Analyzing Early Draft Strategies

Spencer Kerch

@sjkerch I'll start by loading the data in. This function helps me parse the large files and can be adjusted by the size, as more recent regular seasons have data split over multiple csv's.

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.1     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.2     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
getDataParts <- function(path,nParts){
  data <- tibble()
  for (i in seq(nParts)) {
    data <- bind_rows(data,read_csv(paste0(path,"part_",
                                           if_else(i-1<10,paste0(0,i-1),paste0(i-1)),
                                           ".csv"
                                           )
                                    )
                      )
  }
  return(data)
}

#data20 <- read_csv("data/2020/part_00.csv")
#
#
# finals_21 <- read_csv("data/2021/post_season/finals.csv")
# quarterfinals_21 <- read_csv("data/2021/post_season/quarterfinals.csv")
# semifinals_21 <- read_csv("data/2021/post_season/semifinals.csv")
#

reg_21 <-getDataParts(path="data/2021/regular_season/",6)
```

```
## Rows: 550000 Columns: 15
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (4): draft_id, tournament_entry_id, player_name, position_name
```

```
## dbl  (10): clock, tournament_round_number, bye_week, projection_adp, pick_or...
## dttm  (1): draft_time
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 550000 Columns: 15
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (4): draft_id, tournament_entry_id, player_name, position_name
## dbl  (10): clock, tournament_round_number, bye_week, projection_adp, pick_or...
## dttm  (1): draft_time
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 550000 Columns: 15
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (4): draft_id, tournament_entry_id, player_name, position_name
## dbl  (10): clock, tournament_round_number, bye_week, projection_adp, pick_or...
## dttm  (1): draft_time
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 550000 Columns: 15
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (4): draft_id, tournament_entry_id, player_name, position_name
## dbl  (10): clock, tournament_round_number, bye_week, projection_adp, pick_or...
## dttm  (1): draft_time
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 550000 Columns: 15
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (4): draft_id, tournament_entry_id, player_name, position_name
## dbl  (10): clock, tournament_round_number, bye_week, projection_adp, pick_or...
## dttm  (1): draft_time
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 46750 Columns: 15
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (4): draft_id, tournament_entry_id, player_name, position_name
## dbl  (10): clock, tournament_round_number, bye_week, projection_adp, pick_or...
## dttm  (1): draft_time
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
fast_22<-getDataParts("data/2022/regular_season/fast/",27)
```

```
## Rows: 330000 Columns: 17
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (6): draft_id, draft_entry_id, tournament_entry_id, tournament_round_d...
## dbl  (10): clock, tournament_round_number, bye_week, projection_adp, pick_or...
## dttm  (1): draft_time
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 330000 Columns: 17
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (6): draft_id, draft_entry_id, tournament_entry_id, tournament_round_d...
## dbl  (10): clock, tournament_round_number, bye_week, projection_adp, pick_or...
## dttm  (1): draft_time
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 16800 Columns: 17
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (6): draft_id, draft_entry_id, tournament_entry_id, tournament_round_d...
## dbl  (10): clock, tournament_round_number, bye_week, projection_adp, pick_or...
## dttm  (1): draft_time
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 330000 Columns: 17
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (6): draft_id, draft_entry_id, tournament_entry_id, tournament_round_d...
## dbl  (10): clock, tournament_round_number, bye_week, projection_adp, pick_or...
## dttm  (1): draft_time
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 330000 Columns: 17
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (6): draft_id, draft_entry_id, tournament_entry_id, tournament_round_d...
## dbl  (10): clock, tournament_round_number, bye_week, projection_adp, pick_or...
## dttm  (1): draft_time
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 16800 Columns: 17
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (6): draft_id, draft_entry_id, tournament_entry_id, tournament_round_d...
## dbl  (10): clock, tournament_round_number, bye_week, projection_adp, pick_or...
## dttm  (1): draft_time
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 330000 Columns: 17
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (6): draft_id, draft_entry_id, tournament_entry_id, tournament_round_d...
## dbl  (10): clock, tournament_round_number, bye_week, projection_adp, pick_or...
## dttm  (1): draft_time
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 330000 Columns: 17
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (6): draft_id, draft_entry_id, tournament_entry_id, tournament_round_d...
## dbl  (10): clock, tournament_round_number, bye_week, projection_adp, pick_or...
## dttm  (1): draft_time
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 16800 Columns: 17
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (6): draft_id, draft_entry_id, tournament_entry_id, tournament_round_d...
## dbl  (10): clock, tournament_round_number, bye_week, projection_adp, pick_or...
## dttm  (1): draft_time
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 330000 Columns: 17
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (6): draft_id, draft_entry_id, tournament_entry_id, tournament_round_d...
## dbl  (10): clock, tournament_round_number, bye_week, projection_adp, pick_or...
## dttm  (1): draft_time
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 330000 Columns: 17
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (6): draft_id, draft_entry_id, tournament_entry_id, tournament_round_d...
## dbl  (10): clock, tournament_round_number, bye_week, projection_adp, pick_or...
## dttm  (1): draft_time
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 16800 Columns: 17
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (6): draft_id, draft_entry_id, tournament_entry_id, tournament_round_d...
## dbl  (10): clock, tournament_round_number, bye_week, projection_adp, pick_or...
## dttm  (1): draft_time
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 330000 Columns: 17
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (6): draft_id, draft_entry_id, tournament_entry_id, tournament_round_d...
## dbl  (10): clock, tournament_round_number, bye_week, projection_adp, pick_or...
## dttm  (1): draft_time
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 330000 Columns: 17
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (6): draft_id, draft_entry_id, tournament_entry_id, tournament_round_d...
## dbl  (10): clock, tournament_round_number, bye_week, projection_adp, pick_or...
## dttm  (1): draft_time
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 16800 Columns: 17
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (6): draft_id, draft_entry_id, tournament_entry_id, tournament_round_d...
## dbl  (10): clock, tournament_round_number, bye_week, projection_adp, pick_or...
## dttm  (1): draft_time
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 330000 Columns: 17
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (6): draft_id, draft_entry_id, tournament_entry_id, tournament_round_d...
## dbl  (10): clock, tournament_round_number, bye_week, projection_adp, pick_or...
## dttm  (1): draft_time
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 330000 Columns: 17
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (6): draft_id, draft_entry_id, tournament_entry_id, tournament_round_d...
## dbl  (10): clock, tournament_round_number, bye_week, projection_adp, pick_or...
## dttm  (1): draft_time
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 16800 Columns: 17
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (6): draft_id, draft_entry_id, tournament_entry_id, tournament_round_d...
## dbl  (10): clock, tournament_round_number, bye_week, projection_adp, pick_or...
## dttm  (1): draft_time
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 330000 Columns: 17
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (6): draft_id, draft_entry_id, tournament_entry_id, tournament_round_d...
## dbl  (10): clock, tournament_round_number, bye_week, projection_adp, pick_or...
## dttm  (1): draft_time
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 330000 Columns: 17
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (6): draft_id, draft_entry_id, tournament_entry_id, tournament_round_d...
## dbl  (10): clock, tournament_round_number, bye_week, projection_adp, pick_or...
## dttm  (1): draft_time
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 16800 Columns: 17
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (6): draft_id, draft_entry_id, tournament_entry_id, tournament_round_d...
## dbl  (10): clock, tournament_round_number, bye_week, projection_adp, pick_or...
## dttm  (1): draft_time
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 330000 Columns: 17
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (6): draft_id, draft_entry_id, tournament_entry_id, tournament_round_d...
## dbl  (10): clock, tournament_round_number, bye_week, projection_adp, pick_or...
## dttm  (1): draft_time
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 330000 Columns: 17
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (6): draft_id, draft_entry_id, tournament_entry_id, tournament_round_d...
## dbl  (10): clock, tournament_round_number, bye_week, projection_adp, pick_or...
## dttm  (1): draft_time
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 16800 Columns: 17
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (6): draft_id, draft_entry_id, tournament_entry_id, tournament_round_d...
## dbl  (10): clock, tournament_round_number, bye_week, projection_adp, pick_or...
## dttm  (1): draft_time
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 330000 Columns: 17
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (6): draft_id, draft_entry_id, tournament_entry_id, tournament_round_d...
## dbl  (10): clock, tournament_round_number, bye_week, projection_adp, pick_or...
## dttm  (1): draft_time
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 330000 Columns: 17
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (6): draft_id, draft_entry_id, tournament_entry_id, tournament_round_d...
## dbl  (10): clock, tournament_round_number, bye_week, projection_adp, pick_or...
## dttm  (1): draft_time
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 16800 Columns: 17
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (6): draft_id, draft_entry_id, tournament_entry_id, tournament_round_d...
## dbl  (10): clock, tournament_round_number, bye_week, projection_adp, pick_or...
## dttm  (1): draft_time
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
mix_22 <- getDataParts("data/2022/regular_season/mixed/",9)
```

```
## Rows: 330000 Columns: 17
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (6): draft_id, draft_entry_id, tournament_entry_id, tournament_round_d...
## dbl  (10): clock, tournament_round_number, bye_week, projection_adp, pick_or...
## dttm  (1): draft_time
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 330000 Columns: 17
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (6): draft_id, draft_entry_id, tournament_entry_id, tournament_round_d...
## dbl  (10): clock, tournament_round_number, bye_week, projection_adp, pick_or...
## dttm  (1): draft_time
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 16800 Columns: 17
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (6): draft_id, draft_entry_id, tournament_entry_id, tournament_round_d...
## dbl  (10): clock, tournament_round_number, bye_week, projection_adp, pick_or...
## dttm  (1): draft_time
##
```

```
## i Use ‘spec()‘ to retrieve the full column specification for this data.
## i Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
## Rows: 330000 Columns: 17
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (6): draft_id, draft_entry_id, tournament_entry_id, tournament_round_d...
## dbl  (10): clock, tournament_round_number, bye_week, projection_adp, pick_or...
## dttm  (1): draft_time
##
## i Use ‘spec()‘ to retrieve the full column specification for this data.
## i Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
## Rows: 330000 Columns: 17
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (6): draft_id, draft_entry_id, tournament_entry_id, tournament_round_d...
## dbl  (10): clock, tournament_round_number, bye_week, projection_adp, pick_or...
## dttm  (1): draft_time
##
## i Use ‘spec()‘ to retrieve the full column specification for this data.
## i Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
## Rows: 16800 Columns: 17
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (6): draft_id, draft_entry_id, tournament_entry_id, tournament_round_d...
## dbl  (10): clock, tournament_round_number, bye_week, projection_adp, pick_or...
## dttm  (1): draft_time
##
## i Use ‘spec()‘ to retrieve the full column specification for this data.
## i Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
## Rows: 330000 Columns: 17
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (6): draft_id, draft_entry_id, tournament_entry_id, tournament_round_d...
## dbl  (10): clock, tournament_round_number, bye_week, projection_adp, pick_or...
## dttm  (1): draft_time
##
## i Use ‘spec()‘ to retrieve the full column specification for this data.
## i Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
## Rows: 330000 Columns: 17
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (6): draft_id, draft_entry_id, tournament_entry_id, tournament_round_d...
## dbl  (10): clock, tournament_round_number, bye_week, projection_adp, pick_or...
## dttm  (1): draft_time
##
## i Use ‘spec()‘ to retrieve the full column specification for this data.
## i Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
## Rows: 16800 Columns: 17
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (6): draft_id, draft_entry_id, tournament_entry_id, tournament_round_d...
## dbl  (10): clock, tournament_round_number, bye_week, projection_adp, pick_or...
## dttm  (1): draft_time
##
```

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

```r
# finals_22 <- read_csv("data/2022/post_season/finals/part_00.csv")
# semis_22 <- read_csv("data/2022/post_season/semifinals/part_00.csv")
quarters_22 <- getDataParts(path="data/2022/post_season/quarterfinals/",3)
```

```
## Rows: 150400 Columns: 17
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (6): draft_id, draft_entry_id, tournament_entry_id, tournament_round_dr...
## dbl (10): clock, tournament_round_number, bye_week, projection_adp, pick_ord...
## lgl  (1): draft_time
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 150400 Columns: 17
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (6): draft_id, draft_entry_id, tournament_entry_id, tournament_round_dr...
## dbl (10): clock, tournament_round_number, bye_week, projection_adp, pick_ord...
## lgl  (1): draft_time
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 150400 Columns: 17
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (6): draft_id, draft_entry_id, tournament_entry_id, tournament_round_dr...
## dbl (10): clock, tournament_round_number, bye_week, projection_adp, pick_ord...
## lgl  (1): draft_time
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Next I'll clean the 2022 data. First, the playoff_team metric in 2022 isn't representative of the team that made the playoffs, so I use the unique playoff id's from the first round to compile a list of playoff teams and change the playoff_team metric to represent teams that successfully made the playoffs that year. I double checked my work and added the fast and slow drafts together into one data frame and created a round drafted metric (which I now realize was unecessary, due to the team_pick_number metric)

```r
playoffTeamsBBMIII <- unique(quarters_22$tournament_entry_id)
fast_22 <- fast_22 %>% mutate(draftType = "fast",
                              playoff_team = if_else(tournament_entry_id%in%playoffTeamsBBMIII, 1,0))
mix_22 <- mix_22%>%mutate(draftType = "slow",
                              playoff_team = if_else(tournament_entry_id%in%playoffTeamsBBMIII, 1,0))

#drafts_22 <- bind_rows(fast_22,mix_22)

fast_22 %>%
  group_by(tournament_entry_id)%>%
  summarize(playoff_team=max(playoff_team))%>%
  ungroup()%>%
```

```
  group_by(playoff_team)%>%
  summarise(n())
```

```
## # A tibble: 2 x 2
##   playoff_team  `n()`
##          <dbl>  <int>
## 1            0 319628
## 2            1  18772
```

```
mix_22 %>%
  group_by(tournament_entry_id)%>%
  summarize(playoff_team=max(playoff_team))%>%
  ungroup()%>%
  group_by(playoff_team)%>%
  summarise(n())
```

```
## # A tibble: 2 x 2
##   playoff_team  `n()`
##          <dbl>  <int>
## 1            0 106502
## 2            1   6298
```

```
#18772+6298=25070 it works

draft_22 <- bind_rows(fast_22,mix_22) %>%
  mutate(pickRd = ceiling(overall_pick_number/12))
```

Now it's time to summarize the data by each specific entrie's draft, and the position they drafted in. I want to explore how different teams employed different strategies in their drafts, specifically when and how often they drafted different positions.

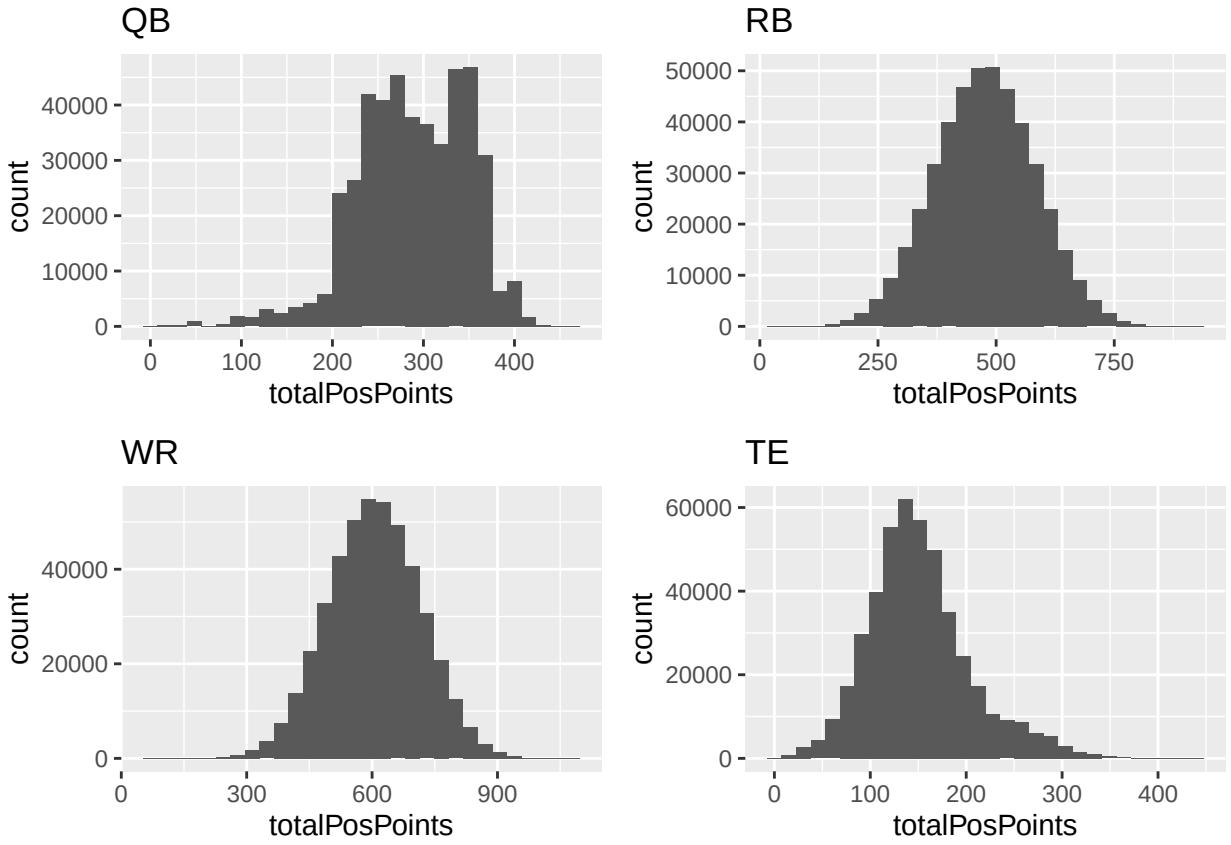
```
posDraftSummary<-draft_22 %>%
  group_by(tournament_entry_id,position_name) %>%
  summarize(avgPosPickNum = mean(overall_pick_number)  ,
            avgPosPickRd = mean(pickRd)  ,
            sumPickNum = sum(overall_pick_number),
            sumPickRd = sum(pickRd),
            totalDrafted = n(),
            meanPosPoints = mean(pick_points),
            totalPosPoints = sum(pick_points),
            totalTeamPoints = max(roster_points),
            pointsRatio = totalPosPoints/totalTeamPoints
            )
```

```
## `summarise()` has grouped output by 'tournament_entry_id'. You can override
## using the `.groups` argument.
```

```
qbPtHis <- posDraftSummary %>% filter(position_name == 'QB') %>% ggplot() +
  geom_histogram(aes(x=totalPosPoints)) + ggtitle("QB")
wrPtHis <- posDraftSummary %>% filter(position_name == 'RB') %>% ggplot() +
  geom_histogram(aes(x=totalPosPoints))+ ggtitle("RB")
```

```
rbPtHis <- posDraftSummary %>% filter(position_name == 'WR') %>% ggplot() +
  geom_histogram(aes(x=totalPosPoints)) + ggtitle("WR")
tePtHis <- posDraftSummary %>% filter(position_name == 'TE') %>% ggplot() +
  geom_histogram(aes(x=totalPosPoints)) + ggtitle("TE")

ggpubr::ggarrange(qbPtHis,wrPtHis,rbPtHis,tePtHis)
```

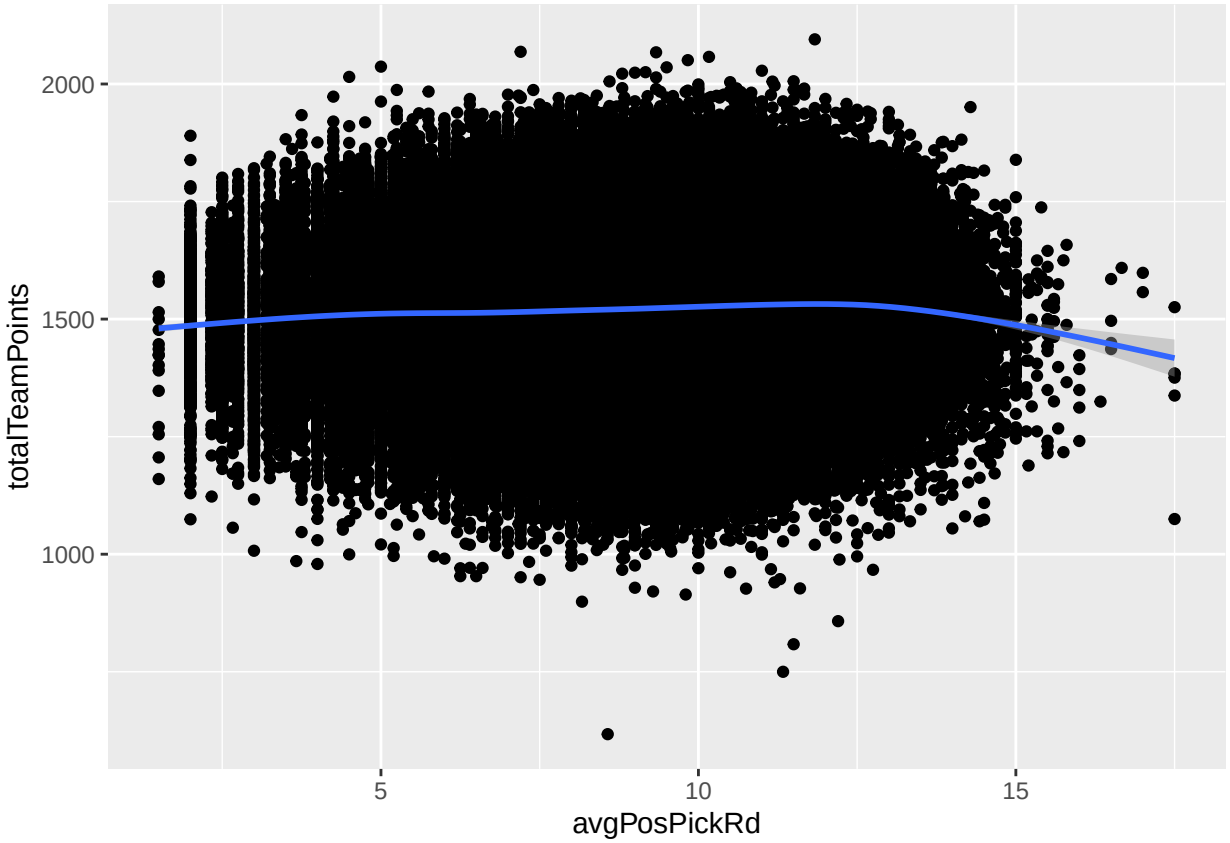
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



I'm interested in how when a team selects their running backs affects their output for the season. No RB is a popular draft strategy I've seen online where you minimize the RB's taken in early rounds.

```
posDraftSummary %>%
  filter(position_name == "RB")%>%
  ggplot(aes(y = totalTeamPoints, x = avgPosPickRd)) +
  geom_point() +
  geom_smooth()
```

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



It appears that overall there is very little correlation, however the running backs taken in later rounds could be causing a lot of avgPosPickRds to be raised even if a team is implementing a no RB strategy.

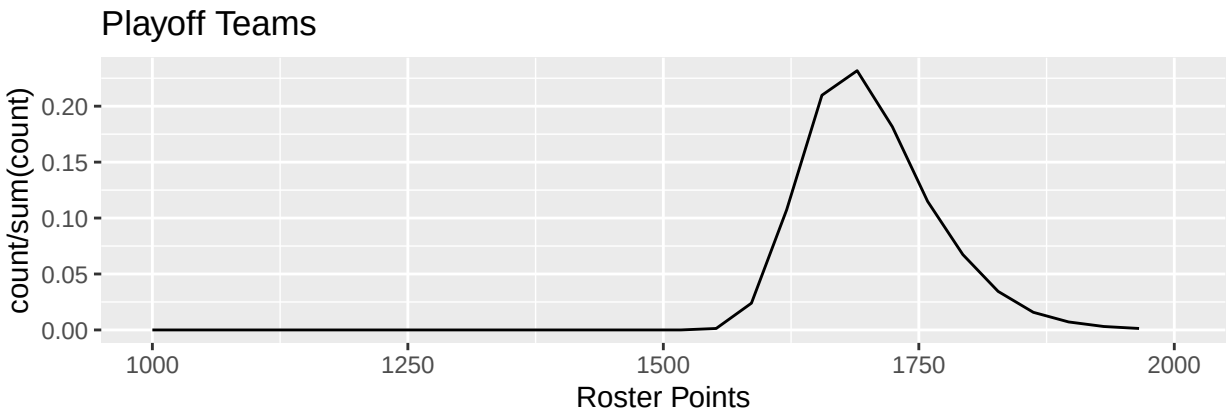
To delve further into early round strategy, I'll filter out rounds and only look at a teams first 5 picks, focusing on strategy in relation to running backs.

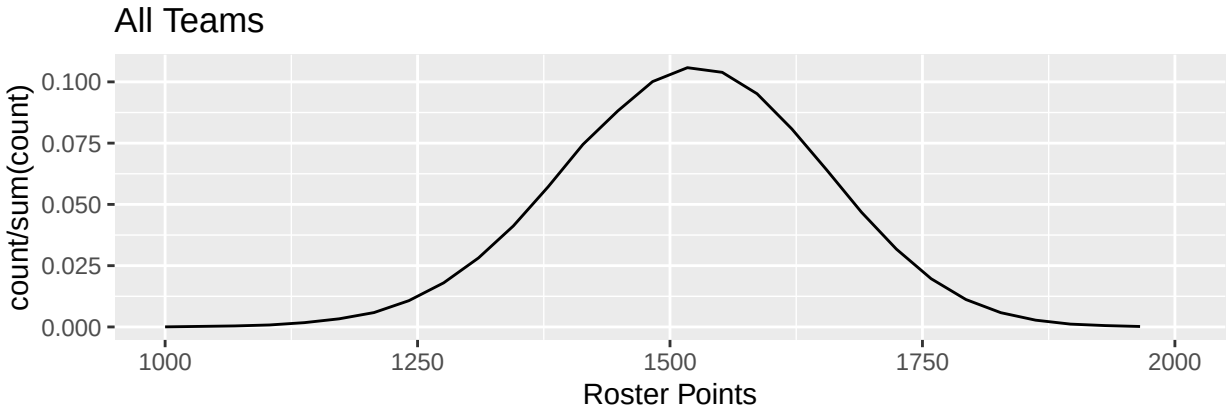
```
earlyDraftStrat<-draft_22%>%
  filter(team_pick_number <=5) %>%
  group_by(tournament_entry_id)%>%
  mutate(firstRbRd = if_else(position_name == "RB", team_pick_number, 100),
         firstRbPk = if_else(position_name == "RB", overall_pick_number, 10000),
         firstRbRd = min(firstRbRd),
         firstRbPk = min(firstRbPk)
         ) %>%
  ungroup()%>%
  pivot_wider(id_cols = c(tournament_entry_id,
                          roster_points, playoff_team,firstRbRd,firstRbPk ),
              names_from = team_pick_number,
              values_from = c(position_name, overall_pick_number)
  )%>%
  mutate(
    positonOrder = paste0(position_name_1,"_", position_name_2,"_",position_name_3,"_",position_name_4,
                          position_name_5),
    pickNumOrder = paste0(overall_pick_number_1,"_",overall_pick_number_2,"_",overall_pick_number_3,"_"
                          overall_pick_number_4,"_",overall_pick_number_5)
    )%>%
  select(-c(matches("[0-9]$")))
```

Now I'll set teams apart by different strategies. Obviously teams can potentially fall into multiple strategy bins, but I did my best to seperate out strategies I noticed among the way teams drafted positions in the first 5 rounds.

```
earlyDraftStrat <- earlyDraftStrat %>%
  mutate(
    noRB = if_else(firstRbRd == 100,1,0),
    rbCt = str_count(positonOrder,"RB"),
    qbCt = str_count(positonOrder,"QB"),
    wrCt = str_count(positonOrder,"WR"),
    goodTE = if_else(str_detect(positonOrder, "TE"),1,0),
    noTE = if_else(str_detect(positonOrder, "TE"),0,1),
    heavyRB = if_else(rbCt >2,1,0),
    twoQB = if_else(qbCt>1,1,0),
    noQB = if_else(str_detect(positonOrder,"QB"),0,1),
    regRG = if_else(rbCt == 1 | rbCt == 2,1,0),
    noWR = if_else(wrCt == 0,1,0),
    regWR = if_else(wrCt == 1| wrCt == 2 ,1,0),
    heavyWR = if_else(wrCt >2,1,0),
    noRB2QB = noRB*twoQB
  )

noRBHist<-earlyDraftStrat %>%
  filter(noRB==1)%>%
  ggplot()+
  geom_freqpoly(aes(x=roster_points,y=..count../sum(..count..)))+
  ggtitle("0 RB")+
  scale_x_continuous(limits=c(1000,2000))+
  labs(x="Roster Points")


playoffHist <- earlyDraftStrat %>%
  filter(playoff_team==1)%>%
  ggplot()+
  geom_freqpoly(aes(x=roster_points,y=..count../sum(..count..)))+
  ggtitle("Playoff Teams")+
  scale_x_continuous(limits=c(1000,2000))+
  labs(x="Roster Points")


allPoints <- earlyDraftStrat %>%
  ggplot()+
  geom_freqpoly(aes(x=roster_points,y=..count../sum(..count..)))+
  ggtitle("All Teams")+
  scale_x_continuous(limits=c(1000,2000))+
  labs(x="Roster Points")



ggpubr::ggarrange(playoffHist,allPoints,ncol=1)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

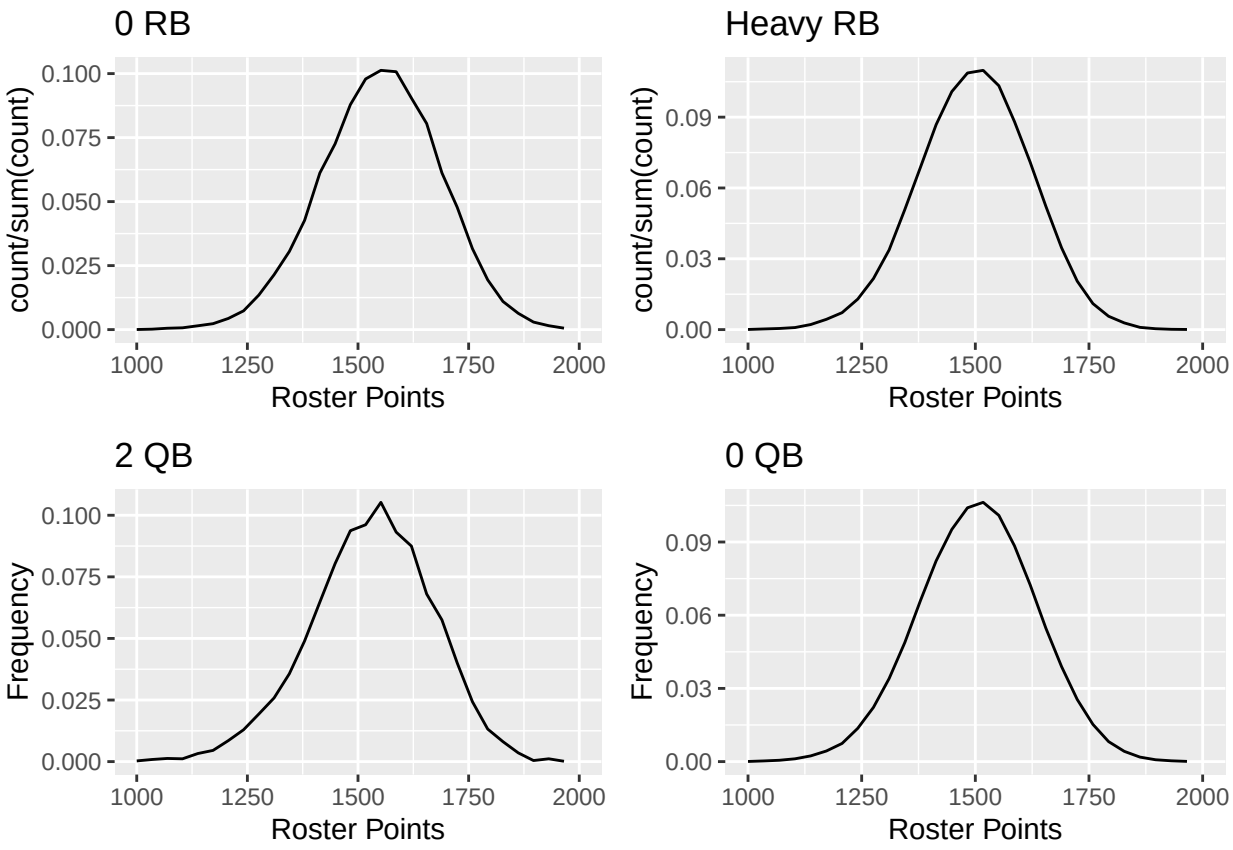





```
hRBHist <- earlyDraftStrat %>%
  filter(heavyRB==1)%>%
  ggplot()+
  geom_freqpoly(aes(x=roster_points,y=..count../sum(..count..)))+
  ggtitle("Heavy RB")+
  scale_x_continuous(limits=c(1000,2000))+
  labs(x="Roster Points")


twoQBHist <- earlyDraftStrat %>%
  filter(twoQB==1)%>%
  ggplot()+
  geom_freqpoly(aes(x=roster_points,y=..count../sum(..count..)))+
  ggtitle("2 QB")+
  scale_x_continuous(limits=c(1000,2000))+
  labs(x="Roster Points",
       y = "Frequency")


noQBHist <- earlyDraftStrat %>%
  filter(noQB==1)%>%
  ggplot()+
  geom_freqpoly(aes(x=roster_points,y=..count../sum(..count..)))+
  ggtitle("0 QB")+
  scale_x_continuous(limits=c(1000,2000))+
  labs(x="Roster Points",
```

```
       y = "Frequency")

ggpubr::ggarrange(noRBHist,hRBHist,twoQBHist,noQBHist)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



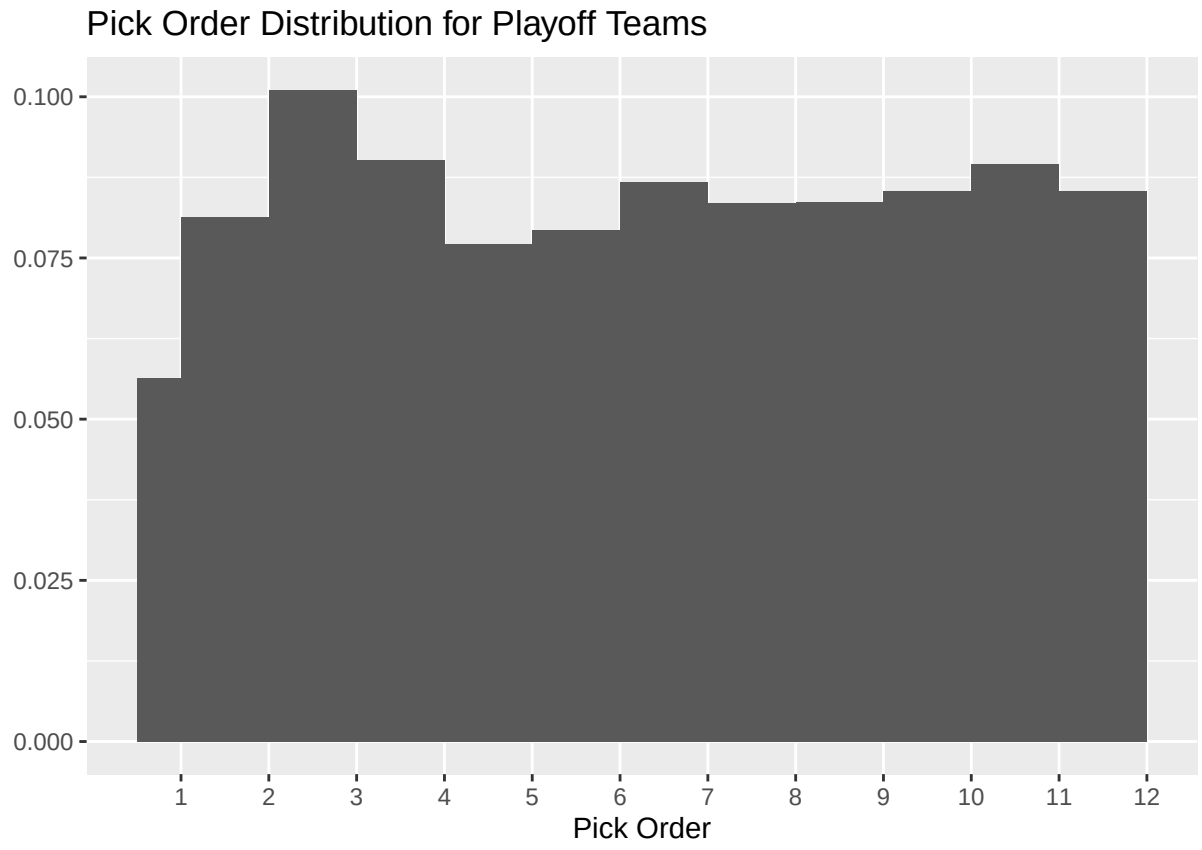
The roster points distribution for all teams and playoff teams shows pretty clearly that the minimum points a team needs on the season to advance past the regular is at least 1600, and the majority of teams that advance to the playoffs are scoring 1700+ over the course of a season, about 200 more than the mean for all teams.

The roster points distributions for different strategies are more similar to each other, a reminder that while strategy will give you a small edge, it is still victim to many other confounding variables, such as the drafter's knowledge and luck.

As a small detour from strategy, I am interested in looking at how a drafters position in the snake draft, something that is out of their control, affects a players odds.

```
pfPicks<-draft_22%>%
  filter(playoff_team == 1)%>%
  ggplot() +
  geom_histogram(aes(x=pick_order, y = ..count../sum(..count..)), binwidth = 1)+
    labs(title = 'Pick Order Distribution for Playoff Teams',
```

```
        x='Pick Order',
        y = '')+
  scale_x_binned(limits = c(.5,12))

pfPicks
```



The distribution is relatively uniform, especially from picks 4 thorugh 12, however a very interesting revelation from this plot is that the playoffs consisted of significantly less players who drafted 1st over all than those at other positions. This goes against conventional wisdom. It could be a quirk of just the 2022 playoffs, but it could also speak to the pressure of drafting first and how that changes strategy. Often a running back is the consensus top player in a draft, and as we will see, drafting a RB early could be less than ideal. Maybe drafting first overall pushes players to go with consensus picks, or maybe the 23 pick wait til they draft again makes them lose out on value at other positons that is not made up by the first pick.

I digress, while that is an interesting aside, draft pick order is not something that can be controlled for, I only include it as I found it too interesting to leave out, and I think it emphasizing the large amount of variation we are dealing with. I'm more interested in a blanked strategy that could be slightly tweaked, rather than a strategy that is dynamic based on draft order.

With that said, let's look at the success of these strategies we created earlier.

```
stategySuccsess<- earlyDraftStrat%>%
  summarise(

    noRBYoffRate = sum(noRB*playoff_team)/sum(noRB),
    noRBRate = sum(noRB)/n(),
    noRBMeanPts = sum(noRB*roster_points)/sum(noRB),
```

```
    goodTEYoffRate = sum(goodTE*playoff_team)/sum(goodTE),
    goodTERate = sum(goodTE)/n(),
    goodTEMeanPts = sum(goodTE*roster_points)/sum(goodTE),

    heavyRBYoffRate = sum(heavyRB*playoff_team)/sum(heavyRB),
    heavyRBRate = sum(heavyRB)/n(),
    heavyRBMeanPts = sum(heavyRB*roster_points)/sum(heavyRB),

    twoQBYoffRate = sum(twoQB*playoff_team)/sum(twoQB),
    twoQBRate = sum(twoQB)/n(),
    twoQBMeanPts = sum(twoQB*roster_points)/sum(twoQB),

    noQBRate = sum(noQB)/n(),
    noQBYoffRate = sum(noQB*playoff_team)/sum(noQB),
    noQBMeanPts = sum(noQB*roster_points)/sum(noQB),

    noTERate = sum(noTE)/n(),
    noTEYoffRate = sum(noTE*playoff_team)/sum(noTE),
    noTEMeanPts = sum(noTE*roster_points)/sum(noTE),

    noRB2QBRate = sum(noRB*twoQB)/n(),
    noRB2QBYoffRate = sum(noRB*twoQB*playoff_team)/sum(noRB*twoQB),
    noRB2QBMeanPts = sum(noRB*twoQB*roster_points)/sum(noRB*twoQB),



    regRBRate=sum(regRG)/n(),
    regRBYoffRate = sum(regRG*playoff_team)/sum(regRG),
    regRBMeanPts = sum(regRG*roster_points)/sum(regRG),

    noWRYoffRate = sum(noWR*playoff_team)/sum(noWR),
    noWRRate = sum(noWR)/n(),
    noWRMeanPts = sum(noWR*roster_points)/sum(noWR),

    regWRYoffRate = sum(regWR*playoff_team)/sum(regWR),
    regWRRate = sum(regWR)/n(),
    regWRMeanPts = sum(regWR*roster_points)/sum(regWR),

    heavyWRYoffRate = sum(heavyWR*playoff_team)/sum(heavyWR),
    heavyWRRate = sum(heavyWR)/n(),
    heavyWRMeanPts = sum(heavyWR*roster_points)/sum(heavyWR),

     baseYoffRate = sum(playoff_team)/n(),
    base_meanPts = mean(roster_points),
  )

rates<- stategySuccsess %>%
  select(-contains("Yoff"),-contains("MeanPts"))%>%
  pivot_longer(
    cols = everything()
  )%>%
  bind_rows(tibble(a=0,b=0))#need 1 extra to bind
```

```
yoffs <-stategySuccsess %>%
  select(contains("Yoff"))%>%
  pivot_longer(
    cols = everything()
  )

meanPts = stategySuccsess %>%
  select(contains("MeanPts"))%>%
  pivot_longer(
    cols = everything()
)#need 1 extra to bind

strategies_df <- bind_cols(rates,yoffs$value, meanPts$value)%>%
  select(-c(a,b))%>%
  mutate(name = if_else(is.na(name),'base',name))%>%
  rename('strategy' = name,
         "freq" = value,
         "yoff_rate" = ...5,
         "mean_roster_pts" = ...6)%>%
  mutate(
    strategy = str_remove(strategy,"Rate"),
    totalObs = nrow(earlyDraftStrat),
    totatStrategyObs = freq*totalObs
  )%>%
  arrange(desc(mean_roster_pts))
```

```
## New names:
## * `` -> `...5`
## * `` -> `...6`
```

```
strategies_df
```

```
## # A tibble: 12 x 6
##    strategy     freq yoff_rate mean_roster_pts totalObs totatStrategyObs
##    <chr>       <dbl>     <dbl>           <dbl>    <int>            <dbl>
##  1 noRB       0.0672    0.0816           1551.   451200            30329
##  2 noRB2QB    0.00292   0.0872           1541.   451200             1319
##  3 twoQB      0.0156    0.0674           1528.   451200             7046
##  4 heavyWR    0.463     0.0598           1525.   451200           208995
##  5 noTE       0.603     0.0561           1524.   451200           271969
##  6 regRB      0.769     0.0570           1523.   451200           347189
##  7 base      NA         0.0556           1521.   451200               NA
##  8 regWR      0.524     0.0523           1517.   451200           236295
##  9 goodTE     0.397     0.0548           1516.   451200           179231
## 10 noQB       0.593     0.0453           1504.   451200           267522
## 11 noWR       0.0131    0.0398           1498.   451200             5910
## 12 heavyRB    0.163     0.0382           1498.   451200            73682
```

As I suggested earlier, no RB seems to be the most dominant strategy. It is only employed by about 7% of entries, but leads to an average of 1550.6 points, over 50 points more than the average best ball entry, and leads to a playoff rate of 8.16%, second only to no RB and 2 QB with 30x the amount of data to support it.

The other strategy I would like to highlight is 2 QB. In typical fantasy this would be a disaster, you can only pick 1 QB to play a week, there is no point in having another high draft capital Quarterback on your roster

if he sits on the bench most weeks, many people avoid drafting a QB early at all in standard leagues, opting to beef up their other offensive starting and flex positions, this however is clearly not the move in best ball, as while over half of entries avoid getting a QB in the first 5 rounds, they average 16 points less than the mean, and make the playoffs 1% less often as well. 2 QB has started to look very appealing to me in the best ball setting. Quarterbacks have the greatest opportunity to boom and lead your team in scoring, especially an elite QB like Mahomes or Hurts. This doesn't happen every week however, and weather conditions or a few tipped passes can result in disaster as well. Having 2 elite QB's can keep you from having the QB positon hurt you, and maximize the opportunity you have to get an explosion of scoring at that position as best ball will set your optimal lineup every week after the games have been played. Additionally, it dilutes the pool of quarterback talent your 11 opponents can choose from. While very few (1.6%) entries opt for a 2 QB strategy, and even fewer(.3%) pair it with a no RB strategy, over 7000 and 1000 entries respectively make it a compelling strategy to at least consider. No RB still has the highest average points, but it makes the playoffs .5% less, which could be variation, but it also could also be explained that it makes opposing teams worse, or draft worse. While 2 QB might be something to consider but not neccesarily enact, I would highly suggest drafting at least 1 QB in the first 5 rounds and no running backs early.

Finally, I would like to see how predictive a linear model considers these variables, and see if it corroborates my earlier analysis.

```
library(tidymodels)
```

```
## -- Attaching packages --------------------------------------- tidymodels 1.0.0 --

## v broom        1.0.4     v rsample      1.0.0
## v dials        1.0.0     v tune         1.0.0
## v infer        1.0.2     v workflows    1.0.0
## v modeldata    1.0.0     v workflowsets 1.0.0
## v parsnip      1.0.0     v yardstick    1.0.0
## v recipes      1.0.1

## -- Conflicts ------------------------------------------ tidymodels_conflicts() --
## x scales::discard() masks purrr::discard()
## x dplyr::filter()   masks stats::filter()
## x recipes::fixed()  masks stringr::fixed()
## x dplyr::lag()      masks stats::lag()
## x yardstick::spec() masks readr::spec()
## x recipes::step()   masks stats::step()
## * Use tidymodels_prefer() to resolve common conflicts.
```

```
set.seed(19)
split_strategy <- initial_split(earlyDraftStrat,.75)
training = training(split_strategy)
testing=testing(split_strategy)

draftStratLM_1<-lm(roster_points~noRB+noTE+heavyRB+
    twoQB+noQB+noWR+heavyWR+noRB2QB,
   data = training)
summary(draftStratLM_1)
```

```
##
## Call:
## lm(formula = roster_points ~ noRB + noTE + heavyRB + twoQB +
##     noQB + noWR + heavyWR + noRB2QB, data = training)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -920.26  -85.00    1.87   86.83  521.95
##
## Coefficients:
##              Estimate Std. Error  t value             Pr(>|t|)
## (Intercept) 1538.3159     0.4630 3322.572 < 0.0000000000000002 ***
## noRB          14.9739     0.9865   15.178 < 0.0000000000000002 ***
## noTE           4.2814     0.5391    7.942  0.00000000000000199 ***
## heavyRB       -5.0368     0.7948   -6.337  0.00000000023416032 ***
## twoQB        -13.2385     2.0225   -6.546  0.00000000005928098 ***
## noQB         -47.2852     0.5467  -86.497 < 0.0000000000000002 ***
## noWR         -24.5814     2.0368  -12.068 < 0.0000000000000002 ***
## heavyWR       18.1146     0.6567   27.585 < 0.0000000000000002 ***
## noRB2QB       -8.8135     4.6055   -1.914               0.0557 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 127.1 on 338391 degrees of freedom
## Multiple R-squared:  0.0343, Adjusted R-squared:  0.03428
## F-statistic:  1502 on 8 and 338391 DF,  p-value: < 0.00000000000000022
```

```
lm_pred_1 <- predict(draftStratLM_1,testing)
```

Because we can assume there are very few entries that are only grouped into 1 strategy (for example, no RB likely implies heavy WR(3+) and can easily be paired with no TE or no QB) I will use the coefficients to see how the model grades different strategies. **Disclaimer: While the r^2 is very low, that is not too unexpected. So much more goes into a draft than just strategy, but for the purpose of analyzing the strategies it will do. Obviously if you implement a no RB strategy but draft poorly ranked wr's or overpay for a few postions you are forsaking any benefit the strategy gave you.**

While noRB2QB has a negative coefficient, after adding the noRB and 2 QB coefficients as well as heavy WR and no TE(I would likley employ the strategy drafting 2 QB and 3 WR in the first 5 rounds) the model predicts about 1543 points (I got this by adding the intercept and the coefficients). However I the model would still predict a strategy of no RB, heavy WR, no TE, and 1 QB as the best(I include 1 QB despite it not being a variable because we don't need analytics to understand 3+ qbs is a bad idea, and 2QB and no QB have -5 and -47 coefficients respectively). The model predicts 1575 points for this strategy.

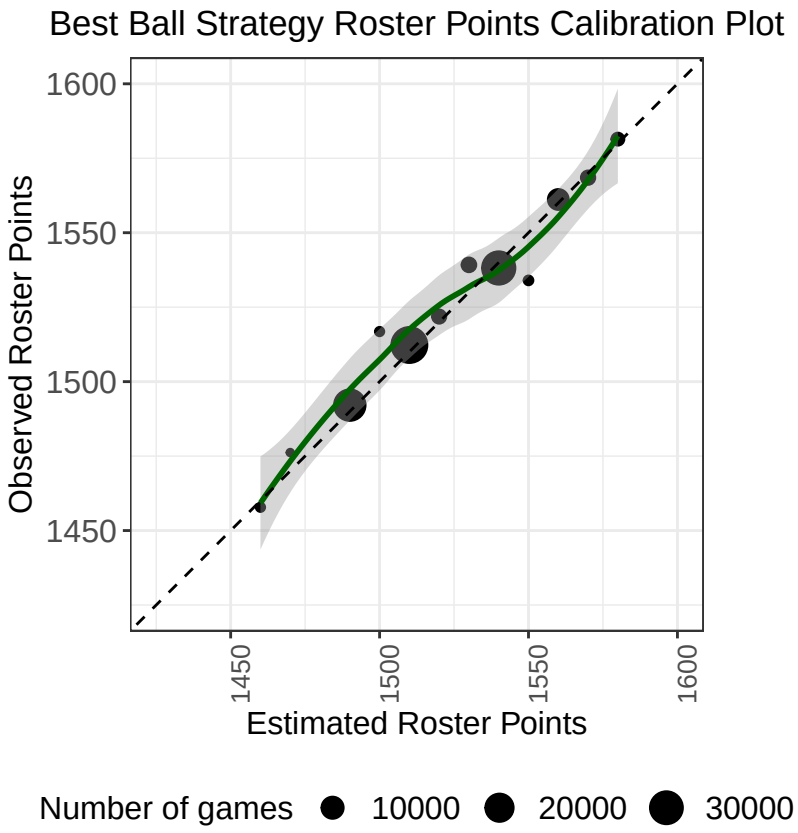
The below plot is solely to ensure the linear regression is working correctly.

```
plot_df <- bind_cols(testing$roster_points,lm_pred_1)%>%
  rename("actual"=...1 , "pred"=...2) %>%
  mutate(bin_pred_pts = round(pred/10)*10)%>%
  group_by(bin_pred_pts)%>%
  summarise(bin_act_pts = mean(actual),
            n_entries = n()
  )%>%
  ungroup()
```

```
## New names:
## * `` -> `...1`
## * `` -> `...2`
```

```r
cal_plot<-plot_df %>%
  ggplot() +
  geom_point(aes(x = bin_pred_pts, y = bin_act_pts, size = n_entries)) +
  geom_smooth(aes(x = bin_pred_pts, y = bin_act_pts), method = "loess", color = "darkgreen") +
  geom_abline(slope = 1, intercept = 0, color = "black", lty = 2) +
  coord_equal() +
  scale_x_continuous(limits = c(1425, 1600),) +
  scale_y_continuous(limits = c(1425, 1600)) +
  labs(
    size = "Number of games",
    x = "Estimated Roster Points",
    y = "Observed Roster Points",
    title = "Best Ball Strategy Roster Points Calibration Plot"
  ) +
  theme_bw() +
  theme(
    plot.title = element_text(hjust = 0.5),
    strip.background = element_blank(),
    strip.text = element_text(size = 12),
    axis.title = element_text(size = 12),
    axis.text.y = element_text(size = 12),
    axis.text.x = element_text(size = 10, angle = 90),
    legend.title = element_text(size = 12),
    legend.text = element_text(size = 12),
    legend.position = "bottom"
  )
cal_plot
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



The above plot bins all the predictions into groups by rounding to the nearest multiple of 25 and taking the averages. For example, the average actual roster points of entries predicted to have around 1550 points, is 1550 points.

Despite the low r^2, the model seems to be working mostly correctly, the only point that lies 25 points below the line is a bit concerning, but it does exist in the smallest group and the rest of the plot is on the line.

In conclusion, early draft strategy can be a very smart way to gain a small edge over the average best ball player when the right strategy is implemented. I would suggest taking many WR's and a QB early. It seems those are the positons that typically have the best value early on comparatively to the rest of their position. However luck still plays a large part, so make sure to pair research and an intelligent drafting together to give yourself a slight edge.