efficiency
- eFG: effective field goal percentage - adjusts FG% to account for higher point value of threes. Measure of the 2P% necessary to match output of a player shooting 2s and 3s.
- PPS: points per shot, measures team efficiency
- FIC/FIC40: sum of offensive and defensive stats given different weights to replicate team impact on floor, can be scaled to 40 minutes.
- (O/D)RTG, EDiff: used to measure total offensive/defensive output, EDiff measures the difference bewteen ORTG and DRTG
- AST/TO Ratio: measures ball control of a team
- STL/TO Ratio: measures defensive output against potential offensive liability
- Pace/Poss: number of offensive possessions the team is accountable for

Data Wrangling Plan

Iteration 1

`Conversion to $tidy$ format.`

a. Read in the data

b. Take a look at the stats

- Note: This data is not in *tidy* format given that it fails the rule: each observation must have its own row. Because the official name is the primary key in this data, the same game is listed three times to account for each referee. Some basic renaming and data manipulation to make it easier to pivot the data into *tidy* form.

c. Check the amount of times each official appears to see if lump is necessary.

d. Make official names one column in preparation for pivoting.

e. Recode official names as a factor.

f. Lump officials with under 3 seasons (246 games) of experience.

g. Delete old official name columns

h. Convert data to *tidy* data by pivoting

i. Replace NAs from pivoting with 0s

```
# a)
stats <- read_csv("C:/Users/frank/OneDrive/Documents/NBA_Stats/2012-18_officialBoxScore.csv")
```

```
## Parsed with column specification:
## cols(
##   .default = col_double(),
##   gmDate = col_date(format = ""),
##   gmTime = col_time(format = ""),
##   seasTyp = col_character(),
##   offLNm = col_character(),
##   offFNm = col_character(),
##   teamAbbr = col_character(),
##   teamConf = col_character(),
##   teamDiv = col_character(),
##   teamLoc = col_character(),
##   teamRslt = col_character(),
##   opptAbbr = col_character(),
##   opptConf = col_character(),
##   opptDiv = col_character(),
##   opptLoc = col_character(),
##   opptRslt = col_character()
## )
```

```
## See spec(...) for full column specifications.
```

```
# b)
glimpse(stats)
```

```
## Observations: 44,284
## Variables: 119
## $ gmDate       <date> 2012-10-30, 2012-10-30, 2012-10-30, 2012-10-30, 2012-10...
## $ gmTime       <time> 19:00:00, 19:00:00, 19:00:00, 19:00:00, 19:00:00, 19:00...
## $ seasTyp      <chr> "Regular", "Regular", "Regular", "Regular", "Regular", "...
## $ offLNm       <chr> "Brothers", "Smith", "Workman", "Brothers", "Smith", "Wo...
## $ offFNm       <chr> "Tony", "Michael", "Haywoode", "Tony", "Michael", "Haywo...
## $ teamAbbr     <chr> "WAS", "WAS", "WAS", "CLE", "CLE", "CLE", "BOS", "BOS", ...
## $ teamConf     <chr> "East", "East", "East", "East", "East", "East", "East", ...
## $ teamDiv      <chr> "Southeast", "Southeast", "Southeast", "Central", "Centr...
## $ teamLoc      <chr> "Away", "Away", "Away", "Home", "Home", "Home", "Away", ...
## $ teamRslt     <chr> "Loss", "Loss", "Loss", "Win", "Win", "Win", "Loss", "Lo...
## $ teamMin      <dbl> 240, 240, 240, 240, 240, 240, 240, 240, 240, 240, 240, 2...
## $ teamDayOff   <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ teamPTS      <dbl> 84, 84, 84, 94, 94, 94, 107, 107, 107, 120, 120, 120, 99...
## $ teamAST      <dbl> 26, 26, 26, 22, 22, 22, 24, 24, 24, 25, 25, 25, 22, 22, ...
```

```
## $ teamTO       <dbl> 13, 13, 13, 21, 21, 21, 16, 16, 16, 8, 8, 8, 12, 12, 12,...
## $ teamSTL      <dbl> 11, 11, 11, 7, 7, 7, 4, 4, 4, 8, 8, 8, 9, 9, 9, 6, 6, 6,...
## $ teamBLK      <dbl> 10, 10, 10, 5, 5, 5, 2, 2, 2, 5, 5, 5, 5, 5, 5, 5, 5, 5,...
## $ teamPF       <dbl> 19, 19, 19, 21, 21, 21, 23, 23, 23, 20, 20, 20, 25, 25, ...
## $ teamFGA      <dbl> 90, 90, 90, 79, 79, 79, 75, 75, 75, 79, 79, 79, 85, 85, ...
## $ teamFGM      <dbl> 32, 32, 32, 36, 36, 36, 39, 39, 39, 43, 43, 43, 40, 40, ...
## $ `teamFG%`    <dbl> 0.3556, 0.3556, 0.3556, 0.4557, 0.4557, 0.4557, 0.5200, ...
## $ team2PA      <dbl> 58, 58, 58, 59, 59, 59, 62, 62, 62, 63, 63, 63, 70, 70, ...
## $ team2PM      <dbl> 24, 24, 24, 29, 29, 29, 33, 33, 33, 35, 35, 35, 35, 35, ...
## $ `team2P%`    <dbl> 0.4138, 0.4138, 0.4138, 0.4915, 0.4915, 0.4915, 0.5323, ...
## $ team3PA      <dbl> 32, 32, 32, 20, 20, 20, 13, 13, 13, 16, 16, 16, 15, 15, ...
## $ team3PM      <dbl> 8, 8, 8, 7, 7, 7, 6, 6, 6, 8, 8, 8, 5, 5, 5, 3, 3, 3, 4,...
## $ `team3P%`    <dbl> 0.2500, 0.2500, 0.2500, 0.3500, 0.3500, 0.3500, 0.4615, ...
## $ teamFTA      <dbl> 20, 20, 20, 22, 22, 22, 28, 28, 28, 32, 32, 32, 18, 18, ...
## $ teamFTM      <dbl> 12, 12, 12, 15, 15, 15, 23, 23, 23, 26, 26, 26, 14, 14, ...
## $ `teamFT%`    <dbl> 0.6000, 0.6000, 0.6000, 0.6818, 0.6818, 0.6818, 0.8214, ...
## $ teamORB      <dbl> 18, 18, 18, 18, 18, 18, 7, 7, 7, 5, 5, 5, 9, 9, 9, 15, 1...
## $ teamDRB      <dbl> 21, 21, 21, 36, 36, 36, 34, 34, 34, 31, 31, 31, 31, 31, ...
## $ teamTRB      <dbl> 39, 39, 39, 54, 54, 54, 41, 41, 41, 36, 36, 36, 40, 40, ...
## $ teamPTS1     <dbl> 24, 24, 24, 31, 31, 31, 25, 25, 25, 31, 31, 31, 25, 25, ...
## $ teamPTS2     <dbl> 15, 15, 15, 19, 19, 19, 29, 29, 29, 31, 31, 31, 23, 23, ...
## $ teamPTS3     <dbl> 23, 23, 23, 24, 24, 24, 22, 22, 22, 31, 31, 31, 26, 26, ...
## $ teamPTS4     <dbl> 22, 22, 22, 20, 20, 20, 31, 31, 31, 27, 27, 27, 25, 25, ...
## $ teamPTS5     <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,0,...
## $ teamPTS6     <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,0,...
## $ teamPTS7     <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,0,...
## $ teamPTS8     <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,0,...
## $ `teamTREB%`  <dbl> 41.9355, 41.9355, 41.9355, 58.0645, 58.0645, 58.0645, 53...
## $ `teamASST%`  <dbl> 81.2500, 81.2500, 81.2500, 61.1111, 61.1111, 61.1111, 61...
## $ `teamTS%`    <dbl> 0.4251, 0.4251, 0.4251, 0.5300, 0.5300, 0.5300, 0.6127, ...
## $ `teamEFG%`   <dbl> 0.4000, 0.4000, 0.4000, 0.5000, 0.5000, 0.5000, 0.5600, ...
## $ `teamOREB%`  <dbl> 33.3333, 33.3333, 33.3333, 46.1538, 46.1538, 46.1538, 18...
## $ `teamDREB%`  <dbl> 53.8462, 53.8462, 53.8462, 66.6667, 66.6667, 66.6667, 87...
## $ `teamTO%`    <dbl> 11.6279, 11.6279, 11.6279, 19.1466, 19.1466, 19.1466, 15...
## $ `teamSTL%`   <dbl> 12.3678, 12.3678, 12.3678, 7.8704, 7.8704, 7.8704, 4.211...
## $ `teamBLK%`   <dbl> 11.2434, 11.2434, 11.2434, 5.6217, 5.6217, 5.6217, 2.105...
## $ teamBLKR     <dbl> 17.2414, 17.2414, 17.2414, 8.4746, 8.4746, 8.4746, 3.225...
## $ teamPPS      <dbl> 0.9333, 0.9333, 0.9333, 1.1899, 1.1899, 1.1899, 1.4267, ...
## $ teamFIC      <dbl> 67.250, 67.250, 67.250, 74.000, 74.000, 74.000, 75.250, ...
## $ teamFIC40    <dbl> 56.0417, 56.0417, 56.0417, 61.6667, 61.6667, 61.6667, 62...
## $ teamOrtg     <dbl> 94.4447, 94.4447, 94.4447, 105.6882, 105.6882, 105.6882,...
## $ teamDrtg     <dbl> 105.6882, 105.6882, 105.6882, 94.4447, 94.4447, 94.4447,...
## $ teamEDiff    <dbl> -11.2435, -11.2435, -11.2435, 11.2435, 11.2435, 11.2435,...
## $ `teamPlay%`  <dbl> 0.3765, 0.3765, 0.3765, 0.4390, 0.4390, 0.4390, 0.4643, ...
## $ teamAR       <dbl> 18.8679, 18.8679, 18.8679, 16.7072, 16.7072, 16.7072, 18...
## $ `teamAST/TO` <dbl> 2.0000, 2.0000, 2.0000, 1.0476, 1.0476, 1.0476, 1.5000, ...
## $ `teamSTL/TO` <dbl> 84.6154, 84.6154, 84.6154, 33.3333, 33.3333, 33.3333, 25...
## $ opptAbbr     <chr> "CLE", "CLE", "CLE", "WAS", "WAS", "WAS", "MIA", "MIA", ...
## $ opptConf     <chr> "East", "East", "East", "East", "East", "East", "East", ...
## $ opptDiv      <chr> "Central", "Central", "Central", "Southeast", "Southeast...
## $ opptLoc      <chr> "Home", "Home", "Home", "Away", "Away", "Away", "Home", ...
## $ opptRslt     <chr> "Win", "Win", "Win", "Loss", "Loss", "Loss", "Win", "Win...
## $ opptMin      <dbl> 240, 240, 240, 240, 240, 240, 240, 240, 240, 240, 240, 2...
## $ opptDayOff   <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
```

```
## $ opptPTS      <dbl> 94, 94, 94, 84, 84, 84, 120, 120, 120, 107, 107, 107, 91...
## $ opptAST      <dbl> 22, 22, 22, 26, 26, 26, 25, 25, 25, 24, 24, 24, 24, 24, ...
## $ opptTO       <dbl> 21, 21, 21, 13, 13, 13, 8, 8, 8, 16, 16, 16, 14, 14, 14,...
## $ opptSTL      <dbl> 7, 7, 7, 11, 11, 11, 8, 8, 8, 4, 4, 4, 6, 6, 6, 9, 9, 9,...
## $ opptBLK      <dbl> 5, 5, 5, 10, 10, 10, 5, 5, 5, 2, 2, 2, 5, 5, 5, 5, 5, 5,...
## $ opptPF       <dbl> 21, 21, 21, 19, 19, 19, 20, 20, 20, 23, 23, 23, 21, 21, ...
## $ opptFGA      <dbl> 79, 79, 79, 90, 90, 90, 79, 79, 79, 75, 75, 75, 77, 77, ...
## $ opptFGM      <dbl> 36, 36, 36, 32, 32, 32, 43, 43, 43, 39, 39, 39, 38, 38, ...
## $ `opptFG%`    <dbl> 0.4557, 0.4557, 0.4557, 0.3556, 0.3556, 0.3556, 0.5443, ...
## $ oppt2PA      <dbl> 59, 59, 59, 58, 58, 58, 63, 63, 63, 62, 62, 62, 64, 64, ...
## $ oppt2PM      <dbl> 29, 29, 29, 24, 24, 24, 35, 35, 35, 33, 33, 33, 35, 35, ...
## $ `oppt2P%`    <dbl> 0.4915, 0.4915, 0.4915, 0.4138, 0.4138, 0.4138, 0.5556, ...
## $ oppt3PA      <dbl> 20, 20, 20, 32, 32, 32, 16, 16, 16, 13, 13, 13, 13, 13, ...
## $ oppt3PM      <dbl> 7, 7, 7, 8, 8, 8, 8, 8, 8, 6, 6, 6, 3, 3, 3, 5, 5, 5, 7,...
## $ `oppt3P%`    <dbl> 0.3500, 0.3500, 0.3500, 0.2500, 0.2500, 0.2500, 0.5000, ...
## $ opptFTA      <dbl> 22, 22, 22, 20, 20, 20, 32, 32, 32, 28, 28, 28, 31, 31, ...
## $ opptFTM      <dbl> 15, 15, 15, 12, 12, 12, 26, 26, 26, 23, 23, 23, 12, 12, ...
## $ `opptFT%`    <dbl> 0.6818, 0.6818, 0.6818, 0.6000, 0.6000, 0.6000, 0.8125, ...
## $ opptORB      <dbl> 18, 18, 18, 18, 18, 18, 5, 5, 5, 7, 7, 7, 15, 15, 15, 9,...
## $ opptDRB      <dbl> 36, 36, 36, 21, 21, 21, 31, 31, 31, 34, 34, 34, 31, 31, ...
## $ opptTRB      <dbl> 54, 54, 54, 39, 39, 39, 36, 36, 36, 41, 41, 41, 46, 46, ...
## $ opptPTS1     <dbl> 31, 31, 31, 24, 24, 24, 31, 31, 31, 25, 25, 25, 29, 29, ...
## $ opptPTS2     <dbl> 19, 19, 19, 15, 15, 15, 31, 31, 31, 29, 29, 29, 17, 17, ...
## $ opptPTS3     <dbl> 24, 24, 24, 23, 23, 23, 31, 31, 31, 22, 22, 22, 20, 20, ...
## $ opptPTS4     <dbl> 20, 20, 20, 22, 22, 22, 27, 27, 27, 31, 31, 31, 25, 25, ...
## $ opptPTS5     <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ opptPTS6     <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ opptPTS7     <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ opptPTS8     <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ `opptTREB%`  <dbl> 58.0645, 58.0645, 58.0645, 41.9355, 41.9355, 41.9355, 46...
## $ `opptASST%`  <dbl> 61.1111, 61.1111, 61.1111, 81.2500, 81.2500, 81.2500, 58...
## $ `opptTS%`    <dbl> 0.5300, 0.5300, 0.5300, 0.4251, 0.4251, 0.4251, 0.6446, ...
## $ `opptEFG%`   <dbl> 0.5000, 0.5000, 0.5000, 0.4000, 0.4000, 0.4000, 0.5949, ...
## $ `opptOREB%`  <dbl> 33.3333, 33.3333, 33.3333, 46.1538, 46.1538, 46.1538, 13...
## $ `opptDREB%`  <dbl> 66.6667, 66.6667, 66.6667, 53.8462, 53.8462, 53.8462, 86...
## $ `opptTO%`    <dbl> 19.1466, 19.1466, 19.1466, 11.6279, 11.6279, 11.6279, 7....
## $ `opptSTL%`   <dbl> 7.8704, 7.8704, 7.8704, 12.3678, 12.3678, 12.3678, 8.422...
## $ `opptBLK%`   <dbl> 5.6217, 5.6217, 5.6217, 11.2434, 11.2434, 11.2434, 5.264...
## $ opptBLKR     <dbl> 8.4746, 8.4746, 8.4746, 17.2414, 17.2414, 17.2414, 7.936...
## $ opptPPS      <dbl> 1.1899, 1.1899, 1.1899, 0.9333, 0.9333, 0.9333, 1.5190, ...
## $ opptFIC      <dbl> 74.000, 74.000, 74.000, 67.250, 67.250, 67.250, 97.000, ...
## $ opptFIC40    <dbl> 61.6667, 61.6667, 61.6667, 56.0417, 56.0417, 56.0417, 80...
## $ opptOrtg     <dbl> 105.6882, 105.6882, 105.6882, 94.4447, 94.4447, 94.4447,...
## $ opptDrtg     <dbl> 94.4447, 94.4447, 94.4447, 105.6882, 105.6882, 105.6882,...
## $ opptEDiff    <dbl> 11.2435, 11.2435, 11.2435, -11.2435, -11.2435, -11.2435,...
## $ `opptPlay%`  <dbl> 0.4390, 0.4390, 0.4390, 0.3765, 0.3765, 0.3765, 0.5244, ...
## $ opptAR       <dbl> 16.7072, 16.7072, 16.7072, 18.8679, 18.8679, 18.8679, 19...
## $ `opptAST/TO` <dbl> 1.0476, 1.0476, 1.0476, 2.0000, 2.0000, 2.0000, 3.1250, ...
## $ `opptSTL/TO` <dbl> 33.3333, 33.3333, 33.3333, 84.6154, 84.6154, 84.6154, 10...
## $ poss         <dbl> 88.9409, 88.9409, 88.9409, 88.9409, 88.9409, 88.9409, 94...
## $ pace         <dbl> 88.9409, 88.9409, 88.9409, 88.9409, 88.9409, 88.9409, 94...
```

```
# c)
check_off <- stats %>%
```

```
  select(offLNm) %>%
  group_by(offLNm) %>%
    dplyr::summarize(num_off = length(offLNm))

#kable(check_off)
kable(head(check_off), caption = "Truncated List of Referees and Games Officiated")
```

Table 1: Truncated List of Referees and Games Officiated

offLNm	num_off
Acosta	68
Adair	12
Adams	684
Anderson	406
Ayotte	734
Barnaky	728

```
#Some refs barely have referee games, lump them into a section called "Other"
stats <- stats %>%
  # d) and e)
  mutate(., off = as_factor(str_c(offFNm, offLNm, sep = "_")),
         # f)
         off = fct_lump_min(off, 246),
         # Add a value column for pivoting purposes later on
         val = 1) %>%
  # g)
  select(., -offFNm, -offLNm) %>%
  # h)
  pivot_wider(., names_from = off,
              values_from = val,
              values_fn = list(val = length)) %>%
  # i)
  mutate_if(., is.numeric, ~replace_na(., 0))
```

Iteration 2

`Convert data to tidy data, remove columns and modify names, recode factors, and modify date-times.`

a. Check for NA values.

b. Remove useless columns

- Some columns are irrelevant, such as the team division, or minutes played by each team. Some columns contained stats rarely talked about in basketball analytics and by fans were discarded. Such include: points by quarters, teamBLKR, and many more. Also, metrics like BLK% are generally used for players, not for team analyses.

c. Turn column names to lowercase

d. Add a _ in column names for separation and clarity

e. $team_abbr$, $team_conf$, $team_rslt$, $oppt_abbr$, $oppt_conf$, $oppt_rslt$ should all be coded as factors

f. Convert to dates to date-time and delete previous date and time separate columns.

g. Check levels of columns mentioned in e).

```
# function written to reorder the team names levels by descending
# alpha order for a plot later
fct_sort = function(f, .fun = sort) {
  fct_relevel(f, .fun(levels(f), decreasing = TRUE))
}

# a)
any(is.na(stats))
```

```
## [1] FALSE
```

```
# No NA values
stats1 <- stats %>%
  #slice(1:24) %>%
  # b)
  select(., -seasTyp, -teamDiv, -teamMin, -teamDayOff, -teamLoc,
         -teamBLKR, -`teamPlay%`, -teamAR, -`teamASST%`, -`teamBLK%`,
         -`teamSTL%`, -`teamTO%`, -`teamOREB%`, -`teamDREB%`,
         -c(opptAbbr:poss), -c(Tony_Brothers:Gediminas_Petraitis)) %>%
  # c)
  rename_all(tolower) %>%
  # d)
  setNames(gsub("team", "team_", names(.))) %>%
  mutate(.,
         # e)
         team_abbr = fct_sort(as_factor(team_abbr)),
         team_conf = as_factor(team_conf),
         team_rslt = fct_relevel(as_factor(team_rslt), "Win", "Loss"),
         # f)
         dttm = parse_datetime(str_c(gmdate, gmtime, sep = " ")),
         # add this to for pivoting the data
         val = 1) %>%
  # delete these columns after using them in mutate above
  select(., -gmdate, -gmtime)

# g)
levels(stats1$team_abbr)
```

```
##  [1] "WAS" "UTA" "TOR" "SAC" "SA"  "POR" "PHO" "PHI" "ORL" "OKC" "NY"  "NO"
## [13] "MIN" "MIL" "MIA" "MEM" "LAL" "LAC" "IND" "HOU" "GS"  "DET" "DEN" "DAL"
## [25] "CLE" "CHI" "CHA" "BOS" "BKN" "ATL"
```

```
levels(stats1$team_conf)
```

```
## [1] "East" "West"
```

```r
levels(stats1$team_rslt)
```

```
## [1] "Win"  "Loss"
```

Iteration 3:

`Purpose is to manipulate the actual data values for better visualization and analysis.`

a. Convert STL/TO ratio into decimals.

b. Add columns for points scored in regulation and overtime (OT).

c. Add boolean column indicating whether game went to OT.

d. Delete the columns representing points by quarters.

e. Reorder columns so identifiers like time, team playing, win/loss, etc., are close together, also so basic stats, percentage stats, and advanced stats are grouped together and are in conventional order: PTS, AST, ORB, DRB, TRB.

f. Prepare a summary table.

```r
stats2 <- stats1 %>%
  # a)
  mutate(., `team_stl/to` = `team_stl/to`/100,
         # b)
         team_reg_pts = team_pts1 + team_pts2 + team_pts3 + team_pts4,
         team_ot_pts = team_pts5 + team_pts6 + team_pts7 + team_pts8,
         # c)
         ot = if_else(team_ot_pts > 0, TRUE, FALSE)
         ) %>%
  # d)
  select(., -c(team_pts1:team_pts8)) %>%
  # e)
  select(., dttm, team_abbr:team_rslt, ot, team_pts, team_reg_pts,
         team_ot_pts, team_ast, team_orb:team_trb, team_to:pace)
# f)
# summary tables created using the papeR
# (Hofner, B., 2020)
kable(papeR::summarize_numeric(as.data.frame(stats2)), caption = "NBA Statistics Summary of Numeric Data
```

```
## Registered S3 method overwritten by 'papeR':
##   method    from
##   Anova.lme car
```

```
## Factors are dropped from the summary
```

Table 2: NBA Statistics Summary of Numeric Data

	N	Mean	SD	Min	Q1	Median	Q3	Max
team_pts	14758	102.29	12.22	58.00	94.00	102.00	110.00	149.00
team_reg_pts	14758	101.57	11.90	58.00	94.00	101.00	109.00	149.00
team_ot_pts	14758	0.72	3.20	0.00	0.00	0.00	0.00	42.00
team_ast	14758	22.39	5.08	6.00	19.00	22.00	26.00	47.00
team_orb	14758	10.54	3.87	0.00	8.00	10.00	13.00	38.00
team_drb	14758	32.62	5.34	12.00	29.00	32.00	36.00	56.00
team_trb	14758	43.16	6.47	17.00	39.00	43.00	47.00	81.00
team_to	14758	14.37	3.90	2.00	12.00	14.00	17.00	31.00
team_stl	14758	7.75	2.92	0.00	6.00	8.00	10.00	21.00
team_blk	14758	4.86	2.58	0.00	3.00	5.00	6.00	18.00
team_pf	14758	20.13	4.34	5.00	17.00	20.00	23.00	42.00
team_fga	14758	84.10	7.26	60.00	79.00	84.00	89.00	129.00
team_fgm	14758	38.21	5.02	19.00	35.00	38.00	42.00	58.00
team_fg%	14758	0.46	0.06	0.27	0.42	0.45	0.49	0.68
team_2pa	14758	60.10	8.60	28.00	54.00	60.00	66.00	113.00
team_2pm	14758	29.64	5.07	11.00	26.00	30.00	33.00	52.00
team_2p%	14758	0.50	0.07	0.25	0.45	0.49	0.54	0.80
team_3pa	14758	24.00	7.13	4.00	19.00	24.00	29.00	61.00
team_3pm	14758	8.57	3.56	0.00	6.00	8.00	11.00	25.00
team_3p%	14758	0.35	0.10	0.00	0.29	0.35	0.42	0.82
team_fta	14758	22.80	7.42	1.00	18.00	22.00	27.00	64.00
team_ftm	14758	17.30	6.00	1.00	13.00	17.00	21.00	52.00
team_ft%	14758	0.76	0.10	0.14	0.70	0.77	0.83	1.00
team_treb%	14758	50.00	5.25	27.91	46.51	50.00	53.49	72.09
team_ts%	14758	0.54	0.06	0.33	0.50	0.54	0.59	0.79
team_efg%	14758	0.51	0.07	0.28	0.46	0.51	0.55	0.76
team_pps	14758	1.22	0.15	0.71	1.11	1.22	1.32	1.88
team_fic	14758	76.25	16.24	18.88	65.12	75.75	86.88	147.88
team_fic40	14758	63.09	13.43	15.73	53.87	62.55	71.77	123.32
team_ortg	14758	107.06	11.31	65.31	99.43	107.08	114.55	155.62
team_drtg	14758	107.06	11.31	65.31	99.43	107.08	114.55	155.62
team_ediff	14758	0.00	14.41	-62.41	-9.54	0.00	9.54	62.41
team_ast/to	14758	1.70	0.71	0.32	1.22	1.57	2.00	10.00
team_stl/to	14758	0.58	0.28	0.00	0.38	0.53	0.71	5.00
pace	14758	94.87	5.16	77.00	91.33	94.76	98.27	121.66

```
kable(papeR::summarize_factor(as.data.frame(stats2)), caption = "NBA Statistics Summary of Factor Data")
```

```
## Non-factors are dropped from the summary
```

Table 3: NBA Statistics Summary of Factor Data

	Level	N	%
team_abbr	WAS	492	3.3
	UTA	492	3.3
	TOR	492	3.3
	SAC	492	3.3
	SA	492	3.3

	Level	N	%
	POR	492	3.3
	PHO	492	3.3
	PHI	492	3.3
	ORL	492	3.3
	OKC	492	3.3
	NY	492	3.3
	NO	492	3.3
	MIN	492	3.3
	MIL	492	3.3
	MIA	492	3.3
	MEM	492	3.3
	LAL	492	3.3
	LAC	492	3.3
	IND	491	3.3
	HOU	492	3.3
	GS	492	3.3
	DET	492	3.3
	DEN	492	3.3
	DAL	492	3.3
	CLE	492	3.3
	CHI	492	3.3
	CHA	492	3.3
	BOS	491	3.3
	BKN	492	3.3
	ATL	492	3.3
team_conf	East	7378	50.0
	West	7380	50.0
team_rslt	Win	7379	50.0
	Loss	7379	50.0

```
glimpse(stats2)
```

```
## Observations: 14,758
## Variables: 40
## $ dttm          <dttm> 2012-10-30 19:00:00, 2012-10-30 19:00:00, 2012-10-30 2...
## $ team_abbr     <fct> WAS, CLE, BOS, MIA, DAL, LAL, DEN, PHI, IND, TOR, HOU, ...
## $ team_conf     <fct> East, East, East, East, West, West, West, East, East, E...
## $ team_rslt     <fct> Loss, Win, Loss, Win, Win, Loss, Loss, Win, Win, Loss, ...
## $ ot            <lgl> FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE,...
## $ team_pts      <dbl> 84, 94, 107, 120, 99, 91, 75, 84, 90, 88, 105, 96, 87, ...
## $ team_reg_pts  <dbl> 84, 94, 107, 120, 99, 91, 75, 84, 90, 88, 105, 96, 87, ...
## $ team_ot_pts   <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ team_ast      <dbl> 26, 22, 24, 25, 22, 24, 19, 18, 22, 18, 28, 21, 14, 21,...
## $ team_orb      <dbl> 18, 18, 7, 5, 9, 15, 16, 14, 9, 15, 12, 10, 11, 14, 10,...
## $ team_drb      <dbl> 21, 36, 34, 31, 31, 31, 38, 33, 37, 27, 33, 26, 29, 32,...
## $ team_trb      <dbl> 39, 54, 41, 36, 40, 46, 54, 47, 46, 42, 45, 36, 40, 46,...
## $ team_to       <dbl> 13, 21, 16, 8, 12, 14, 22, 16, 19, 10, 20, 16, 21, 18, ...
## $ team_stl      <dbl> 11, 7, 4, 8, 9, 6, 9, 13, 3, 12, 12, 12, 10, 8, 9, 7, 8...
## $ team_blk      <dbl> 10, 5, 2, 5, 5, 5, 5, 11, 10, 8, 3, 4, 5, 10, 8, 6, 5, ...
## $ team_pf       <dbl> 19, 21, 23, 20, 25, 21, 22, 14, 16, 18, 24, 15, 29, 19,...
## $ team_fga      <dbl> 90, 79, 75, 79, 85, 77, 88, 85, 78, 91, 79, 79, 84, 79,...
```

```
## $ team_fgm      <dbl> 32, 36, 39, 43, 40, 38, 33, 30, 37, 33, 39, 35, 34, 33,...
## $ `team_fg%`    <dbl> 0.3556, 0.4557, 0.5200, 0.5443, 0.4706, 0.4935, 0.3750,...
## $ team_2pa      <dbl> 58, 59, 62, 63, 70, 64, 70, 60, 67, 74, 52, 63, 67, 70,...
## $ team_2pm      <dbl> 24, 29, 33, 35, 35, 35, 29, 23, 32, 27, 29, 29, 28, 31,...
## $ `team_2p%`    <dbl> 0.4138, 0.4915, 0.5323, 0.5556, 0.5000, 0.5469, 0.4143,...
## $ team_3pa      <dbl> 32, 20, 13, 16, 15, 13, 18, 25, 11, 17, 27, 16, 17, 9, ...
## $ team_3pm      <dbl> 8, 7, 6, 8, 5, 3, 4, 7, 5, 6, 10, 6, 6, 2, 6, 4, 11, 6,...
## $ `team_3p%`    <dbl> 0.2500, 0.3500, 0.4615, 0.5000, 0.3333, 0.2308, 0.2222,...
## $ team_fta      <dbl> 20, 22, 28, 32, 18, 31, 11, 21, 16, 19, 23, 26, 16, 33,...
## $ team_ftm      <dbl> 12, 15, 23, 26, 14, 12, 5, 17, 11, 16, 17, 20, 13, 25, ...
## $ `team_ft%`    <dbl> 0.6000, 0.6818, 0.8214, 0.8125, 0.7778, 0.3871, 0.4545,...
## $ `team_treb%`  <dbl> 41.9355, 58.0645, 53.2468, 46.7532, 46.5116, 53.4884, 5...
## $ `team_ts%`    <dbl> 0.4251, 0.5300, 0.6127, 0.6446, 0.5327, 0.5020, 0.4039,...
## $ `team_efg%`   <dbl> 0.4000, 0.5000, 0.5600, 0.5949, 0.5000, 0.5130, 0.3977,...
## $ team_pps      <dbl> 0.9333, 1.1899, 1.4267, 1.5190, 1.1647, 1.1818, 0.8523,...
## $ team_fic      <dbl> 67.250, 74.000, 75.250, 97.000, 72.250, 70.375, 49.375,...
## $ team_fic40    <dbl> 56.0417, 61.6667, 62.7083, 80.8333, 60.2083, 58.6458, 4...
## $ team_ortg     <dbl> 94.4447, 105.6882, 112.6515, 126.3381, 108.1034, 99.367...
## $ team_drtg     <dbl> 105.6882, 94.4447, 126.3381, 112.6515, 99.3678, 108.103...
## $ team_ediff    <dbl> -11.2435, 11.2435, -13.6866, 13.6866, 8.7356, -8.7356, ...
## $ `team_ast/to` <dbl> 2.0000, 1.0476, 1.5000, 3.1250, 1.8333, 1.7143, 0.8636,...
## $ `team_stl/to` <dbl> 0.846154, 0.333333, 0.250000, 1.000000, 0.750000, 0.428...
## $ pace          <dbl> 88.9409, 88.9409, 94.9832, 94.9832, 91.5790, 91.5790, 9...
```

The tidy data contains the unique identifiers of date-time and team. This is divided into two rows: the first representing the home team and the second representing the away team of the game played in that day and time. The following columns represent useful team statistics measured in some capacity of that specific game played.

With workable data, we will try to answer some of our questions:

Results/Discussion

Question 1

`1. What statistics do winning teams share that contributes to their success the most?`

First, which teams won the most over this time period?

```r
plot <- stats2 %>%
  # Take only the wins
  filter(., team_rslt == "Win") %>%
  # Group by team and count number of wins using summarize
  group_by(team_abbr) %>%
  summarize(n = n()) %>%
  ggplot(aes(y = fct_reorder(team_abbr, n), x = n)) +
  geom_col() +
  theme_minimal() +
  theme(line=element_blank()) +
  labs(x = "Games", y = "Team", fill = "Wins")
```

Modifications to the data. The data was filtered for only wins, and this was calculated by grouping the data by teams and counting the number of rows for each team.

```
plot + plot_annotation("Win/Loss from 2012-2018 of NBA teams")
```

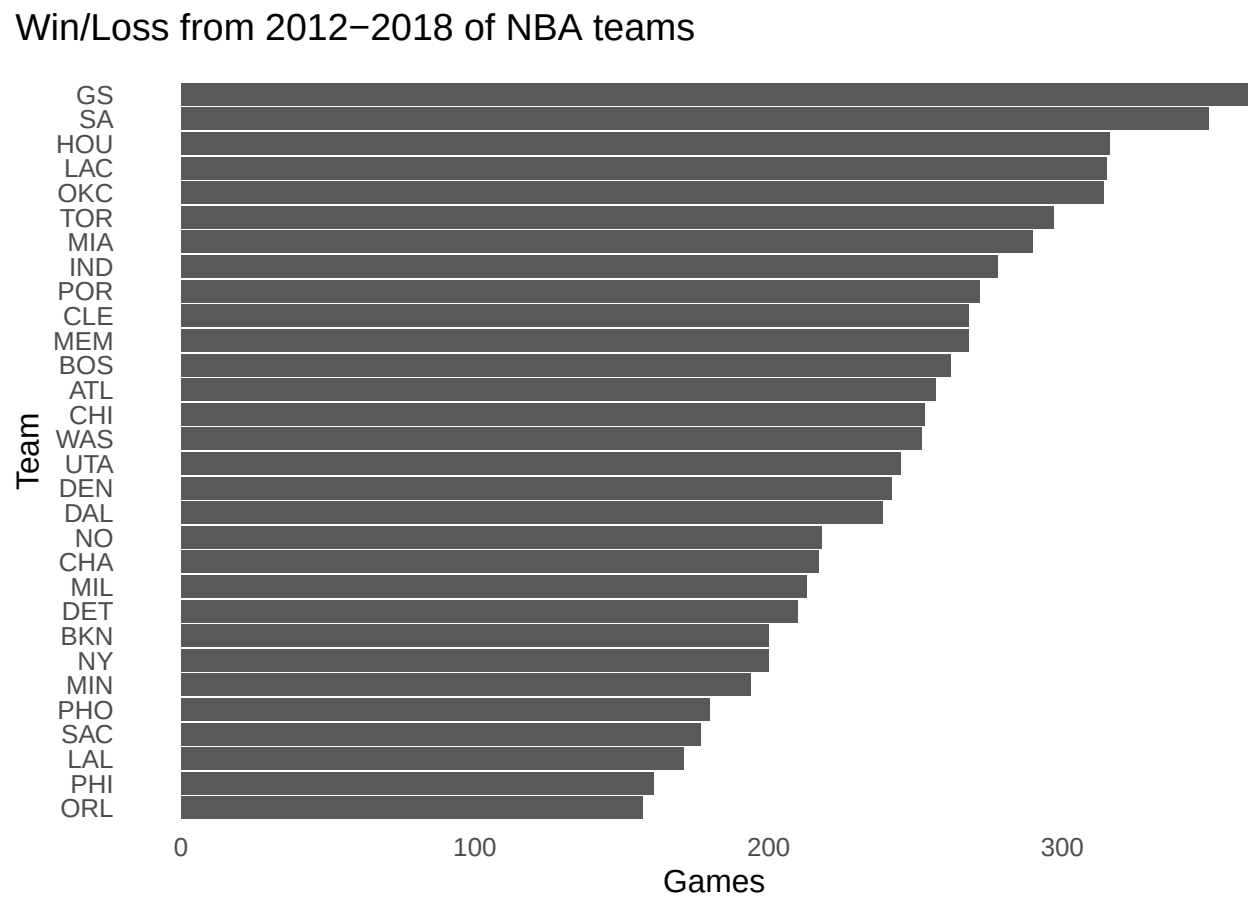

Figure 1: Sum of wins and losses of NBA teams over 2012-2018.

From Figure 1, we can see which teams performed the best and worst in this time frame. Taking the top 10 and bottom 10 teams, a comparison of stats will be performed. By looking at statistics of the winning group that far outweigh stats of the losing group, it is possible to identify metrics that correlate to success.

Top 10 teams: GS, SA, HOU, LAC, OKC, TOR, MIA, IND, POR, CLE

Bottom 10 teams: MIL, DET, BKN, NY, MIN, PHO, SAC, LAL, PHI, ORL

```
# Vector used for creating colour coded columns
cols <- c("AST"="magenta","ORB"="green","DRB"="blue",
          "STL" = "cyan", "BLK" = "black", "TO" = "purple")

grp1 <- stats2 %>%
  # Choose only the top 10 teams
  filter(., str_detect(team_abbr,
                       "(?!SAC)(GS|SA|HOU|LAC|OKC|TOR|MIA|IND|POR|CLE)")) %>%
  # Calculate medians for graphing the vertical lines on the plot (which increase clarity)
  mutate(., med_ast = median(team_ast), med_orb = median(team_orb),
         med_drb = median(team_drb), med_stl = median(team_stl),
```

```
        med_blk = median(team_blk), med_to = median(team_to)) %>%
  # Pivot data so that stat is in one column and can be graphed on an axis
  pivot_longer(., c(team_ast, team_orb, team_drb, team_stl, team_blk,
                    team_to),
               names_to = "stat",
               values_to = "value") %>%
  ggplot(aes(y = stat, x = value, fill = "Coral")) +
  # Modify size of dots and whiskers of boxplot
  geom_boxplot(lwd = 0.25, outlier.size = 0.35) +
  # Create vertical lines on the plot for better comparisons
  # Could have been done more efficiently after pivoting data,
  # but that would have not allowed the colours of lines
  geom_vline(aes(xintercept = med_ast, colour = "AST"), linetype = "dashed") +
  geom_vline(aes(xintercept = med_orb, colour = "ORB"), linetype = "dashed") +
  geom_vline(aes(xintercept = med_drb, colour = "DRB"), linetype = "dashed") +
  geom_vline(aes(xintercept = med_stl, colour = "STL"), linetype = "dashed") +
  geom_vline(aes(xintercept = med_blk, colour = "BLK"), linetype = "dashed") +
  geom_vline(aes(xintercept = med_to, colour = "TO"), linetype = "dashed") +
  # Labels and Scales
  labs(x = element_blank(), y = "Statistic", fill = "Winning Teams") +
  scale_x_continuous(limits = c(0, 51)) +
  scale_y_discrete(labels = c(team_ast = "AST", team_orb = "ORB",
                              team_drb = "DRB", team_stl = "STL",
                              team_blk = "BLK", team_to = "TO")) +
  theme(axis.ticks = element_blank()) +
  # Manually add scale for vertical lines
  scale_colour_manual(name = "Medians", values = cols)

grp2 <- stats2 %>%
  # Choose bottom 10 teams
  filter(., str_detect(team_abbr, "MIL|DET|BKN|NY|MIN|PHO|SAC|LAL|PHI|ORL")) %>%
  # Calculate medians for graphing the vertical lines on the plot (which increase clarity)
  mutate(., med_ast = median(team_ast), med_orb = median(team_orb),
         med_drb = median(team_drb), med_stl = median(team_stl),
         med_blk = median(team_blk), med_to = median(team_to)) %>%
  # Pivot data so that stat is in one column and can be graphed on an axis
  pivot_longer(., c(team_ast, team_orb, team_drb, team_stl, team_blk,
                    team_to),
               names_to = "stat",
               values_to = "value") %>%
  ggplot(aes(y = stat, x = value)) +
  geom_boxplot(lwd = 0.25, outlier.size = 0.35) +
  # Vertical Lines
  geom_vline(aes(xintercept = med_ast, colour = "AST"), linetype = "dashed") +
  geom_vline(aes(xintercept = med_orb, colour = "ORB"), linetype = "dashed") +
  geom_vline(aes(xintercept = med_drb, colour = "DRB"), linetype = "dashed") +
  geom_vline(aes(xintercept = med_stl, colour = "STL"), linetype = "dashed") +
  geom_vline(aes(xintercept = med_blk, colour = "BLK"), linetype = "dashed") +
  geom_vline(aes(xintercept = med_to, colour = "TO"), linetype = "dashed") +
  # Labels
  labs(x = "Value", y = "Statistic") +
  scale_x_continuous(limits = c(0, 51)) +
  scale_y_discrete(labels = c(team_ast = "AST", team_orb = "ORB",
```

```
                              team_drb = "DRB", team_stl = "STL",
                              team_blk = "BLK", team_to = "TO")) +
  theme(axis.ticks = element_blank()) +
  # No legend for the fill
  guides(fill = FALSE) +
  # Manually add scale for vertical lines
  scale_colour_manual(name = "Medians", values = cols)
```

Modifications to the data: filtered out only the best/worst teams so only values of those plots were used. Added via mutate a median column that contains the median values of the stats plotted. This is used for the plotting vertical lines on the plot, which are useful as they allow for better comparisons to be made. Data was pivoted longer, putting the statistics plotted above into one column so they could be plotted on one axis, useful for the clarity of the graph.

```
# Plot using patchwork
grp1 / grp2 + plot_annotation("Stat Comparison of the winningest and worst NBA teams from 2012-2018",
                              tag_levels = c("A", "B")) +
  # Use one common legend
  plot_layout(guides = "collect")
```

```
## Warning: Removed 1 rows containing non-finite values (stat_boxplot).

## Warning: Removed 1 rows containing non-finite values (stat_boxplot).
```

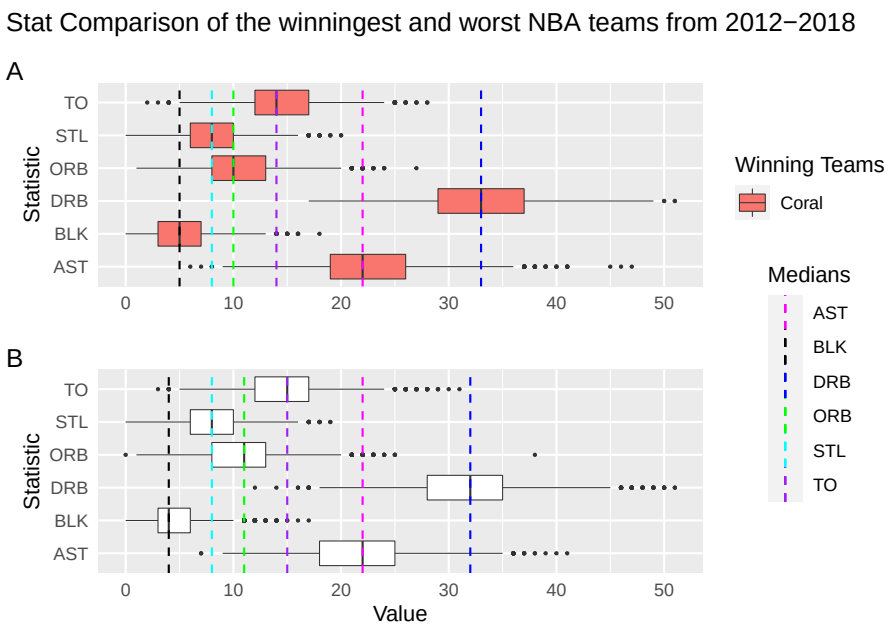

Figure 2: Basic stats disparity between the top 10 and worst 10 NBA teams during this time frame. The dashed line represents the median for each statistic across all 10 teams for ease of comparison between teams. Teams in A: GS, SA, HOU, LAC, OKC, TOR, MIA, IND, POR, CLE. Teams in B: MIL, DET, BKN, NY, MIN, PHO, SAC, LAL, PHI, ORL

Figure 2 shows that assists and steals may not be as impactful to winning as other statistics. It also reveals that better teams are worse at offensive rebounding. Discussed in further detail:

STL

Interestingly, we see here that the difference in both the distribution and median of steals between winning and non-winning teams is not very large. One possible contributor is the fact that STL and BLK are not necessarily metrics that measure defenses entirely well. (Franks et al. 2015) STLs are often risky, reaching in improperly by uncoordinated defenses or against savvy players (James Harden is a big example) can be detrimental more often than good. The act of going for a steal allows the offense to draw fouls and drive by you for an easy bucket, and teams who do it more may suffer from these consequences.

ORB

Furthermore, we see that teams who are worse are actually better in terms of offensive rebounds. I would be surprised that this is reflected in the data, if not for the 2019 Milwaukee Bucks, as well as other teams. A strategy the Bucks employ is to sacrifice the offensive rebound to get back on defense and being prepared to stopping the next possession. Chasing rebounds moves players out of position, and can lead to transition buckets. You lose the opportunity to score again, but gain coordination on the other end. It's possible that this is a reflection of the more organized defensive efforts of better teams.

AST

In the data above, both good and bad teams share a similar median amount of assists. However, a look at the distributions reveals that good teams will more often pass the ball than poorer teams. It's fascinating, however, that this disparity is not larger. In today's NBA, optimizing shot quality through making extra passes and forcing enemy defensive rotations has been heavily emphasized, thus it's strange to see that this is not a major contributor to winning.

TO and BLK

These statistics decrease and increase respectively with winning teams. This is expected: turn the ball over less, and you have more possessions to score. Blocked shots lead to defensive stops and often transition scoring. While the turnover distribution between the groups of teams are similar, the medians are different, suggesting that good teams are more consistent in having fewer turnovers. Consistency over the uniquely long 82-game NBA season is likely key to success.

DRB

Finally, we see an interesting trend where the top 5 teams defensively outrebound the bottom 5 teams. This is expected, but confusing, given that worse teams are better offensive rebounders, they should be even more capable of defensive rebounding. An article (Masheswaran et al., 2014) suggests that DRB is related to defensive capability, as opposing teams miss their shot attempts more. Another article (Mikolajec et al., 2013), suggests that decreased basket proximity with more spacing by better 3-point shooting teams result in this relationship.

```
# Vector used to create manual legend for vertical lines
cols <- c("FG%"="magenta","2P%"="green","3P%"="blue",
          "FT%" = "cyan", "TS%" = "black", "eFG%" = "purple")

grp3 <- stats2 %>%
  filter(., str_detect(team_abbr,
                       "(?!SAC)(GS|SA|HOU|LAC|OKC|TOR|MIA|IND|POR|CLE)")) %>%
  mutate(., med_fg = median(`team_fg%`), `med_2p` = median(`team_2p%`),
         med_3p = median(`team_3p%`), med_ft = median(`team_ft%`),
         med_ts = median(`team_ts%`), med_efg = median(`team_efg%`)) %>%
  pivot_longer(., c(`team_fg%`, `team_2p%`, `team_3p%`, `team_ft%`,
                    `team_ts%`, `team_efg%`),
               names_to = "stat",
               values_to = "value") %>%
  #pivot_longer(., c(med_fg, med_2p, med_3p, med_ft, med_ts, med_efg),
  #             names_to = "med",
```

```
  #             values_to = "val") %>%
  ggplot(aes(y = stat, x = value, fill = "Coral")) +
  geom_boxplot(lwd = 0.25, outlier.size = 0.35) +
  # Plot horizontal lines corresponding to each statistic. Would have been done with pivot
  # for one succinct geom_hline, however, that does not allow for colour modifications
  # (colour parameter requires "name of colour")
  geom_vline(aes(xintercept = med_fg, colour = "FG%"), linetype = "dashed") +
  geom_vline(aes(xintercept = med_2p, colour = "2P%"), linetype = "dashed") +
  geom_vline(aes(xintercept = med_3p, colour = "3P%"), linetype = "dashed") +
  geom_vline(aes(xintercept = med_ft, colour = "FT%"), linetype = "dashed") +
  geom_vline(aes(xintercept = med_ts, colour = "TS%"), linetype = "dashed") +
  geom_vline(aes(xintercept = med_efg, colour = "eFG%"), linetype = "dashed") +
  labs(x = element_blank(), y = "Statistic", fill = "Winning Teams") +
  scale_x_continuous(limits = c(0, 1),
                     labels = percent) +
  scale_y_discrete(labels = c(`team_2p%` = "2P%", `team_3p%` = "3P%",
                              `team_fg%` = "FG%", `team_ft%` = "FT%",
                              `team_efg%` = "eFG%", `team_ts%` = "TS%"))+
  theme(axis.ticks = element_blank()) +
  scale_colour_manual(name = "Medians", values = cols)

grp4 <- stats2 %>%
  filter(., str_detect(team_abbr, "MIL|DET|BKN|NY|MIN|PHO|SAC|LAL|PHI|ORL")) %>%
  mutate(., med_fg = median(`team_fg%`), `med_2p` = median(`team_2p%`),
         med_3p = median(`team_3p%`), med_ft = median(`team_ft%`),
         med_ts = median(`team_ts%`), med_efg = median(`team_efg%`)) %>%
  pivot_longer(., c(`team_fg%`, `team_2p%`, `team_3p%`, `team_ft%`,
                    `team_ts%`, `team_efg%`),
               names_to = "stat",
               values_to = "value") %>%
  ggplot(aes(y = stat, x = value)) +
  geom_boxplot(lwd = 0.25, outlier.size = 0.35) +
  geom_vline(aes(xintercept = med_fg, colour = "FG%"), linetype = "dashed") +
  geom_vline(aes(xintercept = med_2p, colour = "2P%"), linetype = "dashed") +
  geom_vline(aes(xintercept = med_3p, colour = "3P%"), linetype = "dashed") +
  geom_vline(aes(xintercept = med_ft, colour = "FT%"), linetype = "dashed") +
  geom_vline(aes(xintercept = med_ts, colour = "TS%"), linetype = "dashed") +
  geom_vline(aes(xintercept = med_efg, colour = "eFG%"), linetype = "dashed") +
  labs(x = "Percent", y = "Statistic", fill = "Blue") +
  scale_x_continuous(limits = c(0,1),
                     labels = percent) +
  scale_y_discrete(labels = c(`team_2p%` = "2P%", `team_3p%` = "3P%",
                              `team_fg%` = "FG%", `team_ft%` = "FT%",
                              `team_efg%` = "eFG%", `team_ts%` = "TS%")) +
  theme(axis.ticks = element_blank()) +
  guides(fill = FALSE) +
  scale_colour_manual(name = "Medians", values = cols)
```

Modifications to the data: All modifications are the same as in Figure 2, but when pivoting the data the statistics placed into one column are different (these are shooting statistics). The reasoning and the usefulness behind these modifications can also be found as previously stated.

```
grp3 / grp4 + plot_annotation("Shooting %s of the winningest and worst NBA teams from 2012-2018",
                              tag_levels = c("A", "B")) +
  plot_layout(guides = "collect")
```

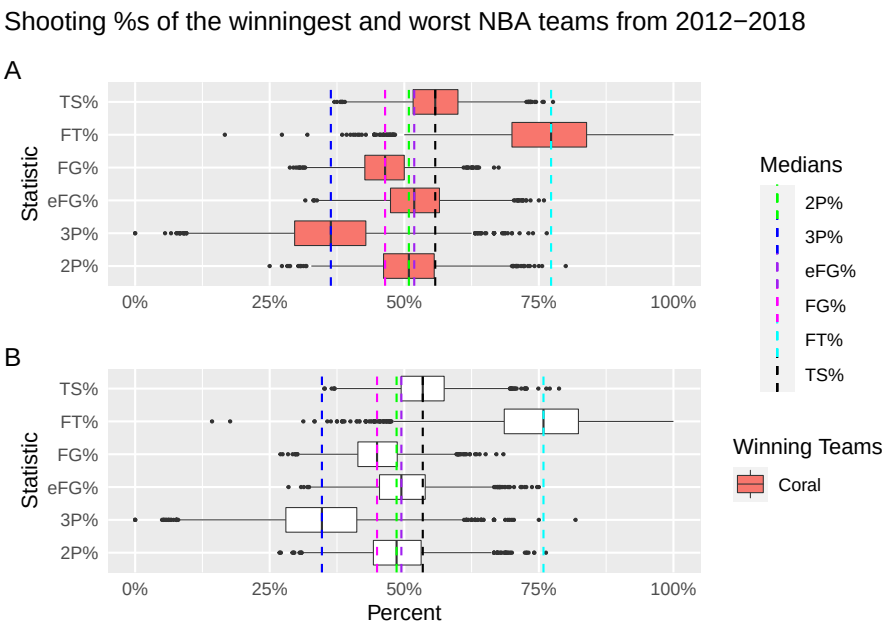

Figure 3: A comparison looking at the different shooting metrics of the best 5 and worst 5 NBA teams.

Figure 3 looks entirely at shooting percentage metrics. Undoubtedly, the better you shoot, the more points you are able to score. We see that reflected in this plot: every shooting metric leans in the favour of the winning group. This really underscores the importance of shooting in today's NBA, highlighting its change from the Shaq-era of post play.

3P% and 2P%

However, given the advent of the three-point shot, I would have thought that 2P% wouldn't matter nearly as much. This might reflect the importance of finishing at the rim, a high percentage shot with similar value to the three-pointer (Romanowich, Bourret, & Vollmer, 2007). In the era where shooting is emphasized, interior finishing remains understated. Being able to hit contested layups can still allow you to be an effective, positive player. Examples of this include All-Stars Giannis Antetokounmpo, Ben Simmons, Joel Embiid, and Russell Westbrook all of whom have at some point in their careers shot 3s far below league average (NBA Advanced Stats, 2020).

```
plt_netrtg <- stats2 %>%
  # take only the wins
  filter(., team_rslt == "Win") %>%
  group_by(team_abbr) %>%
  # Add the median pace for each team as well as use the count function for the
  # number of wins per team
  summarise(., med_rtg = median(team_ediff),
            n = n()) %>%
  ggplot(aes(x = med_rtg, y = n, colour = team_abbr)) +
  geom_point() +
  labs(y = "Wins", x = "Net Rating", colour = "Team")
```

```r
pltpace <- stats2 %>%
  # take only the wins
  filter(., team_rslt == "Win") %>%
  group_by(team_abbr) %>%
  # Add the median pace for each team as well as use the count function for the
  # number of wins per team
  summarise(., med_pace = median(pace),
            n = n()) %>%
  ggplot(aes(x = med_pace, y = n, colour = team_abbr)) +
  geom_point() +
  labs(y = "Wins", x = "Pace", colour = "Team")
```

Modifications to the data: filter only the wins for use in plotting the wins as an axis. Group by team to calculate the wins and median net rating or median pace using the summarize function. Median net rating and median pace are taken instead of using all rating and pace entries for a simpler plot.

```
plt_netrtg / pltpace +
  plot_annotation("Team pace and rating correlated to wins",
                  tag_levels = c("A", "B")) +
  plot_layout(guides = "collect")
```

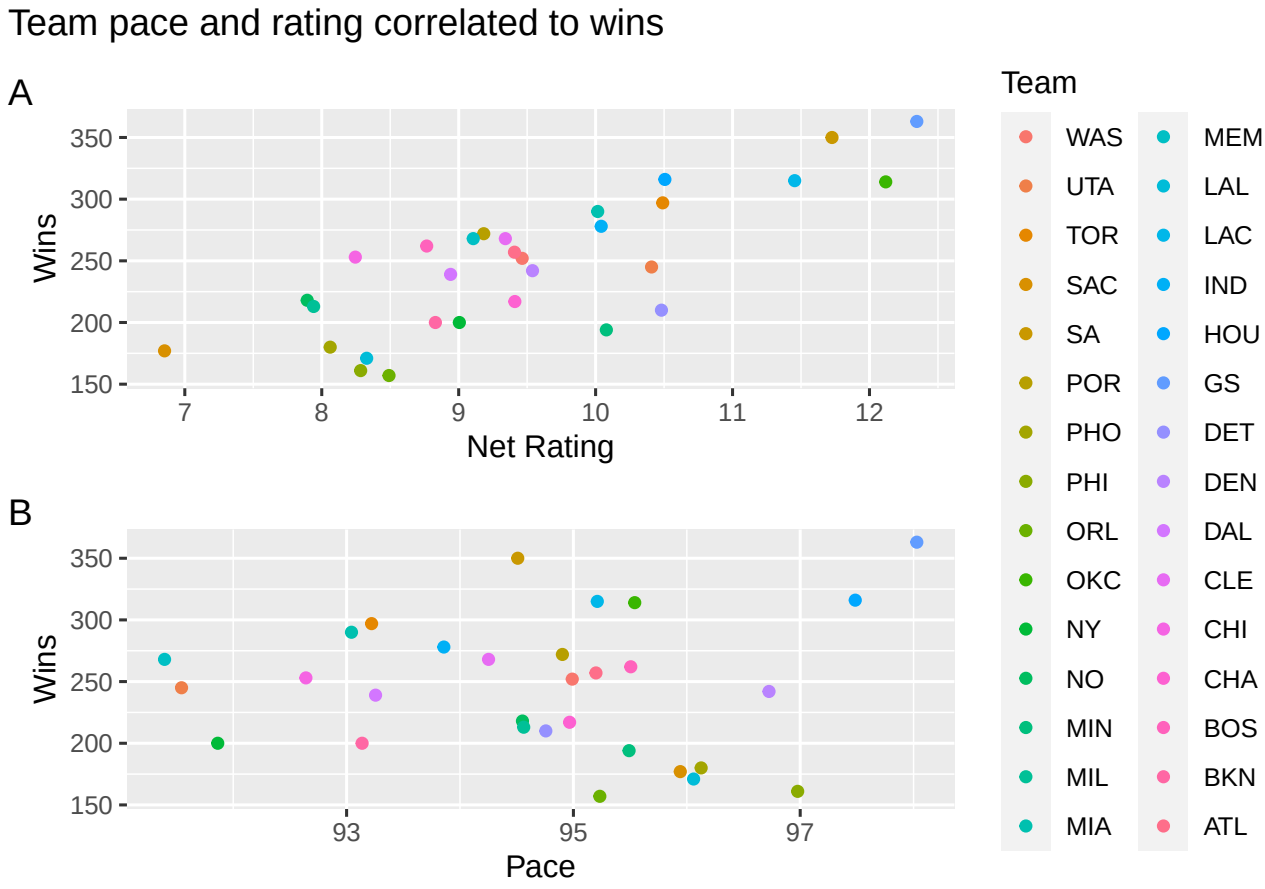

Figure 4: Median pace and Net Rating of each team over 2012-2018 graphed against amount of wins.

Net Rating

From the plot, a trend of net rating to wins is visible. This indicates this is a reliable metric for predicting team success. Net rating is calculated as $ORTG - DRTG$ (Basketball-Reference, 2004). However, when plotting ORTG and DRTG on their own, no correlation is seen. This means teams can be exceptional at offense and average defensively, but still be one of the best teams in the league.

Pace

Looking at the final important stat: pace of the game, we don't see a relationship between pace and winning. Some teams prefer to play slow to optimize their shots, where other teams like to wear out their opponents and catch them in easy transition buckets. Both stratgeies work as seen above; this game plan is often based on the speeds, skills, and play styles of your players.

Question 2

How has the understanding of these winning stats changed as coaches game plan differently based on the knowledge of mistakes in previous years? (ex., do teams shoot more threes now?)

```
plt1 <- stats2 %>%
  # Select certain stats and time
  select(., dttm, team_ast, team_orb, team_drb, team_stl, team_blk, team_to) %>%
  # Change names for better facet labels
  setNames(toupper(gsub("team_", "", names(.)))) %>%
  # Make the statistics all one column for faceting by statistic
  pivot_longer(., c(AST, ORB, DRB, STL, BLK, TO),
               names_to = "stat",
               values_to = "val") %>%
  ggplot() +
  # Plot loess line
  geom_smooth(aes(x = DTTM, y = val)) +
  # Display only the year on the axis
  scale_x_datetime(date_labels = "%Y") +
  # 45 degree angle x axis labs
  guides(x = guide_axis(angle = 45)) +
  # Plot graphs for each stat in 6 columns
  facet_wrap(.~stat, ncol=6) +
  labs(x = "Time", y = "Value")

plt2 <- stats2 %>%
  # Comments are the same as above, just selecting different stats.
  select(., dttm, `team_fg%`, `team_ft%`, `team_2p%`, `team_3p%`, `team_ts%`,
         `team_efg%`) %>%
  setNames(toupper(gsub("team_", "", names(.)))) %>%
  pivot_longer(., c(`FG%`, `FT%`, `2P%`, `3P%`, `TS%`, `EFG%`),
               names_to = "stat",
               values_to = "val") %>%
  ggplot() +
  geom_smooth(aes(x = DTTM, y = val)) +
  scale_x_datetime(date_labels = "%Y") +
  scale_y_continuous(labels = percent) +
  guides(x = guide_axis(angle = 45)) +
  facet_wrap(.~stat, ncol=6) +
  labs(x = "Time", y = "Percentage")
```

Modifications to the data: select only certain stats that are desired for plotting. Make names uppercase and in the format of AST, BLK, STL, etc. for better labels on the plot. Pivot to put stats into one column for plotting and faceting purposes.

```
plt1/plt2 + plot_annotation("Trends in Basic and Shooting Statistics from 2012-2018",
                            tag_levels = c("A", "B"))
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```

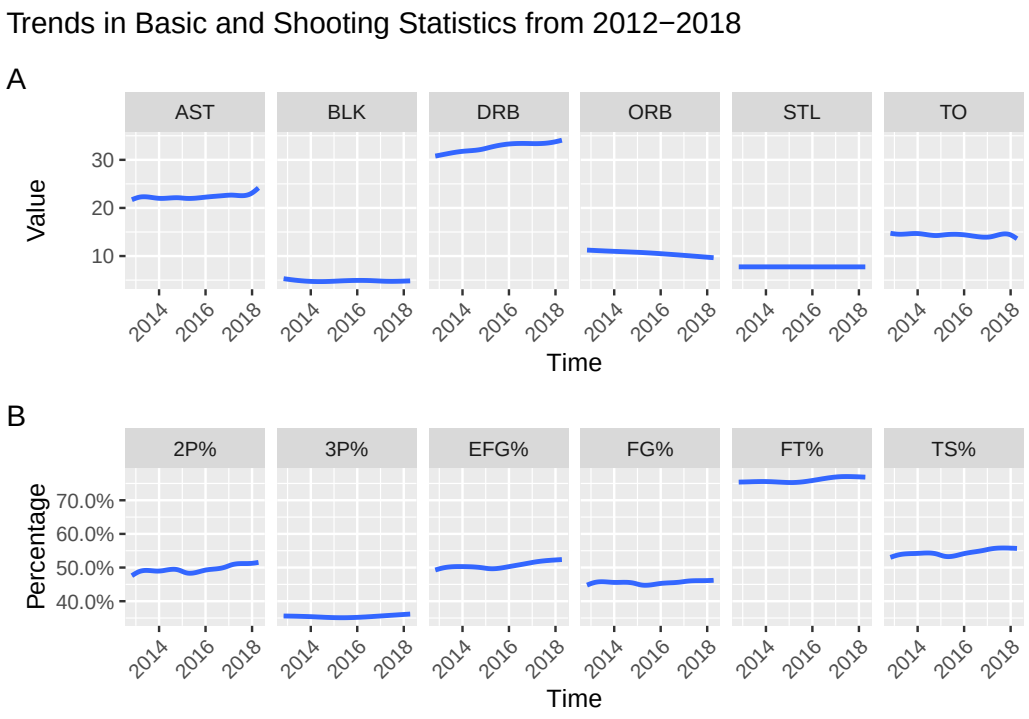

Figure 5: Loess regression of trends in certain statistics from 2012-2018.

From the data shown, there are some very conservative trends present. Metrics such as number of STLs and BLKs per game have largely stayed the same. However, given that the NBA has been around for 74 years, and this is a snapshot of only 6, this is not unexpected. Interestingly, many of the stats that winning teams possessed over losing teams in Figures 2 and 3 show increases over time.

AST

Assists have enjoyed a small increase since 2012, especially trending upwards in recent years towards the end of the 2018-19 season. This makes sense: assists and finding the optimal shot has been shown to be positively correlated to winning. (Melnick, 2001)

DRB

Steadily rising across the years is DRB. As previously mentioned, DRBs may be an indicator of better defense throughout the years. More misses lead to more DRBs, and we see an increase in defensive boards (Masheswaran et al, 2014). With the top teams nowadays boasting stellar defenses over offenses (NBA Advanced Statistics, 2020), the growing development and importance of defense should not be understated.

ORB

We can see that ORBs are decreasing throughout the years, perhaps indicating that teams have begun to recognize that being coordinated on defense is more important. It seems coaches have began to include this more in their game plans. The decline of the "big man" in the NBA may also play a factor, with many teams without traditional centres, getting offensive rebounds is more difficult.

2P% and 3P%

Interestingly, 2-pointers are being made more often. This may be due to the increased spacing of today's game, so it's easier to hit midrange shots and layups because teams are guarding the 3-point line more.

Surprisingly, 3-point % has not risen. This may be due to better coverages of 3s as the defense evolves. However, a look at a plot summarizing attempted shots shows definitely there have been more 3s attempted, followed by an increase in 3PM.

```
plt3 <- stats2 %>%
  # Comments are the same as Figure 4 (vars: plt1, plt2), just selecting different stats.
  select(., dttm, team_3pa, team_3pm, team_2pa, team_2pm, team_fta, team_ftm) %>%
  setNames(toupper(gsub("team_", "", names(.)))) %>%
  pivot_longer(., c(`3PA`, `3PM`, `2PA`, `2PM`, `FTA`, `FTM`),
               names_to = "stat",
               values_to = "val") %>%
  ggplot() +
  geom_smooth(aes(x = DTTM, y = val)) +
  scale_x_datetime(date_labels = "%Y") +
  guides(x = guide_axis(angle = 45)) +
  facet_wrap(.~stat, ncol=6) +
  labs(x = "Time", y = "Value")
```

Modifications to the data: the same as in Figure 5, just selecting different stats to be included. The reasoning and the usefulness behind these modifications can also be found as previously stated.

```
plt3 + plot_annotation("Types of shots attempted from 2012-2018")
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```

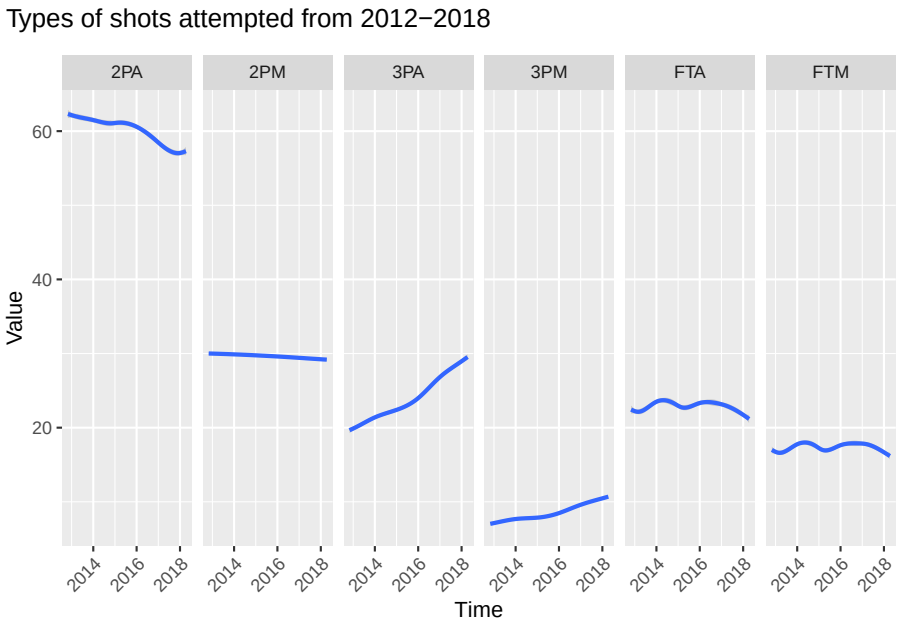

Figure 6: Attempted shots over time.

The Advent of the 3-pointer?

It's plain from the above that the league has been shooting and making more threes. It's well known that 3-pointers elicit more value from shots than 2-pointers (Marty & Lucey, 2017). However, from the data above, taking a league-median 3-point shooter (~35% in 2018 from Fig. 5) and 2-point shooter (~52% in 2018) gives the following:

$0.36 * 3 = 1.08$ points per shot $0.52 * 2 = 1.04$ points per shot

0.04 points per shot is not a big difference! Of course, in most games, almost 100 shots are taken, and games are often decided by a point. Still, threes are important, but it should not be understated that 2-point shots can still be valuable contributors to winning.

Spacing

As we can also see, a decrease in 2PA does not indicate similarly less 2PM, indicating that as a league, it is has become easier to score 2P shots. This again can be possibly attributed to the spacing of today's game. Giannis and the Bucks is a prime exaple of this: the spacing of the countless 3-point shooters allows Giannis to wreck inside. Resultantly, he maintains a well above league average career 52.6% FG (which includes 2P% and 3P%) while having an abysmal 28.5% 3P%. (NBA Advanced Statistics, 2020)

FT%, FTA, FTM

Interestingly, there is a decrease in FTA and FTM. Getting to the line can often be more valuable than a shot attempt: using 2018 league median FT% form Figure 5,

$0.36 * 3 = 1.08$ points per shot (3s) $0.52 * 2 = 1.04$ points per shot (2s) $0.75 * 2 = 1.50$ points per shot (FT)

This explains the increase in FT% makes sense: players would want to convert these "free" shots. The decrease in FTA, and resultantly FTM, might be attributed to coaches and players fouling less. Given the power of free throws, avoiding handing them out to opposing players is wise.

Discussion and Conclusion

Ultimately, analysis of NBA data sourced from Kaggle has led to some valuable insights on facets of the game. From this data, we have found some predicted as well as unlikely answers to our research questions. To recapitualte:

- What statistics do winning teams share that contributes to their success the most?
- How has the understanding of these winning stats changed as coaches game plan differently based on the knowledge of mistakes in previous years? (ex., do teams shoot more threes now?)

We found that offensive rebounds have been declining in the league and are not correlated with team success, while better shooting clearly is related to winning, and has been improving within the league. In my analysis, increases in defensive rebounds (which may be indicative of defensive prowess), blocks, and decreases in turnovers are factors indicative of winning, and following suit from 2012-2018 these metrics increased/decreased appropriately. The league has been attempting and making more threes, leaving increased spacing for easier 2-pointers.

However, there are major limitations to this project. In figures 2 and 3, only 20/30 NBA teams were sampled, and compared solely on median values. Plotting different stats on the same axis may have diminished trends, but STLs, BLKs, and TOs were also plotted independently revealing no major changes in trend. These are not shown for report brevity. Future work would look to more sophisticatedly statistically analyze the graphs plotted as well as to delve deeper into advanced stats to explain speculation and ideas held in this report. Regardless, it's clear that coaches and GMs alike have used statistical analysis to their advantage when building the perfect game plans to winning – an homage to the power of languages like R and its applicability in real life.

References

2013-14 Los Angeles Clippers Roster and Stats. (n.d.). Basketball-Reference.Com. Retrieved April 6, 2020, from https://www.basketball-reference.com/teams/LAC/2014.html

Applying multiple functions to data frame | R-bloggers. (n.d.). Retrieved April 6, 2020, from https://www.r-bloggers.com/applying-multiple-functions-to-data-frame/

Data Visualization. (2016). Retrieved April 6, 2020, from https://socviz.co/lookatdata.html#why-look-at-data

Franks, A., Miller, A., Bornn, L., Goldsberry K. (2015). Retrieved April 6, 2020, from https://pdfs.semanticscholar.org/1016/c66483e546eee19e0f1a5bdc811876950158.pdf

Glossary. (2004). Basketball-Reference.Com. Retrieved April 6, 2020, from https://www.basketball-reference.com/about/glossary.html

Grolemund, G., & Wickham, H. (2020). R for Data Science. Retrieved April 6, 2020, from https://r4ds.had.co.nz/

Hofner, B. Using papeR with Markdown. (2020). Retrieved April 6, 2020, from https://cran.r-project.org/web/packages/papeR/vignettes/papeR_introduction.html

HTHSCI 1M03: Introdution to Data Science. (2020). Retrieved April 6, 2020, from https://ptaitatmcmaster.github.io/HTHSCI_1M03_W2020/

Irizarry, R. A. (2019). Introduction to data science: Data analysis and prediction algorithms with r. CRC Press.

Marty, R., & Lucey, S. (2017). Retrieved April 6, 2020, from http://www.sloansportsconference.com/wp-content/uploads/2017/02/1505.pdf

Melnick, M. J. (2001). Relationship between Team Assists and Win-Loss Record in the National Basketball Association. Perceptual and Motor Skills, 92(2), 595–602. https://doi.org/10.2466/pms.2001.92.2.595

Mikołajec, K., Maszczyk, A., & Zając, T. (2013). Game Indicators Determining Sports Performance in the NBA. Journal of Human Kinetics, 37(1), 145–151. https://doi.org/10.2478/hukin-2013-0035

NBA Advanced Stats. (2020). NBA.com/Stats. Retrieved April 6, 2020, from https://stats.nba.com/

Romanowich, P., Bourret, J., & Vollmer, T. R. (2007). Further Analysis of the Matching Law to Describe Two- and Three-Point Shot Allocation by Professional Basketball Players. Journal of Applied Behavior Analysis, 40(2), 311–315. https://doi.org/10.1901/jaba.2007.119-05

Silver, N. (2019, July 9). A Better Way To Evaluate NBA Defense. FiveThirtyEight. https://fivethirtyeight.com/features/a-better-way-to-evaluate-nba-defense/

Teams Offensive Rebounding. (n.d.). NBA Stats. Retrieved April 6, 2020, from https://stats.nba.com/teams/offensive-rebounding/

Transition defense has left offensive rebounds on the cutting room floor. (n.d.). Retrieved April 6, 2020, from https://www.espn.com/nba/story/_/id/14505051/transition-defense-left-offensive-rebounds-cutting-room-floor