R Notebook

In order to install this package you must have devtools installed. To do so type this into your console: install.packages("devtools") Once devtools is installed you will need to pull this package from github by typing this into your console: devtools::install_github("PatrickChodowski/NBAr") once this is installed all you have to do is include NBAr in your library like I have done below by running: library(NBAr) These functions also require you to have the tidyverse library installed as well

```
library(NBAr)
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 3.6.3
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.1.0     v dplyr   1.0.4
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1
```

```
## Warning: package 'ggplot2' was built under R version 3.6.3
```

```
## Warning: package 'tibble' was built under R version 3.6.3
```

```
## Warning: package 'tidyr' was built under R version 3.6.3
```

```
## Warning: package 'readr' was built under R version 3.6.3
```

```
## Warning: package 'purrr' was built under R version 3.6.3
```

```
## Warning: package 'dplyr' was built under R version 3.6.3
```

```
## Warning: package 'forcats' was built under R version 3.6.3
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

#Example data

#sample variables

```
season <- 2018
gamelist <- c(21800001:21800003)
game_id <- 21800001
```

#Season variable represents the season we want to look at. #amelist is a list of game id numbers according to the NBA website #game id is a single game id number according to the NBA website

#get_boxscores() ## @param game_id Game's ID in NBA.com DB ## @param boxscore_type Type of boxscore, one from the list: c("advanced","misc","playertrack","traditional") ## @param verbose Defalt TRUE - prints additional information

```
traditional_multiple_games <- map(gamelist, ~get_boxscore(.,boxscore_type = 'traditional')) %>% compact
```

```
## https://stats.nba.com/stats/boxscoretraditionalv2?EndPeriod=10&EndRange=28800&GameID=0021800001&Rang

## Warning in .parse_hms(..., order = "MS", quiet = quiet): Some strings failed to
## parse, or all strings are NAs

## Warning in .parse_hms(..., order = "MS", quiet = quiet): Some strings failed to
## parse, or all strings are NAs

## Rows: 26
## Columns: 29
## $ game_id           <int> 21800001, 21800001, 21800001, 21800001, 21800001, 21~
## $ team_id           <int> 1610612755, 1610612755, 1610612755, 1610612755, 1610~
## $ team_abbreviation <chr> "PHI", "PHI", "PHI", "PHI", "PHI", "PHI", "PHI", "PH~
## $ team_city         <chr> "Philadelphia", "Philadelphia", "Philadelphia", "Phi~
## $ player_id         <int> 203967, 203496, 203954, 1628365, 1627732, 101161, 20~
## $ player_name       <chr> "Dario Saric", "Robert Covington", "Joel Embiid", "M~
## $ start_position    <chr> "F", "F", "C", "G", "G", "", "", "", "", "", "", "",~
## $ comment           <chr> "", "", "", "", "", "", "", "", "", "", "", "DNP - C~
## $ fgm               <int> 3, 3, 9, 2, 7, 1, 7, 2, 0, 0, 0, NA, NA, 9, 4, 4, 5,~
## $ fga               <int> 8, 10, 21, 7, 14, 1, 17, 5, 4, 0, 0, NA, NA, 17, 12,~
## $ fg_pct            <dbl> 0.375, 0.300, 0.429, 0.286, 0.500, 1.000, 0.412, 0.4~
## $ fg3m              <int> 0, 2, 1, 0, 0, 0, 2, 0, 0, 0, 0, NA, NA, 1, 1, 0, 1,~
## $ fg3a              <int> 4, 7, 4, 0, 0, 0, 8, 1, 2, 0, 0, NA, NA, 5, 3, 1, 3,~
## $ fg3_pct           <dbl> 0.000, 0.286, 0.250, 0.000, 0.000, 0.000, 0.250, 0.0~
## $ ftm               <int> 0, 0, 4, 1, 5, 3, 0, 0, 1, 0, 0, NA, NA, 4, 1, 1, 1,~
## $ fta               <int> 0, 0, 5, 2, 10, 4, 0, 0, 2, 0, 0, NA, NA, 5, 2, 1, 2~
## $ ft_pct            <dbl> 0.00, 0.00, 0.80, 0.50, 0.50, 0.75, 0.00, 0.00, 0.50~
## $ oreb              <int> 0, 1, 2, 0, 3, 0, 0, 0, 0, 0, 0, NA, NA, 2, 2, 0, 3,~
## $ dreb              <int> 6, 5, 8, 3, 12, 3, 2, 1, 1, 0, 0, NA, NA, 7, 3, 4, 2~
## $ reb               <int> 6, 6, 10, 3, 15, 3, 2, 1, 1, 0, 0, NA, NA, 9, 5, 4, ~
## $ ast               <int> 1, 0, 2, 2, 8, 1, 1, 3, 0, 0, 0, NA, NA, 3, 0, 2, 2,~
## $ stl               <int> 0, 2, 1, 1, 4, 0, 0, 0, 0, 0, 0, NA, NA, 1, 4, 0, 0,~
## $ blk               <int> 0, 1, 2, 0, 2, 0, 0, 0, 0, 0, 0, NA, NA, 0, 0, 4, 0,~
## $ to                <int> 3, 2, 5, 3, 3, 0, 0, 0, 0, 0, 0, NA, NA, 1, 0, 3, 2,~
## $ pf                <int> 5, 1, 3, 1, 5, 1, 1, 3, 0, 0, 0, NA, NA, 2, 1, 1, 4,~
## $ pts               <int> 6, 8, 23, 5, 19, 5, 16, 4, 1, 0, 0, NA, NA, 23, 10, ~
## $ plus_minus        <int> -4, -11, -8, -16, -8, -10, -10, -11, -8, -2, -2, NA,~
## $ mins              <int> 22, 34, 36, 24, 42, 11, 29, 22, 12, 1, 1, NA, NA, 28~
## $ secs              <int> 54, 13, 49, 20, 44, 11, 46, 26, 43, 27, 27, NA, NA, ~
## https://stats.nba.com/stats/boxscoretraditionalv2?EndPeriod=10&EndRange=28800&GameID=0021800002&Rang
```

```
## Warning in .parse_hms(..., order = "MS", quiet = quiet): Some strings failed to
## parse, or all strings are NAs

## Warning in .parse_hms(..., order = "MS", quiet = quiet): Some strings failed to
## parse, or all strings are NAs

## Rows: 26
## Columns: 29
## $ game_id           <int> 21800002, 21800002, 21800002, 21800002, 21800002, 21~
## $ team_id           <int> 1610612760, 1610612760, 1610612760, 1610612760, 1610~
## $ team_abbreviation <chr> "OKC", "OKC", "OKC", "OKC", "OKC", "OKC", "OKC", "OK~
## $ team_city         <chr> "Oklahoma City", "Oklahoma City", "Oklahoma City", "~
## $ player_id         <int> 202331, 202335, 203500, 1628390, 203471, 203924, 203~
## $ player_name       <chr> "Paul George", "Patrick Patterson", "Steven Adams", ~
## $ start_position    <chr> "F", "F", "C", "G", "G", "", "", "", "", "", "", "",~
## $ comment           <chr> "", "", "", "", "", "", "", "", "", "", "DNP - Coach~
## $ fgm               <int> 9, 2, 6, 0, 7, 2, 3, 1, 1, 2, NA, NA, NA, 9, 1, 6, 5~
## $ fga               <int> 23, 9, 12, 2, 19, 7, 8, 5, 2, 4, NA, NA, NA, 21, 6, ~
## $ fg_pct            <dbl> 0.391, 0.222, 0.500, 0.000, 0.368, 0.286, 0.375, 0.2~
## $ fg3m              <int> 4, 1, 0, 0, 2, 1, 2, 0, 0, 0, NA, NA, NA, 0, 0, 0, 1~
## $ fg3a              <int> 12, 4, 0, 2, 6, 4, 6, 3, 0, 0, NA, NA, NA, 5, 1, 0, ~
## $ fg3_pct           <dbl> 0.333, 0.250, 0.000, 0.000, 0.333, 0.250, 0.333, 0.0~
## $ ftm               <int> 5, 2, 5, 0, 5, 2, 0, 4, 1, 0, NA, NA, NA, 9, 0, 0, 3~
## $ fta               <int> 8, 3, 8, 0, 6, 4, 0, 5, 2, 1, NA, NA, NA, 10, 0, 0, ~
## $ ft_pct            <dbl> 0.625, 0.667, 0.625, 0.000, 0.833, 0.500, 0.000, 0.8~
## $ oreb              <int> 0, 3, 4, 2, 2, 2, 0, 0, 3, 0, NA, NA, NA, 1, 1, 2, 1~
## $ dreb              <int> 2, 2, 7, 2, 6, 0, 2, 3, 4, 1, NA, NA, NA, 7, 12, 1, ~
## $ reb               <int> 2, 5, 11, 4, 8, 2, 2, 3, 7, 1, NA, NA, NA, 8, 13, 3,~
## $ ast               <int> 5, 0, 4, 1, 6, 2, 0, 1, 1, 1, NA, NA, NA, 6, 5, 2, 0~
## $ stl               <int> 4, 1, 2, 0, 3, 0, 0, 0, 1, 1, NA, NA, NA, 1, 3, 0, 0~
## $ blk               <int> 0, 1, 0, 0, 1, 3, 0, 0, 1, 0, NA, NA, NA, 1, 0, 3, 0~
## $ to                <int> 5, 1, 2, 1, 3, 0, 0, 2, 0, 0, NA, NA, NA, 3, 6, 2, 2~
## $ pf                <int> 3, 0, 3, 3, 4, 1, 2, 0, 3, 2, NA, NA, NA, 4, 3, 4, 3~
## $ pts               <int> 27, 7, 17, 0, 21, 7, 8, 6, 3, 4, NA, NA, NA, 27, 2, ~
## $ plus_minus        <int> -16, 3, -6, 0, -7, -11, -5, -1, -2, 5, NA, NA, NA, 1~
## $ mins              <int> 37, 18, 35, 26, 34, 29, 23, 13, 12, 8, NA, NA, NA, 3~
## $ secs              <int> 24, 54, 36, 30, 5, 6, 29, 55, 24, 37, NA, NA, NA, 43~
## https://stats.nba.com/stats/boxscoretraditionalv2?EndPeriod=10&EndRange=28800&GameID=0021800003&Rang

## Warning in .parse_hms(..., order = "MS", quiet = quiet): Some strings failed to
## parse, or all strings are NAs

## Warning in .parse_hms(..., order = "MS", quiet = quiet): Some strings failed to
## parse, or all strings are NAs

## Rows: 26
## Columns: 29
## $ game_id           <int> 21800003, 21800003, 21800003, 21800003, 21800003, 21~
## $ team_id           <int> 1610612749, 1610612749, 1610612749, 1610612749, 1610~
## $ team_abbreviation <chr> "MIL", "MIL", "MIL", "MIL", "MIL", "MIL", "MIL", "MI~
## $ team_city         <chr> "Milwaukee", "Milwaukee", "Milwaukee", "Milwaukee", ~
## $ player_id         <int> 203114, 203507, 201572, 1627763, 202339, 1628978, 10~
## $ player_name       <chr> "Khris Middleton", "Giannis Antetokounmpo", "Brook L~
```

```
## $ start_position    <chr> "F", "F", "C", "G", "G", "", "", "", "", "", "", "",~
## $ comment           <chr> "", "", "", "", "", "", "", "", "", "", "", "DNP - C~
## $ fgm               <int> 7, 9, 4, 4, 7, 3, 6, 1, 0, 1, 0, NA, NA, 2, 2, 1, 4,~
## $ fga               <int> 14, 21, 7, 9, 13, 6, 9, 3, 2, 1, 0, NA, NA, 8, 7, 1,~
## $ fg_pct            <dbl> 0.500, 0.429, 0.571, 0.444, 0.538, 0.500, 0.667, 0.3~
## $ fg3m              <int> 3, 0, 2, 2, 3, 2, 1, 0, 0, 1, 0, NA, NA, 1, 1, 1, 2,~
## $ fg3a              <int> 6, 3, 5, 6, 5, 4, 3, 0, 1, 1, 0, NA, NA, 4, 6, 1, 4,~
## $ fg3_pct           <dbl> 0.500, 0.000, 0.400, 0.333, 0.600, 0.500, 0.333, 0.0~
## $ ftm               <int> 2, 7, 4, 0, 0, 0, 0, 2, 0, 0, 0, NA, NA, 0, 0, 2, 0,~
## $ fta               <int> 2, 12, 4, 0, 0, 0, 0, 2, 0, 0, 0, NA, NA, 0, 2, 2, 0~
## $ ft_pct            <dbl> 1.000, 0.583, 1.000, 0.000, 0.000, 0.000, 0.000, 1.0~
## $ oreb              <int> 1, 3, 0, 0, 2, 1, 1, 2, 1, 0, 0, NA, NA, 1, 1, 0, 2,~
## $ dreb              <int> 8, 15, 3, 3, 2, 1, 8, 4, 1, 1, 0, NA, NA, 6, 8, 3, 0~
## $ reb               <int> 9, 18, 3, 3, 4, 2, 9, 6, 2, 1, 0, NA, NA, 7, 9, 3, 2~
## $ ast               <int> 4, 8, 1, 6, 3, 1, 2, 0, 1, 0, 0, NA, NA, 3, 1, 1, 1,~
## $ stl               <int> 2, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, NA, NA, 2, 0, 1, 1,~
## $ blk               <int> 1, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, NA, NA, 2, 4, 0, 0,~
## $ to                <int> 2, 8, 2, 4, 2, 1, 0, 2, 0, 0, 0, NA, NA, 2, 1, 0, 0,~
## $ pf                <int> 3, 5, 3, 3, 4, 0, 6, 1, 0, 0, 0, NA, NA, 1, 2, 3, 2,~
## $ pts               <int> 19, 25, 14, 10, 17, 8, 13, 4, 0, 3, 0, NA, NA, 5, 5,~
## $ plus_minus        <int> 2, 11, 8, 1, 4, -9, -10, 2, 1, -3, -2, NA, NA, -4, -~
## $ mins              <int> 32, 34, 21, 38, 33, 22, 28, 12, 11, 4, 0, NA, NA, 31~
## $ secs              <int> 46, 46, 27, 0, 32, 5, 59, 1, 20, 57, 7, NA, NA, 49, ~
```

```
traditional <- get_boxscore(game_id, 'tradtitional')
```

```
## https://stats.nba.com/stats/boxscoretradtitionalv2?EndPeriod=10&EndRange=28800&GameID=0021800001&Rang
## [1] "lexical error: invalid char in json text.\n                                                    <!DOCTYPE html
```

##The traditional multiple games allows you to create a table of boxscores froma list of games. In the example it is a list of three games. The function get_boxscores wont let you do this by itself which is why the map is necessary. ##The second is the use of the function on its own which takes in one game ID and returns the boxscore. ##Boxscore types include: advanced, misc, playertrack, and traditional.

#get_team_boxscore() ## @param season Number of the year in which season started ## @param measure_type Specify type of data c('Base','Advanced','Four+Factors','Misc','Scoring') ## @param season_type Choose data for preseason, regular season or postseason. Default parameter is "Regular+Season". c("Regular+Season","Playoffs","Pre+Season","All+Star") ## @param game_segment Choose game half for the data. Empty string means whole game and it is set by default. c("","First+Half","Overtime","Second+Half") # @param period Choose game period for the data. 0 means whole game and it is set by default. as.character(c(0:4)) ## @param date_from Day from which data will be collected. It is set in MM/DD/YYYY format and by default is not specified, so data is calculated for whole season. ## @param date_to Day to which data will be collected. It is set in MM/DD/YYYY format and by default is not specified, so data is calculated for whole season. ## @param outcome Filter by game result. It can be a loss (L) or a win (W). By default parameter is an empty string, so both are taken into account. c("","W","L") ## @param opponent_team_id Filter by opponent's team id from nba.com database. Default"0" means all teams. ## @param verbose Defalt TRUE - prints additional information

```
team_boxscores_all <- get_team_boxscore(season = 2018, date_from = '10/01/2018', date_to = '12/03/2018'
```

```
## https://stats.nba.com/stats/teamgamelogs?DateFrom=10/01/2018&DateTo=12/03/2018&GameSegment=&LastNGame
## Rows: 349
## Columns: 48
```

```
## $ season_year       <chr> "2018-19", "2018-19", "2018-19", "2018-19", "2018-19~
## $ team_id           <int> 1610612764, 1610612744, 1610612746, 1610612739, 1610~
## $ team_abbreviation <chr> "WAS", "GSW", "LAC", "CLE", "OKC", "DEN", "MIN", "LA~
## $ team_name         <chr> "Washington Wizards", "Golden State Warriors", "LA C~
## $ game_id           <int> 21800346, 21800344, 21800349, 21800345, 21800343, 21~
## $ game_date         <chr> "2018-12-03T00:00:00", "2018-12-03T00:00:00", "2018-~
## $ matchup           <chr> "WAS @ NYK", "GSW @ ATL", "LAC @ NOP", "CLE @ BKN", ~
## $ wl                <chr> "W", "W", "W", "W", "W", "W", "W", "W", "W", "W", "W~
## $ min               <int> 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, ~
## $ e_off_rating      <int> 110, 121, 127, 106, 105, 101, 108, 109, 108, 112, 10~
## $ off_rating        <int> 112, 119, 129, 105, 106, 102, 108, 113, 106, 115, 10~
## $ e_def_rating      <int> 107, 100, 127, 100, 75, 100, 95, 89, 102, 105, 100, ~
## $ def_rating        <int> 109, 103, 126, 104, 82, 99, 97, 90, 103, 103, 104, 9~
## $ e_net_rating      <int> 3, 20, 0, 6, 29, 0, 12, 19, 6, 7, 1, 8, 14, 10, 0, 2~
## $ net_rating        <int> 3, 15, 3, 1, 24, 3, 10, 22, 2, 11, 3, 6, 14, 7, 1, 0~
## $ ast_pct           <dbl> 0.682, 0.688, 0.592, 0.395, 0.535, 0.800, 0.611, 0.5~
## $ ast_to            <int> 3, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 1, 1, 1, 1~
## $ ast_ratio         <int> 21, 22, 20, 12, 17, 20, 16, 17, 13, 20, 15, 17, 21, ~
## $ oreb_pct          <dbl> 0.240, 0.222, 0.270, 0.320, 0.227, 0.327, 0.367, 0.3~
## $ dreb_pct          <dbl> 0.589, 0.729, 0.761, 0.672, 0.738, 0.809, 0.786, 0.8~
## $ reb_pct           <dbl> 0.425, 0.512, 0.542, 0.518, 0.532, 0.549, 0.560, 0.5~
## $ tm_tov_pct        <dbl> 0.102, 0.187, 0.120, 0.106, 0.165, 0.194, 0.147, 0.1~
## $ efg_pct           <dbl> 0.549, 0.665, 0.639, 0.455, 0.547, 0.511, 0.506, 0.5~
## $ ts_pct            <dbl> 0.565, 0.698, 0.670, 0.511, 0.586, 0.532, 0.546, 0.5~
## $ e_pace            <int> 99, 108, 99, 94, 107, 103, 95, 108, 106, 104, 100, 1~
## $ pace              <int> 98, 107, 100, 93, 102, 103, 94, 106, 106, 104, 95, 1~
## $ pace_per40        <dbl> 81.67, 89.17, 83.33, 77.92, 85.00, 86.25, 78.33, 88.~
## $ poss              <int> 98, 107, 100, 94, 103, 103, 95, 106, 107, 103, 95, 1~
## $ pie               <dbl> 0.529, 0.606, 0.509, 0.578, 0.710, 0.501, 0.581, 0.6~
## $ gp_rank           <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1~
## $ w_rank            <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1~
## $ l_rank            <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1~
## $ w_pct_rank        <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1~
## $ min_rank          <int> 22, 22, 22, 22, 22, 22, 22, 22, 22, 22, 22, 22, 22, ~
## $ off_rating_rank   <int> 205, 98, 23, 282, 265, 307, 241, 187, 271, 146, 256,~
## $ def_rating_rank   <int> 253, 180, 349, 194, 8, 119, 95, 42, 182, 184, 191, 6~
## $ net_rating_rank   <int> 290, 86, 293, 330, 29, 276, 148, 39, 305, 134, 288, ~
## $ ast_pct_rank      <dbl> 57, 54, 166, 337, 259, 3, 136, 215, 311, 57, 232, 13~
## $ ast_to_rank       <int> 17, 201, 53, 242, 276, 217, 229, 274, 294, 196, 235,~
## $ ast_ratio_rank    <int> 41, 23, 62, 340, 207, 60, 241, 184, 321, 67, 286, 20~
## $ oreb_pct_rank     <dbl> 246, 274, 194, 100, 265, 82, 39, 124, 82, 228, 149, ~
## $ dreb_pct_rank     <dbl> 341, 189, 127, 297, 173, 48, 82, 23, 23, 123, 241, 1~
## $ reb_pct_rank      <dbl> 347, 187, 96, 163, 121, 81, 65, 28, 32, 113, 279, 55~
## $ tm_tov_pct_rank   <dbl> 37, 316, 86, 49, 265, 324, 199, 304, 209, 291, 164, ~
## $ efg_pct_rank      <dbl> 175, 11, 30, 337, 179, 255, 262, 149, 306, 114, 327,~
## $ ts_pct_rank       <dbl> 218, 7, 26, 325, 167, 295, 266, 162, 288, 105, 337, ~
## $ pace_rank         <int> 245, 39, 206, 329, 142, 109, 327, 58, 48, 95, 305, 1~
## $ pie_rank          <dbl> 265, 72, 309, 131, 7, 322, 121, 62, 276, 206, 335, 1~
```

```
team_boxscores <- get_team_boxscore(season)
```

```
## https://stats.nba.com/stats/teamgamelogs?DateFrom=&DateTo=&GameSegment=&LastNGames=&LeagueID=00&Loca
## Rows: 2,460
## Columns: 56
```

```
## $ season_year       <chr> "2018-19", "2018-19", "2018-19", "2018-19", "2018-19~
## $ team_id           <int> 1610612762, 1610612737, 1610612752, 1610612753, 1610~
## $ team_abbreviation <chr> "UTA", "ATL", "NYK", "ORL", "DAL", "LAC", "MIA", "GS~
## $ team_name         <chr> "Utah Jazz", "Atlanta Hawks", "New York Knicks", "Or~
## $ game_id           <int> 21801229, 21801220, 21801223, 21801222, 21801227, 21~
## $ game_date         <chr> "2019-04-10T00:00:00", "2019-04-10T00:00:00", "2019-~
## $ matchup           <chr> "UTA @ LAC", "ATL vs. IND", "NYK vs. DET", "ORL @ CH~
## $ wl                <chr> "L", "L", "L", "W", "L", "W", "L", "L", "W", "L", "L~
## $ min               <int> 53, 48, 48, 48, 48, 53, 48, 48, 48, 48, 48, 48, 48, ~
## $ fgm               <int> 47, 43, 31, 48, 37, 54, 38, 46, 43, 41, 45, 45, 39, ~
## $ fga               <int> 106, 103, 77, 88, 91, 106, 98, 92, 114, 78, 95, 98, ~
## $ fg_pct            <dbl> 0.443, 0.417, 0.403, 0.545, 0.407, 0.509, 0.388, 0.5~
## $ fg3m              <int> 14, 17, 9, 11, 11, 12, 8, 13, 21, 8, 9, 12, 10, 18, ~
## $ fg3a              <int> 35, 41, 30, 30, 37, 28, 41, 30, 58, 29, 24, 30, 33, ~
## $ fg3_pct           <dbl> 0.400, 0.415, 0.300, 0.367, 0.297, 0.429, 0.195, 0.4~
## $ ftm               <int> 29, 31, 18, 15, 9, 23, 10, 12, 6, 24, 10, 33, 11, 13~
## $ fta               <int> 33, 38, 25, 20, 12, 30, 15, 14, 7, 27, 15, 39, 17, 1~
## $ ft_pct            <dbl> 0.879, 0.816, 0.720, 0.750, 0.750, 0.767, 0.667, 0.8~
## $ oreb              <int> 17, 22, 4, 9, 7, 12, 8, 9, 20, 4, 9, 13, 12, 12, 9, ~
## $ dreb              <int> 40, 39, 29, 30, 35, 40, 46, 30, 48, 28, 30, 30, 41, ~
## $ reb               <int> 57, 61, 33, 39, 42, 52, 54, 39, 68, 32, 39, 43, 53, ~
## $ ast               <int> 31, 29, 23, 24, 27, 34, 22, 32, 29, 21, 21, 22, 23, ~
## $ tov               <int> 17, 17, 11, 6, 7, 12, 15, 14, 12, 8, 8, 8, 13, 10, 7~
## $ stl               <int> 8, 5, 7, 5, 5, 7, 9, 3, 7, 3, 4, 12, 6, 5, 4, 5, 6, ~
## $ blk               <int> 11, 7, 5, 2, 4, 5, 4, 5, 3, 4, 3, 7, 4, 1, 10, 2, 0,~
## $ blka              <int> 5, 7, 2, 4, 2, 11, 3, 0, 4, 2, 10, 7, 0, 2, 3, 5, 4,~
## $ pf                <int> 24, 25, 17, 19, 19, 27, 12, 17, 13, 14, 13, 28, 12, ~
## $ pfd               <int> 27, 28, 23, 14, 14, 24, 13, 18, 12, 19, 11, 25, 22, ~
## $ pts               <int> 137, 134, 89, 122, 94, 143, 94, 117, 113, 114, 109, ~
## $ plus_minus        <int> -6, -1, -26, 8, -11, 6, -19, -15, 19, -8, -16, 1, 4,~
## $ gp_rank           <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1~
## $ w_rank            <int> 1231, 1231, 1231, 1, 1231, 1, 1231, 1231, 1, 1231, 1~
## $ l_rank            <int> 1231, 1231, 1231, 1, 1231, 1, 1231, 1231, 1, 1231, 1~
## $ w_pct_rank        <dbl> 1231, 1231, 1231, 1, 1231, 1, 1231, 1231, 1, 1231, 1~
## $ min_rank          <int> 21, 135, 135, 135, 135, 21, 135, 135, 135, 135, 135,~
## $ fgm_rank          <int> 279, 760, 2380, 201, 1858, 25, 1678, 360, 760, 1130,~
## $ fga_rank          <int> 32, 75, 2334, 1301, 896, 32, 232, 755, 12, 2303, 432~
## $ fg_pct_rank       <dbl> 1527, 1967, 2122, 149, 2089, 454, 2266, 535, 2330, 3~
## $ fg3m_rank         <int> 462, 166, 1643, 1110, 1110, 864, 1907, 646, 23, 1907~
## $ fg3a_rank         <int> 728, 237, 1381, 1381, 534, 1645, 237, 1381, 8, 1510,~
## $ fg3_pct_rank      <dbl> 673, 564, 1804, 1049, 1828, 449, 2380, 427, 1130, 20~
## $ ftm_rank          <int> 93, 53, 1059, 1513, 2271, 421, 2177, 1967, 2408, 331~
## $ fta_rank          <int> 212, 73, 868, 1526, 2287, 387, 2083, 2157, 2440, 661~
## $ ft_pct_rank       <dbl> 324, 803, 1684, 1379, 1379, 1269, 2021, 451, 451, 25~
## $ oreb_rank         <int> 92, 4, 2360, 1362, 1881, 649, 1623, 1362, 23, 2360, ~
## $ dreb_rank         <int> 366, 482, 2035, 1905, 1101, 366, 43, 1905, 20, 2150,~
## $ reb_rank          <int> 83, 23, 2375, 1966, 1591, 340, 211, 1966, 3, 2400, 1~
## $ ast_rank          <int> 232, 410, 1394, 1208, 696, 84, 1558, 175, 410, 1752,~
## $ tov_rank          <int> 1809, 1809, 444, 18, 35, 636, 1390, 1135, 636, 96, 9~
## $ stl_rank          <int> 870, 1855, 1203, 1855, 1855, 1203, 615, 2301, 1203, ~
## $ blk_rank          <int> 20, 371, 917, 2069, 1329, 917, 1329, 917, 1731, 1329~
## $ blka_rank         <int> 1133, 1859, 146, 731, 146, 2409, 393, 1, 731, 146, 2~
## $ pf_rank           <int> 1822, 1976, 376, 737, 737, 2208, 23, 376, 40, 80, 40~
## $ pfd_rank          <int> 171, 115, 640, 2311, 2311, 486, 2382, 1725, 2422, 15~
```

```
## $ pts_rank          <int> 53, 83, 2354, 447, 2235, 26, 2235, 733, 1031, 959, 1~
## $ plus_minus_rank   <int> 1566, 1231, 2354, 674, 1929, 831, 2211, 2080, 226, 1~
```

##This only returns teams boxscores over the course of the season. That makes it a large table to sift through. You can use this in the most broad version shown by the second use which gives every boxscore for every team in one season. The first example allows you to filter certain parameters to find specific data. Note: you dont need to use all parameters, season is the only parameter you must imput.

#get_matchups

```
matchup <- get_matchups(game_id)
```

```
## http://stats.nba.com/stats/boxscorematchups?GameID=0021800001
## Rows: 196
## Columns: 38
## $ game_id                 <int> 21800001, 21800001, 21800001, 21800001, 218000~
## $ off_team_id             <int> 1610612755, 1610612755, 1610612755, 1610612755~
## $ off_team_abbreviation   <chr> "PHI", "PHI", "PHI", "PHI", "PHI", "PHI", "PHI~
## $ off_team_city           <chr> "Philadelphia", "Philadelphia", "Philadelphia"~
## $ off_team_nickname       <chr> "76ers", "76ers", "76ers", "76ers", "76ers", "~
## $ off_player_id           <int> 203967, 203967, 203967, 203967, 203967, 203967~
## $ off_player_name         <chr> "Dario Saric", "Dario Saric", "Dario Saric", "~
## $ def_team_id             <int> 1610612738, 1610612738, 1610612738, 1610612738~
## $ def_team_abbreviation   <chr> "BOS", "BOS", "BOS", "BOS", "BOS", "BOS", "BOS~
## $ def_team_city           <chr> "Boston", "Boston", "Boston", "Boston", "Bosto~
## $ def_team_nickname       <chr> "Celtics", "Celtics", "Celtics", "Celtics", "C~
## $ def_player_id           <int> 203382, 1627759, 202330, 201143, 202681, 20269~
## $ def_player_name         <chr> "Aron Baynes", "Jaylen Brown", "Gordon Hayward~
## $ matchup_min             <chr> "0:07", "1:09", "0:44", "0:19", "0:13", "0:59"~
## $ partial_poss            <int> 0, 5, 3, 1, 1, 4, 0, 1, 11, 0, 1, 7, 2, 1, 17,~
## $ pct_defender_total_time <dbl> 0.017, 0.127, 0.088, 0.034, 0.024, 0.157, 0.00~
## $ pct_off_total_time      <dbl> 0.016, 0.167, 0.107, 0.047, 0.031, 0.144, 0.00~
## $ pct_total_time_both_on  <dbl> 0.040, 0.302, 0.154, 0.068, 0.053, 0.453, 0.02~
## $ switches_on             <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ player_pts              <int> 0, 2, 0, 2, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0~
## $ team_pts                <int> 2, 10, 1, 4, 0, 11, 0, 0, 6, 2, 0, 8, 2, 2, 24~
## $ matchup_ast             <int> 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ matchup_potential_ast   <int> 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ matchup_tov             <int> 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0~
## $ matchup_blk             <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ matchup_fgm             <int> 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0~
## $ matchup_fga             <int> 0, 3, 1, 1, 0, 1, 0, 0, 2, 0, 0, 1, 2, 1, 2, 0~
## $ matchup_fg_pct          <dbl> 0.000, 0.333, 0.000, 1.000, 0.000, 1.000, 0.00~
## $ matchup_fg3m            <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0~
## $ matchup_fg3a            <int> 0, 2, 0, 0, 0, 0, 0, 0, 2, 0, 0, 1, 1, 1, 1, 0~
## $ matchup_fg3_pct         <dbl> 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0~
## $ help_blk                <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ help_fgm                <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ help_fga                <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ help_fg_perc            <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ matchup_ftm             <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ matchup_fta             <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ sfl                     <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
```

Gives in detail data on the matchups that take place in one game. Could be really helpful for indepth analysis on singular game data or could be pulled together for larger analysis

#get_playbyplay2

```
pbp2 <- map(gamelist, ~get_playbyplay2(.)) %>% compact() %>% bind_rows()
```

```
## https://data.nba.com/data/10s/v2015/json/mobile_teams/nba/2018/scores/pbp/0021800001_full_pbp.json
## Rows: 507
## Columns: 25
## $ season_id           <dbl> 2018, 2018, 2018, 2018, 2018, 2018, 2018, 2018, 20~
## $ game_id             <dbl> 21800001, 21800001, 21800001, 21800001, 21800001, ~
## $ game_date_id        <dbl> 20181016, 20181016, 20181016, 20181016, 20181016, ~
## $ visit_team          <chr> "PHI", "PHI", "PHI", "PHI", "PHI", "PHI", "PHI", "~
## $ home_team           <chr> "BOS", "BOS", "BOS", "BOS", "BOS", "BOS", "BOS", "~
## $ period              <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,~
## $ event_id            <int> 2, 4, 7, 8, 9, 10, 11, 12, 14, 16, 17, 19, 20, 21,~
## $ clock               <chr> "12:00", "12:00", "11:40", "11:40", "11:18", "11:1~
## $ description         <chr> "Start Period", "Jump Ball Embiid vs Horford (Simm~
## $ loc_x               <int> 0, 0, -53, -53, 0, -148, -148, 95, 4, 4, -8, -46, ~
## $ loc_y               <int> -80, -80, 265, 265, -80, 208, 208, 257, 19, 19, 3,~
## $ opt1                <int> 0, 0, 3, 0, 0, 3, 0, 1, 2, 0, 2, 2, 0, 3, 0, 2, 0,~
## $ opt2                <int> 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0,~
## $ message_type        <int> 0, 0, 1, 0, 3, 1, 0, 1, 41, 0, 41, 6, 0, 79, 0, 73~
## $ event_type          <int> 12, 10, 2, 4, 20, 2, 4, 5, 2, 4, 1, 2, 4, 2, 4, 2,~
## $ player_id           <int> 0, 203954, 203496, 0, 0, 1628369, 203967, 1627732,~
## $ opponent_player_id  <int> NA, 201143, NA, NA, NA, NA, NA, 202330, NA, NA, NA~
## $ secondary_player_id <int> NA, 1627732, NA, NA, NA, NA, NA, NA, NA, NA, 16277~
## $ team_id             <int> 0, 1610612755, 1610612755, 1610612738, 0, 16106127~
## $ offensive_team_id   <int> 0, 1610612755, 1610612755, 1610612755, 1610612738,~
## $ home_score          <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ visit_score         <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 2, 2, 2, 2, 2, 2,~
## $ order_no            <int> 2000, 6000, 7000, 8000, 9000, 10000, 11000, 12000,~
## $ mins                <dbl> 12, 12, 11, 11, 11, 11, 11, 11, 11, 10, 10, 10, 10~
## $ secs                <dbl> 0, 0, 40, 40, 18, 15, 13, 8, 3, 58, 55, 36, 35, 32~
## https://data.nba.com/data/10s/v2015/json/mobile_teams/nba/2018/scores/pbp/0021800002_full_pbp.json
## Rows: 538
## Columns: 25
## $ season_id           <dbl> 2018, 2018, 2018, 2018, 2018, 2018, 2018, 2018, 20~
## $ game_id             <dbl> 21800002, 21800002, 21800002, 21800002, 21800002, ~
## $ game_date_id        <dbl> 20181016, 20181016, 20181016, 20181016, 20181016, ~
## $ visit_team          <chr> "OKC", "OKC", "OKC", "OKC", "OKC", "OKC", "OKC", "~
## $ home_team           <chr> "GSW", "GSW", "GSW", "GSW", "GSW", "GSW", "GSW", "~
## $ period              <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,~
## $ event_id            <int> 2, 4, 7, 9, 10, 12, 13, 14, 15, 16, 17, 18, 19, 21~
## $ clock               <chr> "12:00", "11:57", "11:48", "11:43", "11:31", "11:1~
## $ description         <chr> "Start Period", "Jump Ball Jones vs Adams (Green g~
## $ loc_x               <int> 0, 0, -9, 94, 226, 1, 1, -140, -140, -7, 67, 67, -~
## $ loc_y               <int> -80, -80, 193, 319, 90, 120, 120, 218, 218, 17, 12~
## $ opt1                <int> 0, 0, 1, 0, 3, 2, 1, 3, 0, 2, 2, 1, 2, 0, 0, 2, 0,~
## $ opt2                <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ message_type        <int> 0, 0, 1, 45, 80, 78, 0, 1, 0, 6, 80, 0, 5, 0, 40, ~
## $ event_type          <int> 12, 10, 5, 5, 1, 2, 4, 2, 4, 1, 2, 4, 2, 4, 5, 1, ~
## $ player_id           <int> 0, 1627745, 202691, 202331, 201939, 203500, 203471~
```

```
## $ opponent_player_id  <int> NA, 203500, 203500, NA, NA, NA, NA, NA, NA, NA, NA~
## $ secondary_player_id <int> NA, 203110, NA, NA, 201142, NA, NA, NA, NA, NA, NA~
## $ team_id             <int> 0, 1610612744, 1610612744, 1610612760, 1610612744,~
## $ offensive_team_id   <int> 0, 1610612744, 1610612744, 1610612760, 1610612744,~
## $ home_score          <int> 0, 0, 0, 0, 3, 3, 3, 3, 3, 5, 5, 5, 5, 5, 5, 5, 5,~
## $ visit_score         <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 2,~
## $ order_no            <int> 2000, 6000, 7000, 9000, 10000, 12000, 13000, 14000~
## $ mins                <dbl> 12, 11, 11, 11, 11, 11, 11, 11, 11, 10, 10, 10, 10~
## $ secs                <dbl> 0, 57, 48, 43, 31, 13, 11, 7, 5, 57, 38, 36, 23, 2~
## https://data.nba.com/data/10s/v2015/json/mobile_teams/nba/2018/scores/pbp/0021800003_full_pbp.json
## Rows: 495
## Columns: 25
## $ season_id           <dbl> 2018, 2018, 2018, 2018, 2018, 2018, 2018, 2018, 20~
## $ game_id             <dbl> 21800003, 21800003, 21800003, 21800003, 21800003, ~
## $ game_date_id        <dbl> 20181017, 20181017, 20181017, 20181017, 20181017, ~
## $ visit_team          <chr> "MIL", "MIL", "MIL", "MIL", "MIL", "MIL", "MIL", "~
## $ home_team           <chr> "CHA", "CHA", "CHA", "CHA", "CHA", "CHA", "CHA", "~
## $ period              <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,~
## $ event_id            <int> 2, 4, 7, 9, 11, 12, 13, 14, 15, 16, 18, 19, 20, 22~
## $ clock               <chr> "12:00", "11:56", "11:47", "11:47", "11:47", "11:3~
## $ description         <chr> "Start Period", "Jump Ball Lopez vs Zeller (Anteto~
## $ loc_x               <int> 0, 0, -5, -5, 0, 8, 8, 20, 55, 228, 81, 81, 67, 11~
## $ loc_y               <int> -80, -80, 88, 88, -80, 8, 8, 116, 117, 42, 238, 23~
## $ opt1                <int> 0, 0, 2, 0, 1, 2, 0, 2, 2, 3, 3, 0, 0, 0, 3, 3, 3,~
## $ opt2                <int> 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ message_type        <int> 0, 0, 58, 2, 10, 6, 0, 101, 1, 1, 79, 0, 1, 45, 1,~
## $ event_type          <int> 12, 10, 1, 6, 3, 2, 4, 1, 1, 1, 2, 4, 6, 5, 1, 1, ~
## $ player_id           <int> 0, 201572, 201572, 203469, 201572, 201587, 203114,~
## $ opponent_player_id  <int> NA, 203469, NA, 201572, NA, NA, NA, NA, NA, NA, NA~
## $ secondary_player_id <int> NA, 203507, 203507, NA, NA, NA, NA, NA, NA, 203507~
## $ team_id             <int> 0, 1610612749, 1610612749, 1610612766, 1610612749,~
## $ offensive_team_id   <int> 0, 1610612749, 1610612749, 1610612749, 1610612749,~
## $ home_score          <int> 0, 0, 0, 0, 0, 0, 0, 0, 2, 2, 2, 2, 2, 2, 5, 5, 5,~
## $ visit_score         <int> 0, 0, 2, 2, 3, 3, 3, 5, 5, 8, 8, 8, 8, 8, 8, 11, 1~
## $ order_no            <int> 2000, 6000, 7000, 9000, 11000, 12000, 13000, 14000~
## $ mins                <dbl> 12, 11, 11, 11, 11, 11, 11, 11, 11, 11, 10, 10, 10~
## $ secs                <dbl> 0, 56, 47, 47, 47, 35, 33, 31, 18, 2, 51, 50, 41, ~
```

Gives you an extreamly detailed boxscore that has every action over the course of the game including where the ball was shot from, who was defending and more. Note: I didnt notice any difference between play_by_play and play_by_play2 they both take the same parameters

#get_lineups

```
lineups <- get_lineups(season, 5, 'Base')
```

```
## http://stats.nba.com/stats/leaguedashlineups?Conference=&DateFrom=&DateTo=&Division=&GameSegment=&Gr
## Rows: 2,000
## Columns: 35
## $ player_id_1       <int> 201566, 201571, 202329, 201572, 101107, 201152, 2007~
## $ player_id_2       <int> 202331, 202696, 203081, 202339, 201587, 201954, 2015~
## $ player_id_3       <int> 203500, 203095, 203090, 203114, 202689, 202711, 2019~
## $ player_id_4       <int> 203924, 203932, 203468, 203507, 203087, 203506, 2026~
## $ player_id_5       <int> 1628390, 1628371, 203994, 1627763, 203469, 1626167, ~
```

```
## $ group_name        <chr> "R. Westbrook - P. George - S. Adams - J. Grant - T.~
## $ team_id           <int> 1610612760, 1610612753, 1610612757, 1610612749, 1610~
## $ team_abbreviation <chr> "OKC", "ORL", "POR", "MIL", "CHA", "IND", "TOR", "BO~
## $ gp                <int> 58, 65, 42, 56, 48, 32, 33, 42, 56, 40, 59, 40, 58, ~
## $ w                 <int> 35, 33, 24, 43, 22, 23, 24, 23, 34, 24, 39, 19, 33, ~
## $ l                 <int> 23, 32, 18, 13, 26, 9, 9, 19, 22, 16, 20, 21, 25, 9,~
## $ w_pct             <dbl> 0.603, 0.508, 0.571, 0.768, 0.458, 0.719, 0.727, 0.5~
## $ min               <int> 919, 851, 744, 597, 592, 554, 546, 504, 501, 500, 47~
## $ fgm               <int> 867, 747, 694, 567, 523, 481, 499, 489, 429, 457, 42~
## $ fga               <int> 1811, 1586, 1433, 1152, 1128, 1001, 1025, 970, 912, ~
## $ fg_pct            <dbl> 0.479, 0.471, 0.484, 0.492, 0.464, 0.481, 0.487, 0.5~
## $ fg3m              <int> 227, 183, 157, 155, 164, 116, 131, 139, 102, 145, 13~
## $ fg3a              <int> 567, 526, 438, 450, 445, 289, 374, 362, 295, 363, 35~
## $ fg3_pct           <dbl> 0.400, 0.348, 0.358, 0.344, 0.369, 0.401, 0.350, 0.3~
## $ ftm               <int> 288, 234, 294, 219, 216, 145, 190, 106, 150, 114, 22~
## $ fta               <int> 407, 296, 349, 282, 248, 195, 226, 127, 218, 178, 30~
## $ ft_pct            <dbl> 0.708, 0.791, 0.842, 0.777, 0.871, 0.744, 0.841, 0.8~
## $ oreb              <int> 235, 191, 188, 100, 125, 95, 120, 88, 115, 104, 84, ~
## $ dreb              <int> 709, 647, 555, 505, 417, 379, 408, 368, 381, 348, 34~
## $ reb               <int> 944, 838, 743, 605, 542, 474, 528, 456, 496, 452, 42~
## $ ast               <int> 505, 481, 395, 337, 320, 278, 304, 294, 275, 302, 26~
## $ tov               <int> 271, 248, 187, 170, 137, 161, 139, 124, 162, 151, 13~
## $ stl               <int> 179, 128, 125, 106, 93, 109, 90, 86, 93, 100, 85, 89~
## $ blk               <int> 78, 94, 92, 81, 59, 76, 62, 51, 51, 49, 57, 72, 46, ~
## $ blka              <int> 96, 71, 72, 76, 67, 64, 40, 33, 59, 56, 43, 33, 38, ~
## $ pf                <int> 413, 245, 323, 204, 213, 192, 198, 198, 187, 204, 22~
## $ pfd               <int> 388, 317, 298, 249, 244, 208, 235, 166, 204, 198, 25~
## $ pts               <int> 2249, 1911, 1839, 1508, 1426, 1223, 1319, 1223, 1110~
## $ plus_minus        <int> 160, 48, 139, 92, 93, 53, 105, 59, 58, -2, 121, 78, ~
## $ n                 <int> 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5~
```

```
bulls_lineups <- get_lineups(season, 5, 'Base', team_id = 1610612741)
```

```
## http://stats.nba.com/stats/leaguedashlineups?Conference=&DateFrom=&DateTo=&Division=&GameSegment=&Gr
## Rows: 582
## Columns: 35
## $ player_id_1       <int> 203200, 201577, 203200, 203200, 201577, 203897, 2032~
## $ player_id_2       <int> 203897, 203490, 1627739, 203897, 203897, 1627739, 20~
## $ player_id_3       <int> 203953, 203897, 1627853, 203953, 1627739, 1628374, 1~
## $ player_id_4       <int> 1627853, 1627739, 1628374, 1626166, 1627782, 1628976~
## $ player_id_5       <int> 1628976, 1628374, 1628976, 1628976, 1628374, 1628990~
## $ group_name        <chr> "J. Holiday - Z. LaVine - J. Parker - R. Arcidiacono~
## $ team_id           <int> 1610612741, 1610612741, 1610612741, 1610612741, 1610~
## $ team_abbreviation <chr> "CHI", "CHI", "CHI", "CHI", "CHI", "CHI", "CHI", "CH~
## $ gp                <int> 17, 12, 8, 13, 13, 6, 7, 6, 6, 9, 8, 7, 5, 4, 12, 11~
## $ w                 <int> 3, 6, 4, 3, 4, 0, 1, 1, 1, 5, 3, 2, 1, 1, 3, 3, 2, 4~
## $ l                 <int> 14, 6, 4, 10, 9, 6, 6, 5, 5, 4, 5, 5, 4, 3, 9, 8, 2,~
## $ w_pct             <dbl> 0.176, 0.500, 0.500, 0.231, 0.308, 0.000, 0.143, 0.1~
## $ min               <int> 228, 204, 101, 99, 90, 84, 80, 71, 56, 52, 51, 48, 4~
## $ fgm               <int> 186, 185, 80, 72, 72, 63, 54, 52, 44, 52, 57, 36, 36~
## $ fga               <int> 420, 386, 181, 168, 170, 149, 133, 122, 114, 83, 97,~
## $ fg_pct            <dbl> 0.443, 0.479, 0.442, 0.429, 0.424, 0.423, 0.406, 0.4~
## $ fg3m              <int> 45, 43, 21, 21, 13, 11, 18, 11, 13, 11, 11, 9, 9, 8,~
## $ fg3a              <int> 151, 112, 56, 53, 46, 35, 59, 32, 38, 21, 25, 34, 21~
```

```
## $ fg3_pct           <dbl> 0.298, 0.384, 0.375, 0.396, 0.283, 0.314, 0.305, 0.3~
## $ ftm               <int> 72, 71, 21, 35, 33, 21, 25, 25, 17, 28, 19, 17, 14, ~
## $ fta               <int> 94, 87, 29, 42, 42, 22, 30, 26, 24, 32, 19, 20, 17, ~
## $ ft_pct            <dbl> 0.766, 0.816, 0.724, 0.833, 0.786, 0.955, 0.833, 0.9~
## $ oreb              <int> 35, 44, 15, 18, 18, 12, 18, 9, 20, 8, 9, 7, 7, 4, 6,~
## $ dreb              <int> 174, 144, 68, 62, 85, 59, 48, 54, 34, 45, 43, 35, 43~
## $ reb               <int> 209, 188, 83, 80, 103, 71, 66, 63, 54, 53, 52, 42, 5~
## $ ast               <int> 117, 98, 50, 44, 30, 36, 31, 25, 25, 34, 30, 25, 21,~
## $ tov               <int> 73, 56, 23, 33, 23, 33, 32, 23, 16, 21, 15, 14, 12, ~
## $ stl               <int> 44, 27, 20, 17, 4, 15, 22, 9, 10, 5, 8, 12, 9, 2, 7,~
## $ blk               <int> 25, 20, 14, 11, 13, 6, 1, 6, 6, 7, 5, 5, 5, 8, 4, 3,~
## $ blka              <int> 31, 26, 7, 16, 4, 13, 8, 11, 5, 1, 2, 5, 3, 8, 3, 7,~
## $ pf                <int> 97, 72, 47, 43, 36, 22, 39, 38, 24, 15, 26, 16, 19, ~
## $ pfd               <int> 81, 78, 34, 39, 36, 25, 30, 20, 22, 21, 19, 20, 13, ~
## $ pts               <int> 489, 484, 202, 200, 190, 158, 151, 140, 118, 143, 14~
## $ plus_minus        <int> -22, -23, -7, -44, -1, -42, -30, -12, -24, 57, 22, -~
## $ n                 <int> 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5~
```

Gives you the player ids and names of the lineups of certain players across teams. Allows you to look at different variables across certain lineups like W/L as well as other statistics

#get_on_off

```
onoff_bulls <- get_on_off(season, 1610612741)
```

```
## http://stats.nba.com/stats/teamplayeronoffsummary?DateFrom=&DateTo=&GameSegment=&LastNGames=0&League
## Rows: 22
## Columns: 16
## $ team_id           <int> 1610612741, 1610612741, 1610612741, 1610612741, 1610~
## $ team_abbreviation <chr> "CHI", "CHI", "CHI", "CHI", "CHI", "CHI", "CHI", "CH~
## $ team_name         <chr> "Chicago Bulls", "Chicago Bulls", "Chicago Bulls", "~
## $ player_id         <int> 1628959, 1627853, 1628469, 1628976, 1627739, 1626245~
## $ player_name       <chr> "Alkins, Rawle", "Arcidiacono, Ryan", "Blakeney, Ant~
## $ gp                <int> 10, 81, 57, 44, 46, 60, 73, 38, 44, 63, 6, 74, 29, 5~
## $ min_on            <int> 120, 1961, 829, 1110, 1389, 746, 1430, 1325, 895, 21~
## $ plus_minus_on     <int> -5, -6, -8, -12, -7, -8, -10, -11, -6, -6, -6, -4, -~
## $ off_rating_on     <int> 106, 104, 102, 97, 106, 103, 100, 98, 101, 105, 103,~
## $ def_rating_on     <int> 111, 110, 110, 109, 113, 111, 110, 109, 107, 112, 11~
## $ net_rating_on     <int> -5, -5, -7, -11, -7, -8, -10, -11, -5, -6, -7, -4, -~
## $ min_off           <int> 3861, 2020, 3152, 2871, 2592, 3235, 2455, 514, 3086,~
## $ plus_minus_off    <int> -8, -10, -8, -6, -8, -8, -6, -4, -8, -10, -21, -11, ~
## $ off_rating_off    <int> 104, 102, 104, 106, 102, 104, 105, 101, 104, 101, 10~
## $ def_rating_off    <int> 112, 112, 112, 112, 111, 112, 112, 104, 113, 111, 12~
## $ net_rating_off    <int> -8, -10, -8, -6, -8, -8, -7, -3, -9, -9, -20, -10, -~
```

Gives a list of the players of a team for a given season. The table contains statistics for every players time while on and off the court such as plus/minus and offensive and defensive ratings. It also lists how many games they played in and total minutes played that season. Last number is team ID

#get_players & get_playerbio

```
players <- get_players(season)
```

```
## https://stats.nba.com/stats/commonallplayers?LeagueID=00&Season=2018-19&IsOnlyCurrentSeason=1%20
## Rows: 148
## Columns: 14
## $ player_id                 <dbl> 203518, 203112, 1629061, 1629152, 1628959, 1~
## $ display_last_comma_first  <chr> "Abrines, Alex", "Acy, Quincy", "Adel, Deng"~
## $ display_first_last        <chr> "Alex Abrines", "Quincy Acy", "Deng Adel", "~
## $ rosterstatus              <dbl> 0, 0, 1, 0, 1, 1, 1, 0, 0, 0, 1, 0, 1, 1, 1,~
## $ from_year                 <dbl> 2016, 2012, 2018, 2018, 2018, 2017, 2008, 20~
## $ to_year                   <dbl> 2018, 2018, 2018, 2018, 2018, 2019, 2019, 20~
## $ playercode                <chr> "alex_abrines", "quincy_acy", "deng_adel", "~
## $ team_id                   <dbl> 0, 0, 1610612739, 0, 1610612741, 1610612752,~
## $ team_city                 <chr> "", "", "Cleveland", "", "Chicago", "New Yor~
## $ team_name                 <chr> "", "", "Cavaliers", "", "Bulls", "Knicks", ~
## $ team_abbreviation         <chr> "", "", "CLE", "", "CHI", "NYK", "MIA", "", ~
## $ team_code                 <chr> "", "", "cavaliers", "", "bulls", "knicks", ~
## $ games_played_flag         <chr> "Y", "Y", "Y", "Y", "Y", "Y", "Y", "Y", "Y",~
## $ otherleague_experience_ch <dbl> 0, 11, 11, 11, 11, 11, 0, 11, 1, 11, 0, 0, 1~
```

```
player_bio <- get_playerbio(season)
```

```
## https://stats.nba.com/stats/leaguedashplayerbiostats?College=&Conference=&Country=&DateFrom=&DateTo=&
## Rows: 530
## Columns: 23
## $ player_id            <int> 203932, 1628988, 1627846, 201143, 202329, 1626210~
## $ player_name          <chr> "Aaron Gordon", "Aaron Holiday", "Abdel Nader", "~
## $ team_id              <int> 1610612753, 1610612754, 1610612760, 1610612738, 1~
## $ team_abbreviation    <chr> "ORL", "IND", "OKC", "BOS", "POR", "BKN", "SAC", ~
## $ age                  <int> 23, 22, 25, 33, 28, 26, 27, 25, 25, 26, 25, 26, 2~
## $ player_height        <chr> "6-9", "6-1", "6-6", "6-10", "6-9", "6-8", "6-6",~
## $ player_height_inches <int> 81, 73, 78, 82, 81, 80, 78, 78, 77, 85, 81, 80, 8~
## $ player_weight        <int> 220, 185, 225, 245, 220, 265, 214, 200, 186, 250,~
## $ college              <chr> "Arizona", "UCLA", "Iowa State", "Florida", "Wake~
## $ country              <chr> "USA", "USA", "Egypt", "Dominican Republic", "USA~
## $ draft_year           <chr> "2014", "2018", "2016", "2007", "2010", "Undrafte~
## $ draft_round          <chr> "1", "1", "2", "1", "1", "Undrafted", "1", "2", "~
## $ draft_number         <chr> "4", "23", "58", "3", "8", "Undrafted", "12", "32~
## $ gp                   <int> 78, 50, 61, 68, 81, 5, 64, 31, 25, 77, 21, 72, 14~
## $ pts                  <int> 16, 5, 4, 13, 9, 3, 8, 5, 9, 11, 5, 4, 0, 9, 10, ~
## $ reb                  <int> 7, 1, 1, 6, 7, 3, 3, 1, 2, 5, 3, 3, 1, 3, 3, 1, 2~
## $ ast                  <int> 3, 1, 0, 4, 1, 0, 2, 0, 3, 1, 0, 0, 0, 1, 1, 0, 1~
## $ net_rating           <dbl> 1.5, 7.0, -9.5, 6.1, 8.2, 27.1, -11.1, 1.6, 3.2, ~
## $ oreb_pct             <dbl> 0.047, 0.008, 0.017, 0.062, 0.048, 0.114, 0.021, ~
## $ dreb_pct             <dbl> 0.165, 0.088, 0.139, 0.161, 0.204, 0.405, 0.151, ~
## $ usg_pct              <dbl> 0.213, 0.206, 0.148, 0.188, 0.134, 0.222, 0.185, ~
## $ ts_pct               <dbl> 0.538, 0.518, 0.522, 0.605, 0.568, 0.610, 0.523, ~
## $ ast_pct              <dbl> 0.166, 0.180, 0.044, 0.203, 0.057, 0.176, 0.142, ~
```

Get players gives you a list of players that played that season. There is a category for rostered and some don't have that so I cant tell why they would be on the list if they were never rostered but have the game played flag varable as yes. These people are not assigned a team either. Player bio gives you physical attribues of players that year as well as their stats from the season you imputted

#get_shotchart

```
shors_abrines <- get_shotchart(203518, season)
```

```
## https://stats.nba.com/stats/shotchartdetail?CFID=33&CFPARAMS=2018-19&ContextFilter=&ContextMeasure=F
## Rows: 157
## Columns: 24
## $ grid_type           <chr> "Shot Chart Detail", "Shot Chart Detail", "Shot Ch~
## $ game_id             <int> 21800002, 21800002, 21800002, 21800002, 21800002, ~
## $ game_event_id       <int> 177, 254, 366, 372, 383, 535, 553, 597, 122, 141, ~
## $ player_id           <int> 203518, 203518, 203518, 203518, 203518, 203518, 20~
## $ player_name         <chr> "Alex Abrines", "Alex Abrines", "Alex Abrines", "A~
## $ team_id             <int> 1610612760, 1610612760, 1610612760, 1610612760, 16~
## $ team_name           <chr> "Oklahoma City Thunder", "Oklahoma City Thunder", ~
## $ period              <int> 1, 2, 2, 2, 2, 3, 3, 4, 1, 1, 2, 2, 2, 2, 3, 3, 4,~
## $ minutes_remaining   <int> 0, 7, 1, 0, 0, 1, 0, 9, 2, 1, 11, 9, 7, 0, 3, 0, 8~
## $ seconds_remaining   <int> 41, 38, 21, 52, 4, 21, 24, 49, 39, 48, 14, 10, 19,~
## $ event_type          <chr> "Made Shot", "Missed Shot", "Made Shot", "Missed S~
## $ action_type         <chr> "Driving Layup Shot", "Jump Shot", "Pullup Jump sh~
## $ shot_type           <chr> "2PT Field Goal", "3PT Field Goal", "3PT Field Goa~
## $ shot_zone_basic     <chr> "Restricted Area", "Above the Break 3", "Above the~
## $ shot_zone_area      <chr> "Center(C)", "Left Side Center(LC)", "Left Side Ce~
## $ shot_zone_range     <chr> "Less Than 8 ft.", "24+ ft.", "24+ ft.", "24+ ft."~
## $ shot_distance       <int> 2, 25, 26, 26, 14, 25, 26, 22, 2, 24, 27, 24, 21, ~
## $ loc_x               <int> 11, -200, -112, 92, 90, -226, 154, -226, -19, 143,~
## $ loc_y               <int> 21, 164, 237, 245, 111, 110, 217, 32, 23, 204, 235~
## $ shot_attempted_flag <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,~
## $ shot_made_flag      <int> 1, 0, 1, 0, 0, 0, 1, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0,~
## $ game_date           <int> 20181016, 20181016, 20181016, 20181016, 20181016, ~
## $ htm                 <chr> "GSW", "GSW", "GSW", "GSW", "GSW", "GSW", "GSW", "~
## $ vtm                 <chr> "OKC", "OKC", "OKC", "OKC", "OKC", "OKC", "OKC", "~
```

Has very discriptive data on every shot for every player over the course of a season. In order to make sense of this data it would be useful to manipulate it or possibly put it into a chart. Doesnt produce a graph or chart instead a table. The first number is the player ID

#get_schedule

```
schedule <- get_schedule(season)
```

```
## https://data.nba.com/data/10s/v2015/json/mobile_teams/nba/2018/league/00_full_schedule.json
## Rows: 1,393
## Columns: 12
## $ game_id        <int> 11800001, 11800002, 11800003, 11800004, 11800005, 11800~
## $ gcode          <chr> "20180928/MELPHI", "20180928/BOSCHA", "20180929/PORTOR"~
## $ series         <chr> "", "", "", "", "", "", "", "", "", "", "", "", "", "",~
## $ game_date      <chr> "2018-09-28", "2018-09-28", "2018-09-29", "2018-09-29",~
## $ arena          <chr> "Wells Fargo Center", "Dean E. Smith Center", "Rogers A~
## $ game_date_utc  <chr> "2018-09-28", "2018-09-28", "2018-09-29", "2018-09-30",~
## $ game_time_utc  <chr> "23:00", "23:30", "23:00", "00:30", "00:30", "01:00", "~
## $ visitor        <chr> "MEL", "BOS", "POR", "BJD", "MIN", "PER", "MIA", "CHA",~
## $ home           <chr> "PHI", "CHA", "TOR", "DAL", "GSW", "UTA", "SAS", "BOS",~
## $ season_type    <chr> "PRE", "PRE", "PRE", "PRE", "PRE", "PRE", "PRE", "PRE",~
## $ season_type_id <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1~
## $ season         <int> 2018, 2018, 2018, 2018, 2018, 2018, 2018, 2018, 2018, 2~
```

Gives you the complete NBA schedule for the selected season

#get_tracking

```
tracking <- get_tracking(season, 'Team',measure_type = 'Defense')
```

```
## http://stats.nba.com/stats/leaguedashptstats?College=&Conference=&Country=&DateFrom=&DateTo=&Division
## Rows: 30
## Columns: 13
## $ team_id           <int> 1610612737, 1610612738, 1610612751, 1610612766, 1610~
## $ team_abbreviation <chr> "ATL", "BOS", "BKN", "CHA", "CHI", "CLE", "DAL", "DE~
## $ team_name         <chr> "Atlanta Hawks", "Boston Celtics", "Brooklyn Nets", ~
## $ gp                <int> 82, 82, 82, 82, 81, 82, 82, 82, 82, 82, 82, 82, 82, ~
## $ w                 <int> 29, 49, 42, 39, 22, 19, 33, 54, 41, 57, 53, 48, 48, ~
## $ l                 <int> 53, 33, 40, 43, 59, 63, 49, 28, 41, 25, 29, 34, 34, ~
## $ min               <int> 242, 241, 243, 241, 242, 240, 241, 240, 242, 241, 24~
## $ stl               <int> 8, 8, 6, 7, 7, 6, 6, 7, 6, 7, 8, 8, 6, 7, 8, 7, 7, 8~
## $ blk               <int> 5, 5, 4, 4, 4, 2, 4, 4, 4, 6, 4, 4, 4, 5, 5, 5, 5, 5~
## $ dreb              <int> 34, 34, 35, 33, 34, 31, 35, 34, 33, 36, 31, 33, 35, ~
## $ def_rim_fgm       <int> 16, 15, 15, 17, 16, 17, 16, 16, 16, 15, 17, 15, 17, ~
## $ def_rim_fga       <int> 25, 24, 25, 26, 25, 25, 24, 25, 25, 24, 26, 25, 26, ~
## $ def_rim_fg_pct    <dbl> 0.654, 0.638, 0.620, 0.662, 0.657, 0.694, 0.659, 0.6~
```

You can track different measures by teams or player by season. You can change your measure type to track different things. Measre types include: “Drives”,“Defense”,“CatchShoot”,“Passing”,“Possessions”,“PullUpShot”,“Rebounding”,“

#get_all_tracking

```
all_tracking <- get_all_tracking(season, "Team", measure_types = 'Defense')
```

```
## http://stats.nba.com/stats/leaguedashptstats?College=&Conference=&Country=&DateFrom=&DateTo=&Division
## Rows: 30
## Columns: 13
## $ team_id           <int> 1610612737, 1610612738, 1610612751, 1610612766, 1610~
## $ team_abbreviation <chr> "ATL", "BOS", "BKN", "CHA", "CHI", "CLE", "DAL", "DE~
## $ team_name         <chr> "Atlanta Hawks", "Boston Celtics", "Brooklyn Nets", ~
## $ gp                <int> 82, 82, 82, 82, 81, 82, 82, 82, 82, 82, 82, 82, 82, ~
## $ w                 <int> 29, 49, 42, 39, 22, 19, 33, 54, 41, 57, 53, 48, 48, ~
## $ l                 <int> 53, 33, 40, 43, 59, 63, 49, 28, 41, 25, 29, 34, 34, ~
## $ min               <int> 242, 241, 243, 241, 242, 240, 241, 240, 242, 241, 24~
## $ stl               <int> 8, 8, 6, 7, 7, 6, 6, 7, 6, 7, 8, 8, 6, 7, 8, 7, 7, 8~
## $ blk               <int> 5, 5, 4, 4, 4, 2, 4, 4, 4, 6, 4, 4, 4, 5, 5, 5, 5, 5~
## $ dreb              <int> 34, 34, 35, 33, 34, 31, 35, 34, 33, 36, 31, 33, 35, ~
## $ def_rim_fgm       <int> 16, 15, 15, 17, 16, 17, 16, 16, 16, 15, 17, 15, 17, ~
## $ def_rim_fga       <int> 25, 24, 25, 26, 25, 25, 24, 25, 25, 24, 26, 25, 26, ~
## $ def_rim_fg_pct    <dbl> 0.654, 0.638, 0.620, 0.662, 0.657, 0.694, 0.659, 0.6~
```

This is similar to tracking, the difference here is you dont need to put a measure types in and it will return all of them where the one above needs that variable described. measure types include: “Drives”,“Defense”,“CatchShoot”,“Passing”,“Possessions”,“PullUpShot”,“Rebounding”,“Efficiency”,“SpeedDistance”,“Elbow

#get_playtype

```
playtype <- get_playtype(season, 'T', 'Postup')
```

```
## https://stats.nba.com/stats/synergyplaytypes?LeagueID=00&&PerMode=PerGame&PlayType=Postup&PlayerOrTea
## Rows: 30
## Columns: 22
## $ season_id         <int> 22018, 22018, 22018, 22018, 22018, 22018, 22018, 220~
## $ team_id           <int> 1610612759, 1610612755, 1610612750, 1610612743, 1610~
## $ team_abbreviation <chr> "SAS", "PHI", "MIN", "DEN", "DET", "NOP", "POR", "LA~
## $ team_name         <chr> "San Antonio Spurs", "Philadelphia 76ers", "Minnesot~
## $ play_type         <chr> "Postup", "Postup", "Postup", "Postup", "Postup", "P~
## $ type_grouping     <chr> "Offensive", "Offensive", "Offensive", "Offensive", ~
## $ percentile        <dbl> 0.897, 0.552, 0.586, 0.759, 0.276, 0.862, 0.621, 0.7~
## $ gp                <int> 82, 82, 82, 82, 82, 82, 82, 82, 81, 82, 82, 82, 82, ~
## $ poss_pct          <dbl> 0.122, 0.107, 0.100, 0.091, 0.089, 0.073, 0.072, 0.0~
## $ ppp               <dbl> 0.988, 0.938, 0.944, 0.980, 0.898, 0.987, 0.951, 0.9~
## $ fg_pct            <dbl> 0.485, 0.464, 0.502, 0.504, 0.474, 0.483, 0.506, 0.4~
## $ ft_poss_pct       <dbl> 0.144, 0.220, 0.141, 0.162, 0.178, 0.205, 0.126, 0.1~
## $ tov_poss_pct      <dbl> 0.080, 0.137, 0.152, 0.119, 0.143, 0.103, 0.123, 0.1~
## $ sf_poss_pct       <dbl> 0.121, 0.165, 0.114, 0.129, 0.147, 0.173, 0.100, 0.1~
## $ plusone_poss_pct  <dbl> 0.033, 0.038, 0.028, 0.032, 0.035, 0.031, 0.019, 0.0~
## $ score_poss_pct    <dbl> 0.498, 0.483, 0.477, 0.501, 0.465, 0.511, 0.487, 0.4~
## $ efg_pct           <dbl> 0.485, 0.464, 0.502, 0.504, 0.474, 0.483, 0.506, 0.4~
## $ poss              <int> 13, 12, 11, 10, 9, 8, 8, 7, 8, 8, 8, 7, 7, 7, 6, 7, ~
## $ pts               <int> 13, 11, 10, 9, 8, 8, 7, 7, 7, 7, 7, 6, 6, 6, 6, 6, 6~
## $ fgm               <int> 5, 3, 4, 3, 3, 3, 3, 3, 3, 3, 3, 2, 2, 2, 2, 2, 2, 2~
## $ fga               <int> 10, 8, 8, 7, 7, 6, 6, 6, 6, 6, 6, 5, 5, 5, 5, 5, 4, ~
## $ fgmx              <int> 5, 4, 4, 3, 3, 3, 3, 3, 3, 3, 3, 2, 2, 3, 2, 3, 2, 2~
```

Allows you to look at all 32 teams and how they perform givern a certain playtype. I think it would also be possible to organize by player insead of team to see how certain players do in certain play types. The team aspect of this would be more useful to us.

#get_genteral

```
general <- get_general(season, type = 'Team', 'Base')
```

```
## https://stats.nba.com/stats/leaguedashTeamstats?College=&Conference=&Country=&DateFrom=&DateTo=&Divis
## Rows: 30
## Columns: 56
## $ team_id         <int> 1610612737, 1610612738, 1610612751, 1610612766, 161061~
## $ team_name       <chr> "Atlanta Hawks", "Boston Celtics", "Brooklyn Nets", "C~
## $ gp              <int> 82, 82, 82, 82, 82, 82, 82, 82, 82, 82, 82, 82, 82, 82~
## $ w               <int> 29, 49, 42, 39, 22, 19, 33, 54, 41, 57, 53, 48, 48, 37~
## $ l               <int> 53, 33, 40, 43, 60, 63, 49, 28, 41, 25, 29, 34, 34, 45~
## $ w_pct           <dbl> 0.354, 0.598, 0.512, 0.476, 0.268, 0.232, 0.402, 0.659~
## $ min             <int> 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48~
## $ fgm             <int> 41, 42, 40, 40, 39, 38, 38, 41, 38, 44, 39, 41, 41, 42~
## $ fga             <int> 91, 90, 89, 89, 87, 87, 86, 90, 88, 89, 87, 87, 87, 90~
## $ fg_pct          <dbl> 0.451, 0.465, 0.449, 0.448, 0.453, 0.444, 0.447, 0.466~
## $ fg3m            <int> 13, 12, 12, 11, 9, 10, 12, 11, 12, 13, 16, 9, 10, 10, ~
## $ fg3a            <int> 37, 34, 36, 33, 25, 29, 36, 31, 34, 34, 45, 25, 25, 31~
## $ fg3_pct         <dbl> 0.352, 0.365, 0.353, 0.351, 0.351, 0.355, 0.340, 0.351~
## $ ftm             <int> 17, 15, 19, 18, 16, 16, 18, 15, 17, 16, 19, 15, 22, 16~
```

```
## $ fta              <int> 23, 19, 25, 23, 20, 20, 25, 20, 23, 20, 24, 21, 28, 23~
## $ ft_pct           <dbl> 0.752, 0.802, 0.745, 0.797, 0.783, 0.792, 0.742, 0.755~
## $ oreb             <int> 11, 9, 11, 9, 8, 10, 10, 11, 11, 9, 10, 9, 9, 10, 8, 1~
## $ dreb             <int> 34, 34, 35, 33, 34, 31, 35, 34, 33, 36, 31, 33, 35, 36~
## $ reb              <int> 46, 44, 46, 43, 42, 42, 45, 46, 45, 46, 42, 43, 45, 46~
## $ ast              <int> 25, 26, 23, 23, 21, 20, 23, 27, 22, 29, 21, 26, 24, 25~
## $ tov              <int> 17, 12, 15, 12, 14, 13, 14, 13, 13, 14, 13, 13, 14, 15~
## $ stl              <int> 8, 8, 6, 7, 7, 6, 6, 7, 6, 7, 8, 8, 6, 7, 8, 7, 7, 8, ~
## $ blk              <int> 5, 5, 4, 4, 4, 2, 4, 4, 4, 6, 4, 4, 4, 5, 5, 5, 5, 5, ~
## $ blka             <int> 5, 3, 5, 6, 5, 5, 4, 5, 5, 3, 4, 5, 6, 5, 4, 4, 4, 5, ~
## $ pf               <int> 23, 20, 21, 18, 20, 20, 20, 20, 22, 21, 22, 19, 23, 20~
## $ pfd              <int> 22, 19, 22, 20, 18, 19, 23, 20, 21, 19, 20, 20, 24, 20~
## $ pts              <int> 113, 112, 112, 110, 104, 104, 108, 110, 107, 117, 113,~
## $ plus_minus       <int> -6, 4, 0, -1, -8, -9, -1, 4, 0, 6, 4, 3, 0, -1, -2, 0,~
## $ gp_rank          <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
## $ w_rank           <int> 26, 9, 14, 17, 27, 28, 22, 4, 16, 3, 5, 11, 11, 20, 22~
## $ l_rank           <int> 26, 9, 14, 17, 27, 28, 22, 4, 16, 3, 5, 11, 11, 20, 22~
## $ w_pct_rank       <dbl> 26, 9, 14, 17, 27, 28, 22, 4, 16, 3, 5, 11, 11, 20, 22~
## $ min_rank         <int> 7, 18, 1, 11, 3, 24, 18, 27, 7, 15, 11, 30, 11, 18, 4,~
## $ fgm_rank         <int> 15, 11, 20, 21, 23, 26, 28, 12, 27, 1, 25, 16, 17, 6, ~
## $ fga_rank         <int> 4, 9, 14, 12, 22, 23, 28, 11, 19, 13, 26, 27, 24, 8, 3~
## $ fg_pct_rank      <dbl> 21, 14, 25, 26, 20, 28, 27, 13, 29, 1, 24, 4, 7, 9, 23~
## $ fg3m_rank        <int> 4, 6, 5, 11, 30, 20, 7, 18, 9, 3, 1, 29, 25, 20, 27, 1~
## $ fg3a_rank        <int> 3, 7, 5, 10, 27, 24, 4, 16, 6, 8, 1, 29, 28, 17, 25, 1~
## $ fg3_pct_rank     <dbl> 16, 7, 14, 18, 19, 13, 27, 17, 23, 3, 12, 5, 2, 29, 25~
## $ ftm_rank         <int> 18, 28, 6, 9, 25, 22, 7, 27, 19, 23, 3, 26, 1, 24, 15,~
## $ fta_rank         <int> 12, 29, 4, 16, 26, 27, 5, 25, 17, 28, 7, 23, 1, 14, 18~
## $ ft_pct_rank      <dbl> 21, 4, 24, 6, 11, 7, 25, 20, 23, 5, 9, 22, 8, 29, 15, ~
## $ oreb_rank        <int> 4, 20, 10, 19, 30, 12, 16, 2, 5, 21, 14, 25, 22, 15, 2~
## $ dreb_rank        <int> 18, 16, 10, 22, 21, 28, 14, 17, 24, 3, 29, 23, 8, 4, 2~
## $ reb_rank         <int> 12, 22, 7, 23, 25, 26, 16, 8, 18, 11, 28, 24, 13, 6, 2~
## $ ast_rank         <int> 10, 5, 21, 24, 27, 29, 22, 2, 26, 1, 28, 9, 18, 11, 19~
## $ tov_rank         <int> 30, 3, 26, 2, 19, 9, 20, 8, 11, 21, 6, 10, 22, 29, 15,~
## $ stl_rank         <int> 11, 4, 27, 21, 20, 28, 29, 13, 22, 15, 5, 3, 23, 16, 6~
## $ blk_rank         <int> 14, 10, 28, 18, 26, 30, 26, 24, 29, 1, 18, 20, 22, 8, ~
## $ blka_rank        <int> 24, 2, 22, 29, 28, 27, 6, 14, 17, 1, 8, 21, 30, 20, 13~
## $ pf_rank          <int> 29, 12, 23, 3, 9, 6, 8, 7, 26, 22, 25, 4, 28, 13, 24, ~
## $ pfd_rank         <int> 5, 26, 7, 18, 30, 28, 2, 20, 11, 27, 24, 23, 1, 15, 9,~
## $ pts_rank         <int> 12, 14, 15, 19, 27, 29, 21, 20, 25, 2, 11, 22, 5, 16, ~
## $ plus_minus_rank <int> 26, 6, 15, 18, 27, 30, 20, 8, 17, 2, 5, 10, 13, 23, 24~
## $ cfid             <int> 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10~
## $ cfparams         <chr> "Atlanta Hawks", "Boston Celtics", "Brooklyn Nets", "C~
```

Oragnized by teams. This has general team info and organizes them across multiple statistics and ranks them in these statistics out of the 32 teams rather than the numeric value of the statistic

#get_hustle

```
hustle <- get_hustle(21800001)
```

```
## http://stats.nba.com/stats/hustlestatsboxscore?GameID=0021800001
## Rows: 24
## Columns: 26
## $ game_id                  <int> 21800001, 21800001, 21800001, 21800001, 2180~
```

```
## $ team_id                   <int> 1610612755, 1610612755, 1610612755, 16106127~
## $ team_abbreviation         <chr> "PHI", "PHI", "PHI", "PHI", "PHI", "PHI", "P~
## $ team_city                 <chr> "Philadelphia", "Philadelphia", "Philadelphi~
## $ player_id                 <int> 203967, 203496, 203954, 1628365, 1627732, 10~
## $ player_name               <chr> "Dario Saric", "Robert Covington", "Joel Emb~
## $ start_position            <chr> "F", "F", "C", "G", "G", "", "", "", "", "",~
## $ comment                   <chr> "", "", "", "", "", "", "", "", "", "", "", ~
## $ pts                       <int> 6, 8, 23, 5, 19, 5, 16, 4, 1, 0, 0, 23, 10, ~
## $ contested_shots           <int> 7, 9, 20, 1, 7, 4, 3, 2, 2, 0, 0, 5, 5, 19, ~
## $ contested_shots_2pt       <int> 3, 4, 14, 0, 2, 4, 2, 1, 1, 0, 0, 0, 3, 15, ~
## $ contested_shots_3pt       <int> 4, 5, 6, 1, 5, 0, 1, 1, 1, 0, 0, 5, 2, 4, 3,~
## $ deflections               <int> 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ charges_drawn             <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ screen_assists            <int> 1, 0, 3, 2, 2, 1, 0, 0, 0, 0, 0, 2, 0, 2, 0,~
## $ screen_ast_pts            <int> 2, 0, 6, 5, 5, 2, 0, 0, 0, 0, 0, 5, 0, 4, 0,~
## $ off_loose_balls_recovered <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ def_loose_balls_recovered <int> 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,~
## $ loose_balls_recovered     <int> 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,~
## $ off_boxouts               <int> 2, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ def_boxouts               <int> 2, 0, 9, 2, 4, 2, 0, 0, 1, 0, 0, 1, 3, 5, 0,~
## $ box_out_player_team_rebs  <int> 0, 0, 4, 1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 3, 0,~
## $ box_out_player_rebs       <int> 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ box_outs                  <int> 4, 0, 9, 3, 4, 3, 0, 0, 1, 0, 0, 1, 3, 5, 0,~
## $ mins                      <int> 22, 34, 36, 24, 42, 11, 29, 22, 12, 1, 1, 28~
## $ secs                      <int> 54, 13, 49, 20, 44, 11, 46, 26, 43, 27, 27, ~
```

This might be the most interesting as it is the "hustle" stats that dont show up on a normal stat sheet. This gives you these stats per game. It includes things such as deflections, charges drawn, loose balls recovered, boxouts, etc. This is broken down by player within the game. The number in the function is a game ID

#get_shooting

```
shooting <- get_shooting(season, 'Team', "By+Zone", "Base")
```

```
## https://stats.nba.com/stats/leaguedashTeamshotlocations?College=&Conference=&Country=&DateFrom=&Date
## Rows: 30
## Columns: 25
## $ team_id                  <int> 1610612737, 1610612738, 1610612751, 161061276~
## $ team_name                <chr> "Atlanta Hawks", "Boston Celtics", "Brooklyn ~
## $ fgm_restricted_area      <int> 20, 16, 18, 17, 18, 16, 17, 18, 16, 16, 16, 1~
## $ fga_restricted_area      <int> 33, 26, 30, 29, 30, 27, 26, 29, 27, 24, 26, 3~
## $ fg_pct_restricted_area   <dbl> 0.611, 0.644, 0.612, 0.594, 0.600, 0.597, 0.6~
## $ fgm_in_the_paint         <int> 5, 5, 5, 5, 7, 6, 5, 7, 5, 4, 4, 5, 6, 4, 6, ~
## $ fga_in_the_paint         <int> 13, 13, 14, 14, 17, 15, 13, 17, 14, 10, 10, 1~
## $ fg_pct_in_the_paint      <dbl> 0.378, 0.427, 0.376, 0.409, 0.415, 0.385, 0.3~
## $ fgm_mid_range            <int> 2, 7, 3, 4, 5, 6, 4, 4, 4, 9, 2, 7, 5, 4, 4, ~
## $ fga_mid_range            <int> 7, 16, 8, 12, 14, 15, 10, 12, 10, 19, 4, 18, ~
## $ fg_pct_mid_range         <dbl> 0.363, 0.423, 0.388, 0.406, 0.371, 0.405, 0.3~
## $ fgm_left_corner_3        <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
## $ fga_left_corner_3        <int> 4, 3, 3, 3, 3, 3, 3, 3, 5, 3, 4, 3, 3, 3, 3, ~
## $ fg_pct_left_corner_3     <dbl> 0.373, 0.418, 0.385, 0.367, 0.389, 0.399, 0.3~
## $ fgm_right_corner_3       <int> 2, 1, 1, 1, 1, 0, 1, 1, 1, 1, 2, 1, 1, 1, 1, ~
## $ fga_right_corner_3       <int> 5, 3, 3, 3, 2, 2, 3, 2, 4, 3, 6, 2, 2, 3, 3, ~
## $ fg_pct_right_corner_3    <dbl> 0.406, 0.282, 0.359, 0.454, 0.418, 0.320, 0.3~
```

```
## $ fgm_above_the_break_3    <int> 9, 10, 10, 9, 6, 8, 9, 8, 8, 10, 11, 6, 7, 7,~
## $ fga_above_the_break_3    <int> 26, 27, 28, 26, 19, 22, 28, 24, 25, 27, 33, 1~
## $ fg_pct_above_the_break_3 <dbl> 0.340, 0.371, 0.349, 0.342, 0.339, 0.354, 0.3~
## $ fgm_backcourt            <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ fga_backcourt            <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ fg_pct_backcourt         <dbl> 0.000, 0.000, 0.071, 0.054, 0.000, 0.000, 0.0~
## $ date_from                <chr> "", "", "", "", "", "", "", "", "", "", "", "~
## $ date_to                  <chr> "", "", "", "", "", "", "", "", "", "", "", "~
```

Breaks down shooting by team. Each teams shooting is broken down by sections of the court such as restricted area, paint, midrange, and different areas of the 3-point land. IT contains shooting percentage from every area as well.

#get_shooting_dashboard

```
shooting_dashboard <- get_shooting_dashboard(season = 2018, type = 'Team', general_range = 'Overall')
```

```
## https://stats.nba.com/stats/leaguedashTeamptshot?CloseDefDistRange=&College=&Conference=&Country=&Da
## Rows: 30
## Columns: 26
## $ team_id              <int> 1610612760, 1610612758, 1610612740, 1610612737, 1~
## $ team_name            <chr> "Oklahoma City Thunder", "Sacramento Kings", "New~
## $ team_abbreviation    <chr> "OKC", "SAC", "NOP", "ATL", "MIN", "MIL", "BOS", ~
## $ gp                   <int> 81, 82, 82, 82, 82, 82, 82, 82, 82, 82, 82, 82, 8~
## $ g                    <int> 81, 82, 82, 82, 82, 82, 82, 82, 82, 82, 82, 82, 8~
## $ fga_frequency        <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1~
## $ fgm                  <int> 42, 43, 43, 41, 41, 43, 42, 42, 42, 41, 40, 40, 4~
## $ fga                  <int> 93, 93, 92, 91, 91, 91, 90, 90, 90, 89, 89, 89, 8~
## $ fg_pct               <dbl> 0.454, 0.464, 0.474, 0.451, 0.456, 0.476, 0.465, ~
## $ efg_pct              <dbl> 0.515, 0.525, 0.529, 0.522, 0.511, 0.550, 0.534, ~
## $ fg2a_frequency       <dbl> 0.652, 0.678, 0.676, 0.597, 0.685, 0.580, 0.619, ~
## $ fg2m                 <int> 31, 31, 33, 28, 31, 29, 29, 32, 31, 30, 28, 27, 2~
## $ fg2a                 <int> 61, 63, 62, 54, 62, 52, 56, 59, 59, 58, 55, 53, 5~
## $ fg2_pct              <dbl> 0.511, 0.505, 0.536, 0.518, 0.504, 0.565, 0.527, ~
## $ fg3a_frequency       <dbl> 0.348, 0.322, 0.324, 0.403, 0.315, 0.420, 0.381, ~
## $ fg3m                 <int> 11, 11, 10, 13, 10, 13, 12, 10, 11, 10, 11, 12, 1~
## $ fg3a                 <int> 32, 29, 29, 37, 28, 38, 34, 31, 30, 31, 33, 36, 3~
## $ fg3_pct              <dbl> 0.348, 0.378, 0.344, 0.352, 0.351, 0.352, 0.365, ~
## $ general_range        <chr> "Overall", "Overall", "Overall", "Overall", "Over~
## $ shot_clock_range     <chr> "", "", "", "", "", "", "", "", "", "", "", "", "~
## $ dribble_range        <chr> "", "", "", "", "", "", "", "", "", "", "", "", "~
## $ touch_time_range     <chr> "", "", "", "", "", "", "", "", "", "", "", "", "~
## $ close_def_dist_range <chr> "", "", "", "", "", "", "", "", "", "", "", "", "~
## $ shot_dist_range      <chr> "", "", "", "", "", "", "", "", "", "", "", "", "~
## $ date_from            <chr> "", "", "", "", "", "", "", "", "", "", "", "", "~
## $ date_to              <chr> "", "", "", "", "", "", "", "", "", "", "", "", "~
```

This is a further breakdown of the shooting function. It allows you to look deeper into things such as shotclock range, dribble range, shot distance range, and touch time range

#get_defense

```
defense <- get_defense(season = 2018, type = 'Team', defense_category = 'Overall', per_mode = 'Totals')
```

```
## https://stats.nba.com/stats/leaguedashptTeamdefend?College=&Conference=&Country=&DateFrom=&DateTo=&D
## Rows: 30
## Columns: 11
## $ team_id           <int> 1610612749, 1610612748, 1610612744, 1610612738, 1610~
## $ team_name         <chr> "Milwaukee Bucks", "Miami Heat", "Golden State Warri~
## $ team_abbreviation <chr> "MIL", "MIA", "GSW", "BOS", "LAL", "IND", "UTA", "TO~
## $ gp                <int> 82, 82, 82, 82, 82, 82, 81, 82, 82, 82, 82, 82, 81, ~
## $ g                 <int> 82, 82, 82, 82, 82, 82, 81, 82, 82, 82, 82, 82, 81, ~
## $ freq              <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1~
## $ d_fgm             <int> 3319, 3135, 3277, 3233, 3423, 3163, 3208, 3285, 3221~
## $ d_fga             <int> 7664, 7101, 7377, 7208, 7586, 7025, 7112, 7318, 7035~
## $ d_fg_pct          <dbl> 0.433, 0.441, 0.444, 0.449, 0.451, 0.450, 0.451, 0.4~
## $ normal_fg_pct     <dbl> 0.457, 0.455, 0.457, 0.459, 0.461, 0.459, 0.459, 0.4~
## $ pct_plusminus     <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
```

Is sorted by the 32 teams in this example but can aslo be sorted by players. Can look through different types of defense and can sort by season. There are 5 or more other optional variables you can imput into the funtion to further narrow your data

#get_news

```
news <- get_news(limit = 10)
```

```
## https://stats-prod.nba.com/wp-json/statscms/v1/type/spotlight/?limit=10
## [1] "lexical error: invalid char in json text.\n                                                    <!DOCTYPE HTM
```

Doesnt seem to work for whatever reason. It is supposed to return a data set with weekly news. Doesnt seem too important to what we are doing. limit is the number of recent news you want, defult is 100.

#get_player_movement

```
player_movement <- get_player_movement()
```

```
## https://stats.nba.com/js/data/playermovement/NBA_Player_Movement.json
## Rows: 4,441
## Columns: 9
## $ transaction_type        <chr> "Signing", "Signing", "Trade", "Trade", "Trade~
## $ transaction_date        <chr> "2021-03-22T00:00:00", "2021-03-22T00:00:00", ~
## $ transaction_description <chr> "Cleveland Cavaliers signed guard Quinn Cook t~
## $ team_id                 <int> 1610612739, 1610612751, 1610612746, 1610612758~
## $ team_slug               <chr> "cavaliers", "nets", "clippers", "kings", "kin~
## $ player_id               <int> 1626188, 1628993, 0, 1629662, 0, 1628391, 2015~
## $ player_slug             <chr> "quinn-cook", "alize-johnson", "", "mfiondu-ka~
## $ additional_sort         <int> 0, 0, 1610612758, 1610612746, 1610612746, 1610~
## $ groupsort               <chr> "Signing 1037930", "Signing 1037933", "Trade 2~
```

This function gets all the roster tansactions since 2015. Is similar to the data that we had last semester where we were looking at player transactions. This data set might be even more indepth than the one we worked with

#get_rotowire_status

```
rotowire_status <- get_rotowire_status()
```

```
## https://stats-prod.nba.com/wp-json/statscms/v1/rotowire/player/
## [1] "lexical error: invalid char in json text.\n                                                    <!DOCTYPE HTM
```

I get a similar error to that of the funciton to get news.