# Does optimizing for advance rate reduce season-long ceiling?

## Participants: Jake Boes

Twitter: @JakeBoesFF

With the introduction of larger regular season prizes to Best Ball Mania IV, the regular season scoring potential of a team is more important in 2023 than previous years of best ball. The goal of this project is to analyze and visualize the relationship between advance rate optimized roster constructions and season-long ceiling outcomes. If a player wants to shoot for the top regular season prize, do they need to give up their advance-friendly roster constructions?

The code in this project is entirely original, but some of the concepts were borrowed from other analysts. Specifically, the draft pick value curve and draft capital "bucketing" approach were both borrowed from Mike Leone of Establish the Run.

## Importing Data, Cleaning, Adding Variables

```
library("tidyverse")
library("scales")
library("knitr")
```

```
opts_chunk$set(tidy.opts=list(width.cutoff=60),tidy=TRUE)
```

```
load_bbm3_data <- function() {

    bbm3_data_part_1 <- read.csv("~/Bestball Data
↪   Bowl/BBM3/BBM_III_Regular_Season_Dump_Part_00_01302023.csv")
    bbm3_data_part_2 <- read.csv("~/Bestball Data
↪   Bowl/BBM3/BBM_III_Regular_Season_Dump_Part_01_01302023.csv")
    bbm3_data_part_3 <- read.csv("~/Bestball Data
↪   Bowl/BBM3/BBM_III_Regular_Season_Dump_Part_02_01302023.csv")
    bbm3_data_part_4 <- read.csv("~/Bestball Data
↪   Bowl/BBM3/BBM_III_Regular_Season_Dump_Part_03_01302023.csv")
    bbm3_data_part_5 <- read.csv("~/Bestball Data
↪   Bowl/BBM3/BBM_III_Regular_Season_Dump_Part_04_01302023.csv")
    bbm3_data_part_6 <- read.csv("~/Bestball Data
↪   Bowl/BBM3/BBM_III_Regular_Season_Dump_Part_05_01302023.csv")
    bbm3_data_part_7 <- read.csv("~/Bestball Data
↪   Bowl/BBM3/BBM_III_Regular_Season_Dump_Part_06_01302023.csv")
    bbm3_data_part_8 <- read.csv("~/Bestball Data
↪   Bowl/BBM3/BBM_III_Regular_Season_Dump_Part_07_01302023.csv")
    bbm3_data_part_9 <- read.csv("~/Bestball Data
↪   Bowl/BBM3/BBM_III_Regular_Season_Dump_Part_08_01302023.csv")
    bbm3_data_part_10 <- read.csv("~/Bestball Data
↪   Bowl/BBM3/BBM_III_Regular_Season_Dump_Part_09_01302023.csv")
    bbm3_data_part_11 <- read.csv("~/Bestball Data
↪   Bowl/BBM3/BBM_III_Regular_Season_Dump_Part_010_01302023.csv")
    bbm3_data_part_12 <- read.csv("~/Bestball Data
↪   Bowl/BBM3/BBM_III_Regular_Season_Dump_Part_011_01302023.csv")
```

```r
    bbm3_data_full <- bind_rows(bbm3_data_part_1, bbm3_data_part_2,
        bbm3_data_part_3, bbm3_data_part_4, bbm3_data_part_5,
        bbm3_data_part_6, bbm3_data_part_7, bbm3_data_part_8,
        bbm3_data_part_9, bbm3_data_part_10, bbm3_data_part_11,
        bbm3_data_part_12)

    return(bbm3_data_full)
}

generate_bbm_data <- function() {

    bbm2_data_full <- read.csv("~/Bestball Data
↪   Bowl/BBM2/BBM_II_Data_Dump_Regular_Season_01312022.csv")
    bbm3_data_full <- load_bbm3_data()

    bbm2_data_full$season <- 2021
    bbm3_data_full$season <- 2022

    bbm3_data_full <- select(bbm3_data_full, -c("draft_entry_id",
        "tournament_round_draft_entry_id"))

    bbm_data_full <- bind_rows(bbm2_data_full, bbm3_data_full)

    return(bbm_data_full)

}

bbm_data_full <- generate_bbm_data()
```

We'll start off by importing the data. I'm importing and combining BBM2 and BBM3 into one data set. I'm also adding a "Season" variable to keep track of which teams are BBM2 and BBM3. I didn't end up needing to use this, but it's helpful for future analysis.

```r
bbm_value_curve <- read.csv("~/Bestball Data Bowl/Underdog Draft Capital by Pick -
↪   Underdog.csv") %>%
    select(overall_pick_number, draft_capital_adj)
```

Next I'm importing a draft pick value curve, I'm using the same one that Mike Leone used in his Best Ball Mania Manifesto. It can be found here.

Now that we've got all of our raw data, we'll get into some cleaning and adding variables.

```r
add_qb_capital <- function(bbm_data, bbm_teams, value_curve) {

    bbm_data <- merge(bbm_data, value_curve, by = "overall_pick_number",
        all.x = TRUE)

    bbm_data <- bbm_data[bbm_data$position_name == "QB", ]

    bbm_data <- aggregate(draft_capital_adj ~ tournament_entry_id,
        bbm_data, FUN = sum)

    team_data <- merge(bbm_teams, bbm_data, by = "tournament_entry_id",
```

```
        all.x = TRUE)

    colnames(team_data)[which(names(team_data) == "draft_capital_adj")] <-
    ↪  "qb_draft_capital"

    return(team_data)
}

add_rb_capital <- function(bbm_data, bbm_teams, value_curve) {

    bbm_data <- merge(bbm_data, value_curve, by = "overall_pick_number",
        all.x = TRUE)

    bbm_data <- bbm_data[bbm_data$position_name == "RB", ]

    bbm_data <- aggregate(draft_capital_adj ~ tournament_entry_id,
        bbm_data, FUN = sum)

    team_data <- merge(bbm_teams, bbm_data, by = "tournament_entry_id",
        all.x = TRUE)

    colnames(team_data)[which(names(team_data) == "draft_capital_adj")] <-
    ↪  "rb_draft_capital"

    return(team_data)

}

add_wr_capital <- function(bbm_data, bbm_teams, value_curve) {

    bbm_data <- merge(bbm_data, value_curve, by = "overall_pick_number",
        all.x = TRUE)

    bbm_data <- bbm_data[bbm_data$position_name == "WR", ]

    bbm_data <- aggregate(draft_capital_adj ~ tournament_entry_id,
        bbm_data, FUN = sum)

    team_data <- merge(bbm_teams, bbm_data, by = "tournament_entry_id",
        all.x = TRUE)

    colnames(team_data)[which(names(team_data) == "draft_capital_adj")] <-
    ↪  "wr_draft_capital"

    return(team_data)

}

add_te_capital <- function(bbm_data, bbm_teams, value_curve) {

    bbm_data <- merge(bbm_data, value_curve, by = "overall_pick_number",
        all.x = TRUE)
```

```
    bbm_data <- bbm_data[bbm_data$position_name == "TE", ]

    bbm_data <- aggregate(draft_capital_adj ~ tournament_entry_id,
        bbm_data, FUN = sum)

    team_data <- merge(bbm_teams, bbm_data, by = "tournament_entry_id",
        all.x = TRUE)

    colnames(team_data)[which(names(team_data) == "draft_capital_adj")] <-
    ↪  "te_draft_capital"

    return(team_data)

}
```

Each of these functions left-merges the value curve into the full BBM data set then subsets by the respective position. Next, they aggregate the draft capital by BBM entry and merge it into the team data that was passed into it initially.

```
generate_qb_quantiles <- function(bbm_teams) {

    qb_quantiles <- quantile(bbm_teams$qb_draft_capital, probs = seq(0,
        1, 0.2))
    qb_quantiles_1 <- qb_quantiles[2]
    qb_quantiles_2 <- qb_quantiles[3]
    qb_quantiles_3 <- qb_quantiles[4]
    qb_quantiles_4 <- qb_quantiles[5]

    bbm_teams <- bbm_teams %>%
        rowwise() %>%
        mutate(qb_quantile = case_when(qb_draft_capital < qb_quantiles_1 ~
            "Lowest Draft Capital", qb_draft_capital >= qb_quantiles_1 &
            qb_draft_capital < qb_quantiles_2 ~ "Low Draft Capital",
            qb_draft_capital >= qb_quantiles_2 & qb_draft_capital <
                qb_quantiles_3 ~ "Average Draft Capital", qb_draft_capital >=
                qb_quantiles_3 & qb_draft_capital < qb_quantiles_4 ~
                "High Draft Capital", qb_draft_capital >= qb_quantiles_4 ~
                "Highest Draft Capital"))

    bbm_teams$qb_quantile <- factor(bbm_teams$qb_quantile, levels = c("Lowest Draft
↪   Capital",
        "Low Draft Capital", "Average Draft Capital", "High Draft Capital",
        "Highest Draft Capital"))
    return(bbm_teams)
}
generate_rb_quantiles <- function(bbm_teams) {

    rb_quantiles <- quantile(bbm_teams$rb_draft_capital, probs = seq(0,
        1, 0.2))
    rb_quantiles_1 <- rb_quantiles[2]
    rb_quantiles_2 <- rb_quantiles[3]
    rb_quantiles_3 <- rb_quantiles[4]
    rb_quantiles_4 <- rb_quantiles[5]
```

```
    bbm_teams <- bbm_teams %>%
        rowwise() %>%
        mutate(rb_quantile = case_when(rb_draft_capital < rb_quantiles_1 ~
            "Lowest Draft Capital", rb_draft_capital >= rb_quantiles_1 &
            rb_draft_capital < rb_quantiles_2 ~ "Low Draft Capital",
            rb_draft_capital >= rb_quantiles_2 & rb_draft_capital <
                rb_quantiles_3 ~ "Average Draft Capital", rb_draft_capital >=
                rb_quantiles_3 & rb_draft_capital < rb_quantiles_4 ~
                "High Draft Capital", rb_draft_capital >= rb_quantiles_4 ~
                "Highest Draft Capital"))

    bbm_teams$rb_quantile <- factor(bbm_teams$rb_quantile, levels = c("Lowest Draft
↪   Capital",
        "Low Draft Capital", "Average Draft Capital", "High Draft Capital",
        "Highest Draft Capital"))
    return(bbm_teams)

}
generate_wr_quantiles <- function(bbm_teams) {

    wr_quantiles <- quantile(bbm_teams$wr_draft_capital, probs = seq(0,
        1, 0.2))
    wr_quantiles_1 <- wr_quantiles[2]
    wr_quantiles_2 <- wr_quantiles[3]
    wr_quantiles_3 <- wr_quantiles[4]
    wr_quantiles_4 <- wr_quantiles[5]

    bbm_teams <- bbm_teams %>%
        rowwise() %>%
        mutate(wr_quantile = case_when(wr_draft_capital < wr_quantiles_1 ~
            "Lowest Draft Capital", wr_draft_capital >= wr_quantiles_1 &
            wr_draft_capital < wr_quantiles_2 ~ "Low Draft Capital",
            wr_draft_capital >= wr_quantiles_2 & wr_draft_capital <
                wr_quantiles_3 ~ "Average Draft Capital", wr_draft_capital >=
                wr_quantiles_3 & wr_draft_capital < wr_quantiles_4 ~
                "High Draft Capital", wr_draft_capital >= wr_quantiles_4 ~
                "Highest Draft Capital"))

    bbm_teams$wr_quantile <- factor(bbm_teams$wr_quantile, levels = c("Lowest Draft
↪   Capital",
        "Low Draft Capital", "Average Draft Capital", "High Draft Capital",
        "Highest Draft Capital"))
    return(bbm_teams)

}
generate_te_quantiles <- function(bbm_teams) {

    te_quantiles <- quantile(bbm_teams$te_draft_capital, probs = seq(0,
        1, 0.2))
    te_quantiles_1 <- te_quantiles[2]
    te_quantiles_2 <- te_quantiles[3]
    te_quantiles_3 <- te_quantiles[4]
```

```
    te_quantiles_4 <- te_quantiles[5]

    bbm_teams <- bbm_teams %>%
        rowwise() %>%
        mutate(te_quantile = case_when(te_draft_capital < te_quantiles_1 ~
            "Lowest Draft Capital", te_draft_capital >= te_quantiles_1 &
            te_draft_capital < te_quantiles_2 ~ "Low Draft Capital",
            te_draft_capital >= te_quantiles_2 & te_draft_capital <
                te_quantiles_3 ~ "Average Draft Capital", te_draft_capital >=
                te_quantiles_3 & te_draft_capital < te_quantiles_4 ~
                "High Draft Capital", te_draft_capital >= te_quantiles_4 ~
                "Highest Draft Capital"))

    bbm_teams$te_quantile <- factor(bbm_teams$te_quantile, levels = c("Lowest Draft
↪   Capital",
        "Low Draft Capital", "Average Draft Capital", "High Draft Capital",
        "Highest Draft Capital"))
    return(bbm_teams)

}
```

These functions take the team data with the position by position draft capital we just added and put them into one of five buckets depending on much draft capital they spent at the position. I named the buckets: Lowest Draft Capital, Low Draft Capital, Average Draft Capital, High Draft Capital, and Highest Draft Capital. This makes the visualizations much more readable outside the context of this document.

```
add_num_qb <- function(bbm_data, bbm_teams) {

    bbm_data <- bbm_data[bbm_data$position_name == "QB", ]

    bbm_data <- bbm_data %>%
        mutate(num_qb = 1)

    bbm_data <- aggregate(num_qb ~ tournament_entry_id, bbm_data,
        FUN = sum)

    bbm_teams <- merge(bbm_teams, bbm_data, by = "tournament_entry_id",
        all.x = TRUE)

    return(bbm_teams)
}
add_num_rb <- function(bbm_data, bbm_teams) {

    bbm_data <- bbm_data[bbm_data$position_name == "RB", ]

    bbm_data <- bbm_data %>%
        mutate(num_rb = 1)

    bbm_data <- aggregate(num_rb ~ tournament_entry_id, bbm_data,
        FUN = sum)

    bbm_teams <- merge(bbm_teams, bbm_data, by = "tournament_entry_id",
        all.x = TRUE)
```

```
    return(bbm_teams)
}
add_num_wr <- function(bbm_data, bbm_teams) {

    bbm_data <- bbm_data[bbm_data$position_name == "WR", ]

    bbm_data <- bbm_data %>%
        mutate(num_wr = 1)

    bbm_data <- aggregate(num_wr ~ tournament_entry_id, bbm_data,
        FUN = sum)

    bbm_teams <- merge(bbm_teams, bbm_data, by = "tournament_entry_id",
        all.x = TRUE)

    return(bbm_teams)
}
add_num_te <- function(bbm_data, bbm_teams) {

    bbm_data <- bbm_data[bbm_data$position_name == "TE", ]

    bbm_data <- bbm_data %>%
        mutate(num_te = 1)

    bbm_data <- aggregate(num_te ~ tournament_entry_id, bbm_data,
        FUN = sum)

    bbm_teams <- merge(bbm_teams, bbm_data, by = "tournament_entry_id",
        all.x = TRUE)

    return(bbm_teams)
}
```

The final functions add the quantity of each position that each team drafted.

```
teams <- select(bbm_data_full, "draft_time", "tournament_entry_id",
    "pick_order", "roster_points", "season") %>%
    distinct()

teams <- add_qb_capital(bbm_data_full, teams, bbm_value_curve)
teams <- add_rb_capital(bbm_data_full, teams, bbm_value_curve)
teams <- add_wr_capital(bbm_data_full, teams, bbm_value_curve)
teams <- add_te_capital(bbm_data_full, teams, bbm_value_curve)

teams <- generate_qb_quantiles(teams)
teams <- generate_rb_quantiles(teams)
teams <- generate_wr_quantiles(teams)
teams <- generate_te_quantiles(teams)

teams <- add_num_qb(bbm_data_full, teams)
teams <- add_num_rb(bbm_data_full, teams)
teams <- add_num_wr(bbm_data_full, teams)
```

```
teams <- add_num_te(bbm_data_full, teams)

teams$draft_time <- factor(month(ymd_hms(teams$draft_time), label = TRUE))
teams$pick_order <- factor(teams$pick_order)

expected_advance_points <- quantile(teams$roster_points, probs = 0.8333)
teams <- teams %>%
    mutate(expected_advance = case_when(roster_points >= expected_advance_points ~
        TRUE, roster_points < expected_advance_points ~ FALSE))

bbm_teams <- teams
```

With all of our functions created, we can generate the data set that we'll use in the next section. A dataframe with every BBM2 and BBM3 team is fed through each function to add all of the variables we'll need. I've also added a few additional variables such as the month the draft took place and expected_advance. "Expected advance" is a logical variable measuring whether or not a team achieved a regular season score in the top 16.66% of scores. It's effectively measuring whether or not a team should've advanced to the playoffs in a perfectly fair tournament. From a forward looking perspective, it's slightly more helpful than actual advance rate.

## Visualization and Analysis

My analysis is going to begin with identifying a few of the broadly successful advance rate maximizing constructions. We'll then take those constructions and try to figure out if they have measurably lower season-long ceilings. I'm going to intentionally cast a wide net because I don't want to overemphasize individual player outcomes. It would be trivial to find the specific QB-RB-WR-TE quantity and respective draft capital that had the highest advance rate and test the ceiling of that, but there are so many possible combinations that it would massively shrink our samples of each construction. Instead, we'll visualize each position with the roster construction variables calculated above to spot trends.

```
qb_advance_viz <- function(bbm_teams) {

    teams <- subset(bbm_teams, num_qb < 5 & num_qb > 0)

    teams <- aggregate(expected_advance ~ qb_quantile + num_qb,
        teams, FUN = mean)

    teams <- teams %>%
        mutate(advance_over_expected = expected_advance - 0.166667)

    g1 <- ggplot(teams, aes(as.factor(num_qb), advance_over_expected,
        fill = advance_over_expected > 0)) + geom_bar(stat = "identity",
        colour = "black") + scale_y_continuous(breaks = seq(-0.1,
        0.1, 0.01), labels = scales::percent, limits = c(-0.1,
        0.1)) + facet_grid(~qb_quantile) + labs(title = "Advance Rate Over Expected by
         ↪  Draft Capital and Quantity of QBs Rostered (BBM2 & BBM3)",
        x = "Number of QBs Rostered", y = "Advance Rate Over Expected (16.67%)",
        caption = "@JakeBoesFF") + scale_fill_manual(guide = "none",
        values = c("#FF6347", "#228B22")) + theme_light()

    return(g1)
}
```

```r
rb_advance_viz <- function(bbm_teams) {

    teams <- subset(bbm_teams, num_rb < 10 & num_rb > 2)

    teams <- aggregate(expected_advance ~ rb_quantile + num_rb,
        teams, FUN = mean)

    teams <- teams %>%
        mutate(advance_over_expected = expected_advance - 0.166667)

    g1 <- ggplot(teams, aes(as.factor(num_rb), advance_over_expected,
        fill = advance_over_expected > 0)) + geom_bar(stat = "identity",
        colour = "black") + scale_y_continuous(breaks = seq(-0.1,
        0.1, 0.01), labels = scales::percent, limits = c(-0.1,
        0.1)) + facet_grid(~rb_quantile) + labs(title = "Advance Rate Over Expected by
        ↪  Draft Capital and Quantity of RBs Rostered (BBM2 & BBM3)",
        x = "Number of RBs Rostered", y = "Advance Rate Over Expected (16.67%)",
        caption = "@JakeBoesFF") + scale_fill_manual(guide = "none",
        values = c("#FF6347", "#228B22")) + theme_light()

    return(g1)
}
wr_advance_viz <- function(bbm_teams) {

    teams <- subset(bbm_teams, num_wr < 11 & num_wr > 3)

    teams <- aggregate(expected_advance ~ wr_quantile + num_wr,
        teams, FUN = mean)

    teams <- teams %>%
        mutate(advance_over_expected = expected_advance - 0.166667)

    g1 <- ggplot(teams, aes(as.factor(num_wr), advance_over_expected,
        fill = advance_over_expected > 0)) + geom_bar(stat = "identity",
        colour = "black") + scale_y_continuous(breaks = seq(-0.1,
        0.1, 0.01), labels = scales::percent, limits = c(-0.1,
        0.1)) + facet_grid(~wr_quantile) + labs(title = "Advance Rate Over Expected by
        ↪  Draft Capital and Quantity of WRs Rostered (BBM2 & BBM3)",
        x = "Number of WRs Rostered", y = "Advance Rate Over Expected (16.67%)",
        caption = "@JakeBoesFF") + scale_fill_manual(guide = "none",
        values = c("#FF6347", "#228B22")) + theme_light()

    return(g1)
}
te_advance_viz <- function(bbm_teams) {

    teams <- subset(bbm_teams, num_te < 5 & num_te > 0)

    teams <- aggregate(expected_advance ~ te_quantile + num_te,
        teams, FUN = mean)

    teams <- teams %>%
        mutate(advance_over_expected = expected_advance - 0.166667)
```

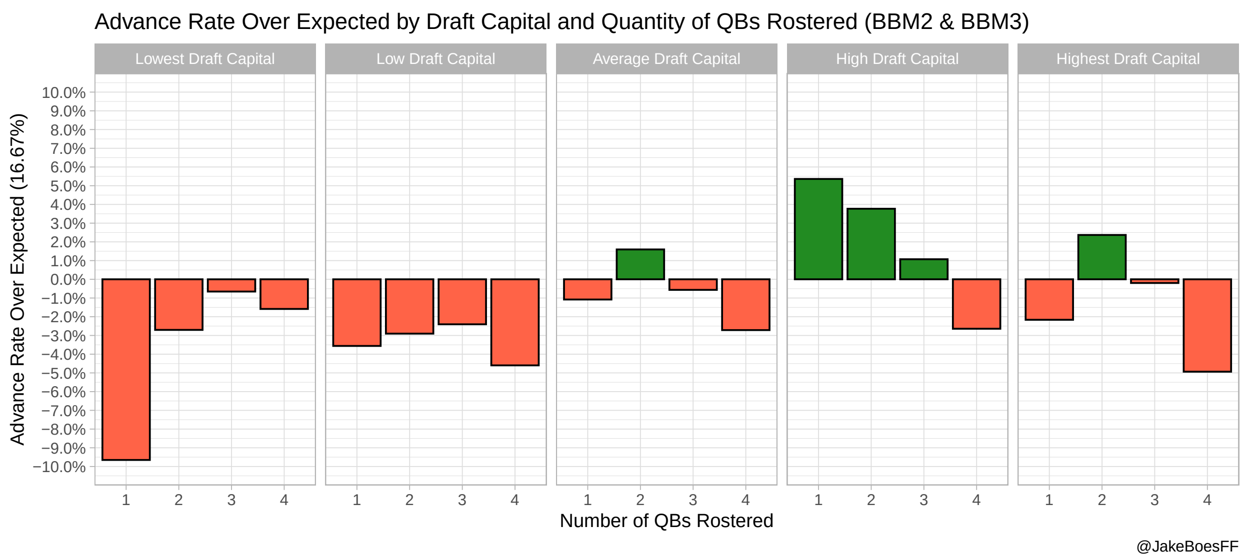
```r
    g1 <- ggplot(teams, aes(as.factor(num_te), advance_over_expected,
        fill = advance_over_expected > 0)) + geom_bar(stat = "identity",
        colour = "black") + scale_y_continuous(breaks = seq(-0.1,
        0.1, 0.01), labels = scales::percent, limits = c(-0.1,
        0.1)) + facet_grid(~te_quantile) + labs(title = "Advance Rate Over Expected by
        ↪ Draft Capital and Quantity of TEs Rostered (BBM2 & BBM3)",
        x = "Number of TEs Rostered", y = "Advance Rate Over Expected (16.67%)",
        caption = "@JakeBoesFF") + scale_fill_manual(guide = "none",
        values = c("#FF6347", "#228B22")) + theme_light()

    return(g1)
}
```

These functions subset the data to only include teams with what I deem to be a reasonable number of players at each position (You'll see what that is when I show the graphs). The data is aggregated to find the percentage of teams that should've advanced by both draft capital and the quantity of players at each position. A new variable is created to denote the expected advance rate of each construction minus the average advance rate and the data is plotted with a common y-axis for readability across graphs.

Green bars below will denote constructions that advance better than expected, red will denote lower than expected. Viz will be ordered QB, RB, WR, TE in each section.

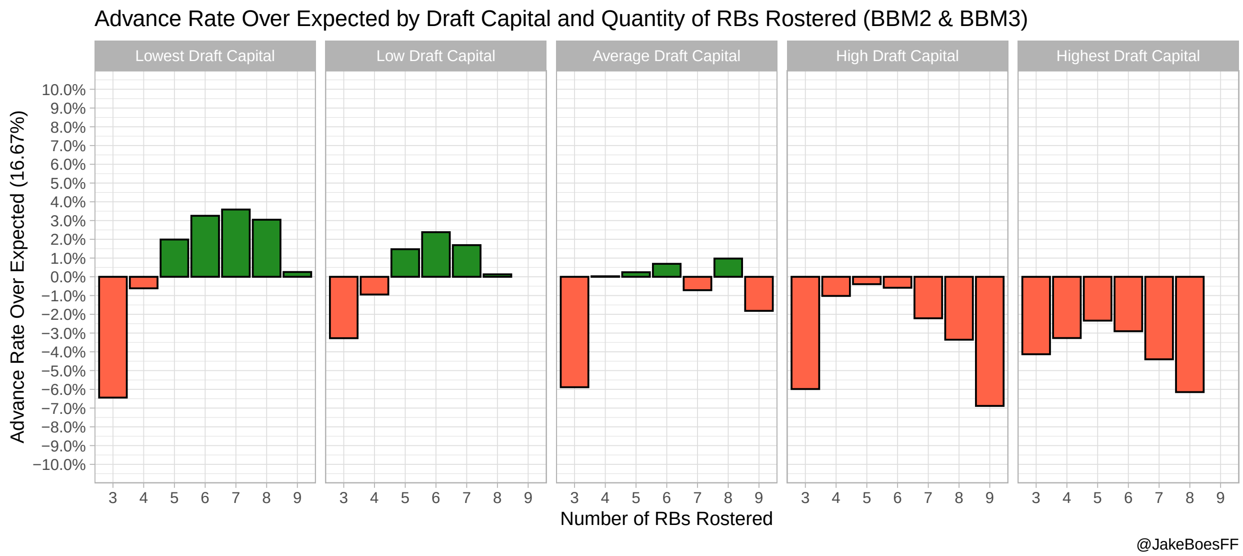
Let's get into it!

```r
qb_advance_viz(bbm_teams)
```

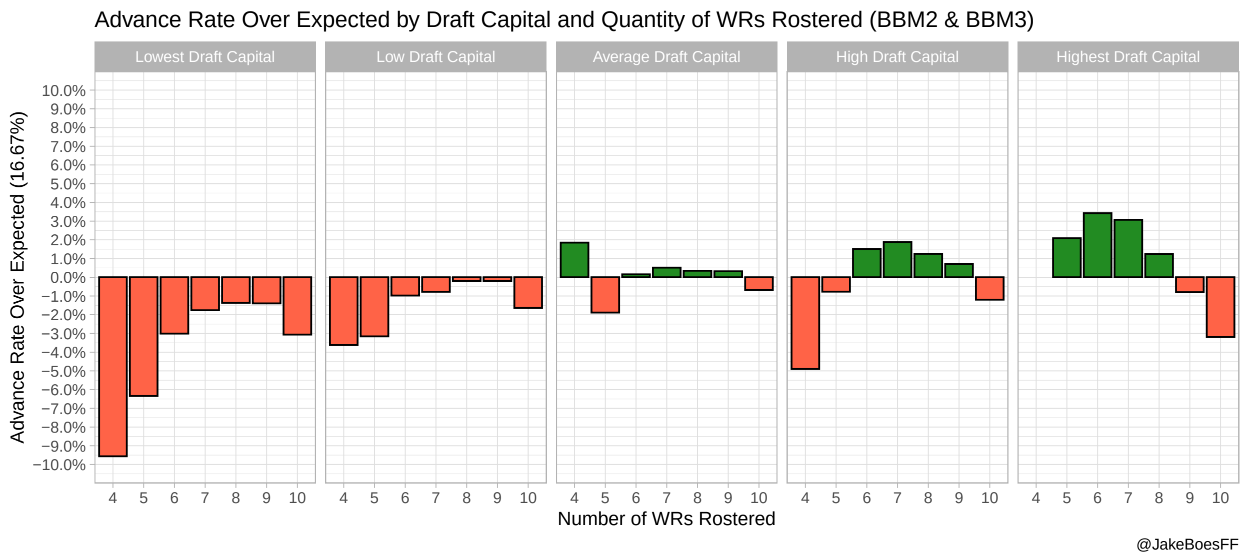



Beautiful! I'll note here before I share the rest of the graphs that I was very generous when I set the bounds for what a reasonable number of players would be at each position. I feel comfortable saying that 5 QBs isn't good and isn't worth looking into, but I don't feel that way about 4 QBs. That's a theme with these, we'll use our best judgement.

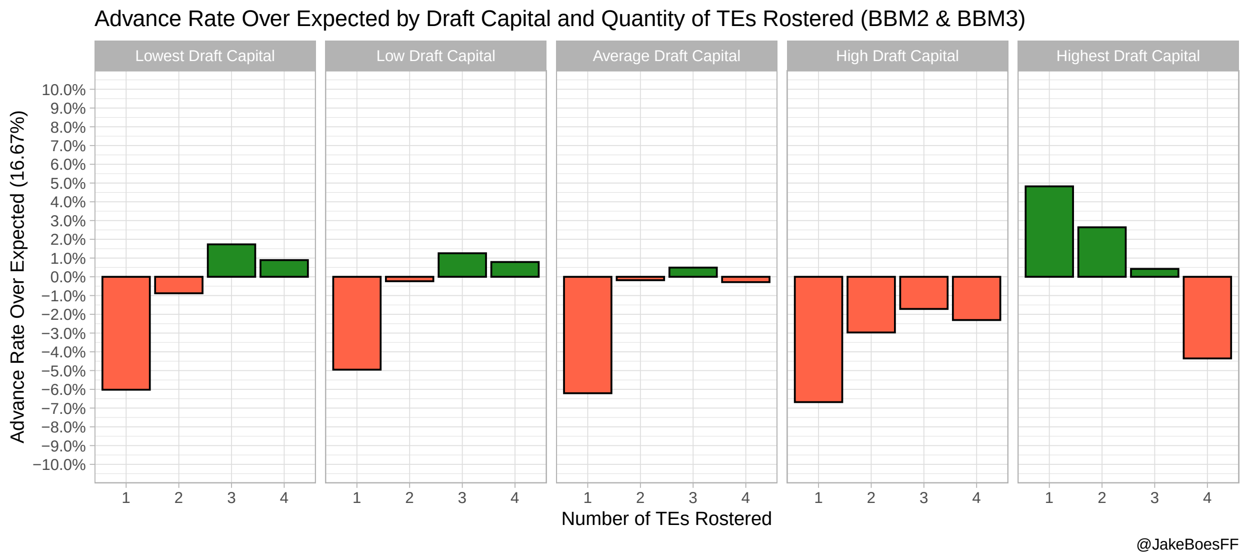
```r
rb_advance_viz(bbm_teams)
```




```
wr_advance_viz(bbm_teams)
```




```
te_advance_viz(bbm_teams)
```




The first thing I notice is that the fringe constructions (10 WR, 3 RB, etc.) are all pretty obviously terrible in terms of advance rate. The best constructions aren't quite as clear, but the few that I'm going to track as we look deeper are the following:

1. High spending at WR is great for your advance rate and it's not picky in terms of quantity.
2. Spending significant capital at QB is good for your advance rate, but overspending is possible. 2 QB in the top 2 buckets seems ideal.
3. High spending at RB has been pretty terrible for advance rate.

Let's reset. We can see the constructions above that will boost advance rates, how do we evaluate their season-long ceiling?

One challenge with analyzing ceiling outcomes is that you almost instantly run into issues with sample sizes. Particularly in something like BBM, where we only have two years of data. We only have so many teams that have crossed the top end point thresholds. Even tougher, a lot of those elite teams were built on specific player outcomes that probably aren't repeatable.

I'm going to try to remedy this issue first by comparing constructions to the average playoff team. I wouldn't consider making the playoffs in isolation to be a season-long ceiling outcome, but it's a reasonable threshold for success. If a team makes the playoffs, it's safe to assume that it avoided most of the catastrophic outcomes reasonably available based on construction. For example, a hyperfragile construction (Highest spending at RB, low quantity of RB) that makes the playoffs likely avoided multiple major injuries in their RB room. When we're only looking at the successful teams of each construction, we should see constructions with higher ceilings pull away.

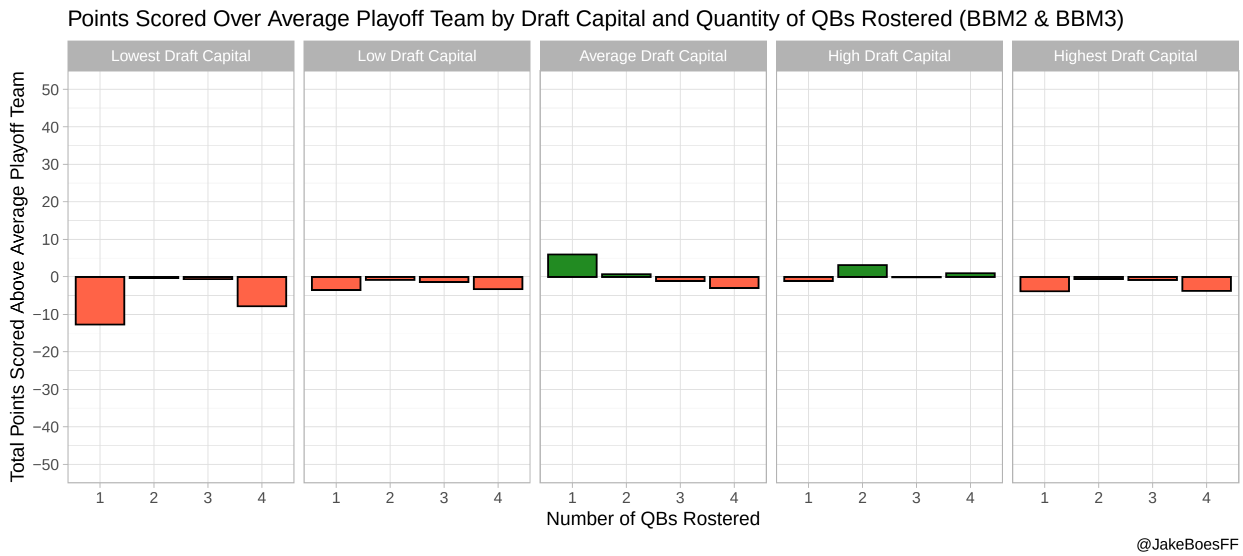
```
qb_ceiling_viz <- function(bbm_teams) {

    teams <- subset(bbm_teams, expected_advance == TRUE)
    advanced_average_points <- mean(teams$roster_points)
    teams <- subset(teams, num_qb < 5 & num_qb > 0)
    teams <- teams %>%
        mutate(points_above_average = roster_points - advanced_average_points)
    teams <- aggregate(points_above_average ~ qb_quantile + num_qb,
        teams, FUN = mean)
```

```r
    g1 <- ggplot(teams, aes(as.factor(num_qb), points_above_average,
        fill = points_above_average > 0)) + geom_bar(stat = "identity",
        colour = "black") + facet_grid(~qb_quantile) + labs(title = "Points Scored Over
        ↪  Average Playoff Team by Draft Capital and Quantity of QBs Rostered (BBM2 &
        ↪  BBM3)",
        x = "Number of QBs Rostered", y = "Total Points Scored Above Average Playoff
        ↪  Team",
        caption = "@JakeBoesFF") + scale_fill_manual(guide = FALSE,
        values = c("#FF6347", "#228B22")) + scale_y_continuous(breaks = seq(-50,
        50, 10), limits = c(-50, 50)) + theme_light()

    return(g1)

}
rb_ceiling_viz <- function(bbm_teams) {

    teams <- subset(bbm_teams, expected_advance == TRUE)
    advanced_average_points <- mean(teams$roster_points)
    teams <- subset(teams, num_rb < 10 & num_rb > 2)
    teams <- teams %>%
        mutate(points_above_average = roster_points - advanced_average_points)
    teams <- aggregate(points_above_average ~ rb_quantile + num_rb,
        teams, FUN = mean)

    g1 <- ggplot(teams, aes(as.factor(num_rb), points_above_average,
        fill = points_above_average > 0)) + geom_bar(stat = "identity",
        colour = "black") + facet_grid(~rb_quantile) + labs(title = "Points Scored Over
        ↪  Average Playoff Team by Draft Capital and Quantity of RBs Rostered (BBM2 &
        ↪  BBM3)",
        x = "Number of RBs Rostered", y = "Total Points Scored Above Average Playoff
        ↪  Team",
        caption = "@JakeBoesFF") + scale_fill_manual(guide = FALSE,
        values = c("#FF6347", "#228B22")) + scale_y_continuous(breaks = seq(-50,
        50, 10), limits = c(-50, 50)) + theme_light()

    return(g1)

}
wr_ceiling_viz <- function(bbm_teams) {

    teams <- subset(bbm_teams, expected_advance == TRUE)
    advanced_average_points <- mean(teams$roster_points)
    teams <- subset(teams, num_wr < 11 & num_wr > 3)
    teams <- teams %>%
        mutate(points_above_average = roster_points - advanced_average_points)
    teams <- aggregate(points_above_average ~ wr_quantile + num_wr,
        teams, FUN = mean)

    g1 <- ggplot(teams, aes(as.factor(num_wr), points_above_average,
        fill = points_above_average > 0)) + geom_bar(stat = "identity",
        colour = "black") + facet_grid(~wr_quantile) + labs(title = "Points Scored Over
        ↪  Average Playoff Team by Draft Capital and Quantity of WRs Rostered (BBM2 &
        ↪  BBM3)",
```

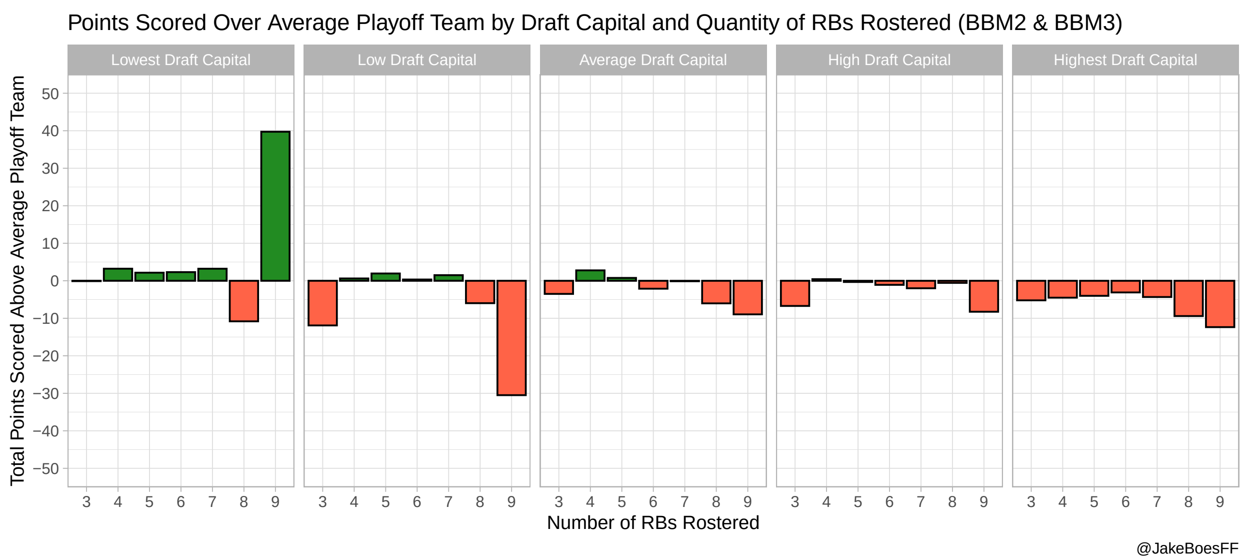
```
        x = "Number of WRs Rostered", y = "Total Points Scored Above Average Playoff
        ↪  Team",
        caption = "@JakeBoesFF") + scale_fill_manual(guide = FALSE,
        values = c("#FF6347", "#228B22")) + scale_y_continuous(breaks = seq(-50,
        50, 10), limits = c(-50, 50)) + theme_light()

    return(g1)

}
te_ceiling_viz <- function(bbm_teams) {

    teams <- subset(bbm_teams, expected_advance == TRUE)
    advanced_average_points <- mean(teams$roster_points)
    teams <- subset(teams, num_te < 5 & num_te > 0)
    teams <- teams %>%
        mutate(points_above_average = roster_points - advanced_average_points)
    teams <- aggregate(points_above_average ~ te_quantile + num_te,
        teams, FUN = mean)

    g1 <- ggplot(teams, aes(as.factor(num_te), points_above_average,
        fill = points_above_average > 0)) + geom_bar(stat = "identity",
        colour = "black") + facet_grid(~te_quantile) + labs(title = "Points Scored Over
        ↪  Average Playoff Team by Draft Capital and Quantity of TEs Rostered (BBM2 &
        ↪  BBM3)",
        x = "Number of TEs Rostered", y = "Total Points Scored Above Average Playoff
        ↪  Team",
        caption = "@JakeBoesFF") + scale_fill_manual(guide = FALSE,
        values = c("#FF6347", "#228B22")) + scale_y_continuous(breaks = seq(-50,
        50, 10), limits = c(-50, 50)) + theme_light()

    return(g1)

}
```

We're going to visualize by position once again. These functions subset the data by the same positional limits I used previously, then remove the teams that didn't score enough to advance and calculate the points scored above the average playoff team. I then once again aggregate by our roster construction metrics and plot away!

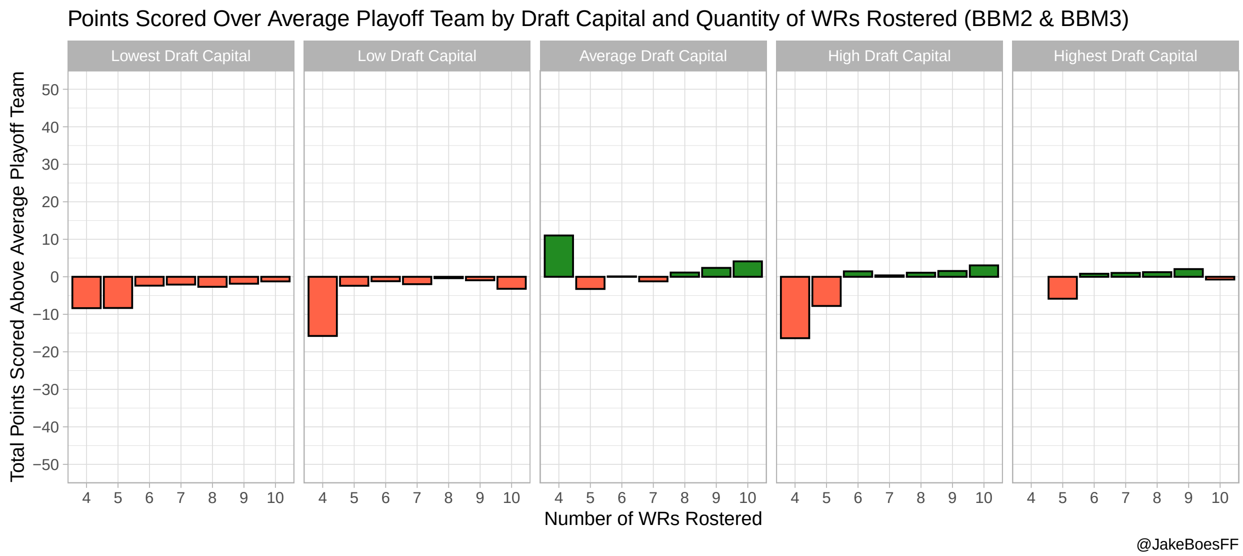
```
qb_ceiling_viz(bbm_teams)
```



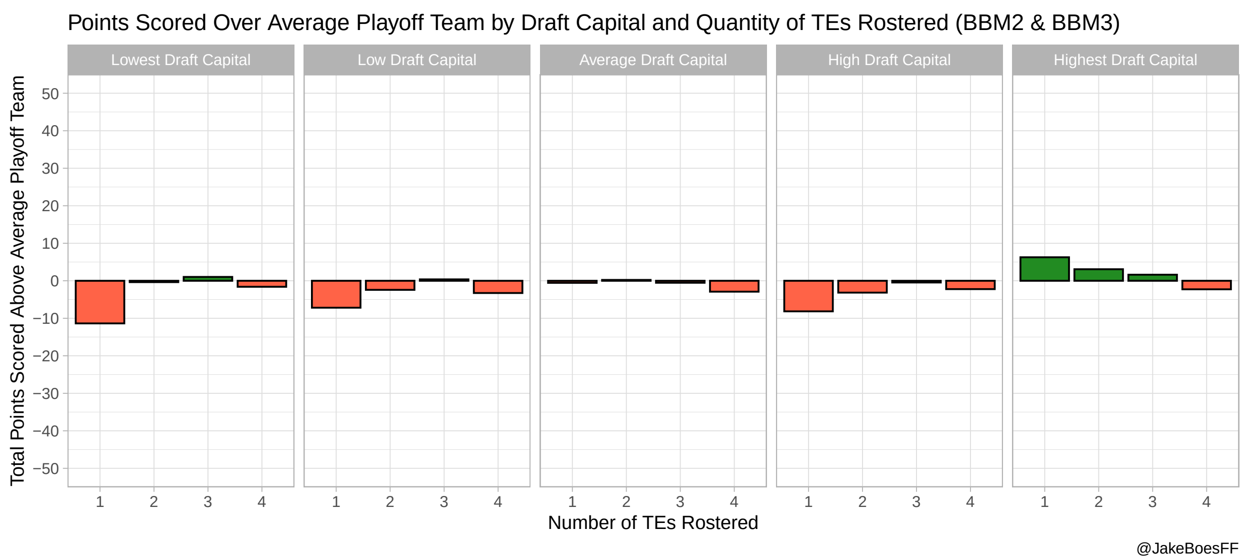

```
rb_ceiling_viz(bbm_teams)
```




```
wr_ceiling_viz(bbm_teams)
```




```
te_ceiling_viz(bbm_teams)
```




These can be a little confusing to digest at first. My suggestion would be to only look at the constructions that we know to be reasonably viable based on advance rates first. (I.E. Don't bother looking at the far ends of each bucket)

If the advance rate optimized constructions had lower season-long ceilings, we may see them lag behind their playoff peers above. However, there's very little variation in season-long points scored among playoff teams in terms of construction. I'd argue that, given a team has advanced, there's little difference in how we should expect that team to have scored based on roster construction. Let's revisit our three advance rate optimized constructions:

1. Playoff teams that are heavily invested in WRs do score a little bit more than teams with low investment, but it's a tiny difference. Roughly 5 more points on average.
2. Spending up at QB essentially didn't matter at all given you advanced. If you go late round QBs and make the playoffs, you'll be scoring right alongside the teams with elite QBs.

3. Spending down at RB appears to be good for your ceiling in addition to being good for your advance rate. I was shocked that playoff teams with heavy RB spending and only 3-4 RBs (The hyperfragile teams) didn't have higher scores. The payoff hasn't been there.

I wrote earlier that I didn't want to massively shrink the sample, but perhaps the roster constructions with the massive ceilings only show up at the very tail end of the distribution. Let's take a look at the top 1% of teams across BBM2 and BBM3 by roster construction. This is a hair over 6K teams total. Since all of these teams represent ceiling outcomes, I'm going to look at prevalence of constructions compared to expectation instead of points scored.

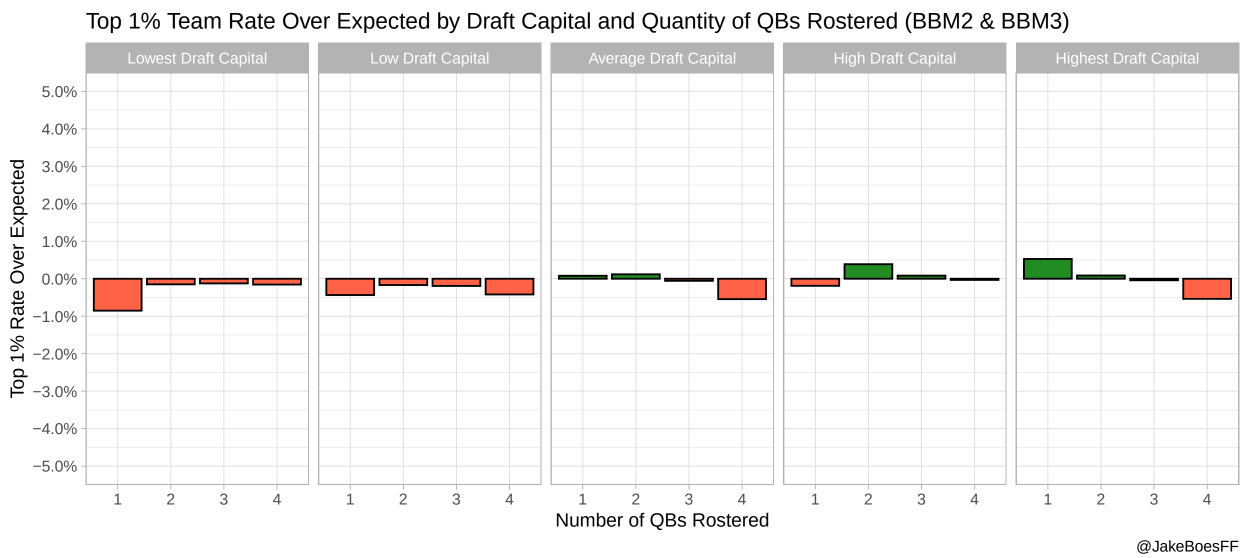
```
qb_max_prevalence <- function(bbm_teams) {

    teams <- subset(bbm_teams, num_qb < 5 & num_qb > 0)
    max_ceiling_threshold <- quantile(teams$roster_points, probs = 0.99)
    teams <- teams %>%
        mutate(top_1_percent = case_when(roster_points > max_ceiling_threshold ~
            TRUE, roster_points < max_ceiling_threshold ~ FALSE))
    teams <- aggregate(top_1_percent ~ qb_quantile + num_qb,
        teams, FUN = mean)
    teams <- teams %>%
        mutate(top_1_percent_over_expected = top_1_percent -
            0.01)

    g1 <- ggplot(teams, aes(as.factor(num_qb), top_1_percent_over_expected,
        fill = top_1_percent_over_expected > 0)) + geom_bar(stat = "identity",
        colour = "black") + scale_y_continuous(breaks = seq(-0.05,
        0.05, 0.01), labels = scales::percent, limits = c(-0.05,
        0.05)) + facet_grid(~qb_quantile) + labs(title = "Top 1% Team Rate Over Expected
     ↪  by Draft Capital and Quantity of QBs Rostered (BBM2 & BBM3)",
        x = "Number of QBs Rostered", y = "Top 1% Rate Over Expected",
        caption = "@JakeBoesFF") + scale_fill_manual(guide = FALSE,
        values = c("#FF6347", "#228B22")) + theme_light()

    return(g1)

}
rb_max_prevalence <- function(bbm_teams) {

    teams <- subset(bbm_teams, num_rb < 10 & num_rb > 2)
    max_ceiling_threshold <- quantile(teams$roster_points, probs = 0.99)
    teams <- teams %>%
        mutate(top_1_percent = case_when(roster_points > max_ceiling_threshold ~
            TRUE, roster_points < max_ceiling_threshold ~ FALSE))
    teams <- aggregate(top_1_percent ~ rb_quantile + num_rb,
        teams, FUN = mean)
    teams <- teams %>%
        mutate(top_1_percent_over_expected = top_1_percent -
            0.01)

    g1 <- ggplot(teams, aes(as.factor(num_rb), top_1_percent_over_expected,
        fill = top_1_percent_over_expected > 0)) + geom_bar(stat = "identity",
        colour = "black") + scale_y_continuous(breaks = seq(-0.05,
        0.05, 0.01), labels = scales::percent, limits = c(-0.05,
        0.05)) + facet_grid(~rb_quantile) + labs(title = "Top 1% Team Rate Over Expected
     ↪  by Draft Capital and Quantity of RBs Rostered (BBM2 & BBM3)",
```

```
        x = "Number of RBs Rostered", y = "Top 1% Rate Over Expected",
        caption = "@JakeBoesFF") + scale_fill_manual(guide = FALSE,
        values = c("#FF6347", "#228B22")) + theme_light()

    return(g1)

}
wr_max_prevalence <- function(bbm_teams) {

    teams <- subset(bbm_teams, num_wr < 11 & num_wr > 3)
    max_ceiling_threshold <- quantile(teams$roster_points, probs = 0.99)
    teams <- teams %>%
        mutate(top_1_percent = case_when(roster_points > max_ceiling_threshold ~
            TRUE, roster_points < max_ceiling_threshold ~ FALSE))
    teams <- aggregate(top_1_percent ~ wr_quantile + num_wr,
        teams, FUN = mean)
    teams <- teams %>%
        mutate(top_1_percent_over_expected = top_1_percent -
            0.01)

    g1 <- ggplot(teams, aes(as.factor(num_wr), top_1_percent_over_expected,
        fill = top_1_percent_over_expected > 0)) + geom_bar(stat = "identity",
        colour = "black") + scale_y_continuous(breaks = seq(-0.05,
        0.05, 0.01), labels = scales::percent, limits = c(-0.05,
        0.05)) + facet_grid(~wr_quantile) + labs(title = "Top 1% Team Rate Over Expected
     ↪  by Draft Capital and Quantity of WRs Rostered (BBM2 & BBM3)",
        x = "Number of WRs Rostered", y = "Top 1% Rate Over Expected",
        caption = "@JakeBoesFF") + scale_fill_manual(guide = FALSE,
        values = c("#FF6347", "#228B22")) + theme_light()

    return(g1)

}
te_max_prevalence <- function(bbm_teams) {

    teams <- subset(bbm_teams, num_te < 5 & num_te > 0)
    max_ceiling_threshold <- quantile(teams$roster_points, probs = 0.99)
    teams <- teams %>%
        mutate(top_1_percent = case_when(roster_points > max_ceiling_threshold ~
            TRUE, roster_points < max_ceiling_threshold ~ FALSE))
    teams <- aggregate(top_1_percent ~ te_quantile + num_te,
        teams, FUN = mean)
    teams <- teams %>%
        mutate(top_1_percent_over_expected = top_1_percent -
            0.01)

    g1 <- ggplot(teams, aes(as.factor(num_te), top_1_percent_over_expected,
        fill = top_1_percent_over_expected > 0)) + geom_bar(stat = "identity",
        colour = "black") + scale_y_continuous(breaks = seq(-0.05,
        0.05, 0.01), labels = scales::percent, limits = c(-0.05,
        0.05)) + facet_grid(~te_quantile) + labs(title = "Top 1% Team Rate Over Expected
     ↪  by Draft Capital and Quantity of TEs Rostered (BBM2 & BBM3)",
        x = "Number of TEs Rostered", y = "Top 1% Rate Over Expected",
```

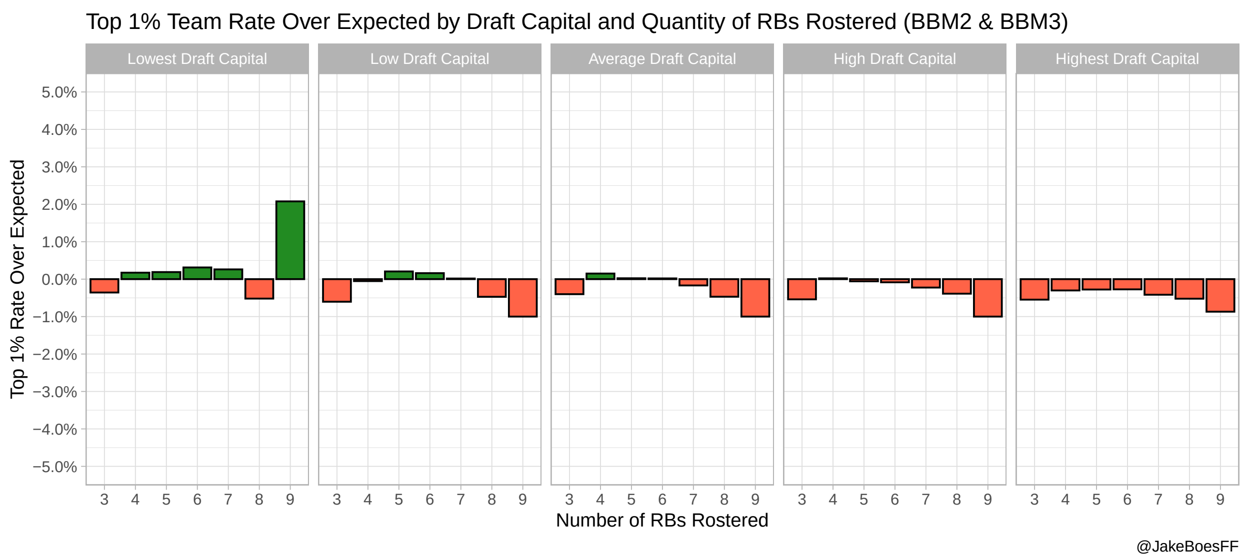
```r
        caption = "@JakeBoesFF") + scale_fill_manual(guide = FALSE,
        values = c("#FF6347", "#228B22")) + theme_light()

    return(g1)

}
```

One function for each position. We're setting a 99th percentile point threshold and creating a subset of only the teams above it.

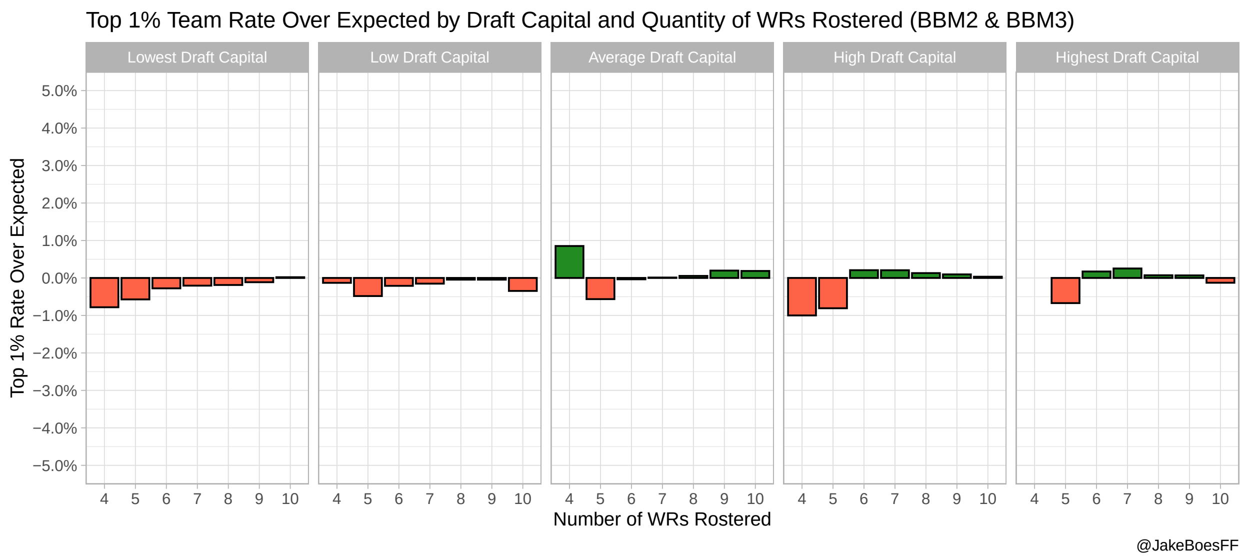
```r
qb_max_prevalence(bbm_teams)
```



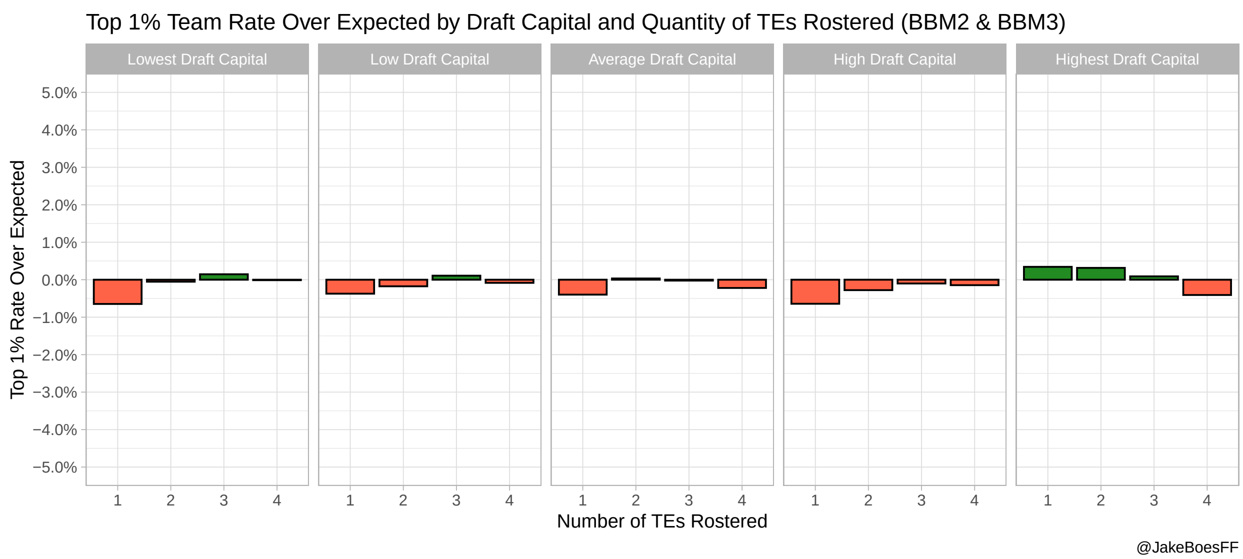

```r
rb_max_prevalence(bbm_teams)
```



```
wr_max_prevalence(bbm_teams)
```




```
te_max_prevalence(bbm_teams)
```




If all of these were equally viable constructions to build a top 1% team, we'd expect to see 0% across the board above. We do not. If the advance rate maximizing constructions had lower season-long ceilings, we may see them perform worse than expectation above.

Let's revisit our three advance rate maximizing constructions:

1. High WR spending continues to be correlated with both high advance rates and high regular season scoring.
2. Expensive QBs shine in the top 1%.
3. High RB spending comes out of this looking downright nonviable. If there's a silver lining, it looks like average RB spending is viable if you're shooting for ceiling outcomes. These are likely anchor RB and double anchor RB builds.

## Conclusion

My goal was to investigate whether optimizing roster construction for advance rate would reduce season-long ceiling outcomes in Best Ball Mania and it seems that it does not. It's possible that there are constructions that are underrepresented across the spectrum until the very top (Top ~100 teams) but I don't think they could be confidently identified with only two years of data. After a few more years of BBM this would be interesting to investigate. Generally speaking, the constructions that provide the highest advance rates are also providing strongest season-long ceiling.