UDisc Analysis

Bryan Jacques

UDisc Player and Course Analysis

This analysis uses data from Udisc's website, which records scores and other data associated with professional disc golf tournaments played throughout the year. It will assume the reader understands the basics of disc golf, which is basically the same as traditional golf, but instead of hitting balls into a cup the players throw discs into a basket. Individual statistics will be examined, as well as statistics pertaining to specific courses and holes.

This analysis will be limited to a subset of tournaments from the professional schedule. The data for the tournaments was taken from here. Results were copied and pasted into excel, saved individually, and a script was made to compile all the scores into one dataset (Compile_Scores.R). Two other datasets were manually created in excel to map the courses to the specific tournaments and rounds they were played, and to track par and length for each hole played (Data/Round_Course_Map.xlsx and Data/Course_Hole_Map.xlsx respectively)

Setup

The first thing we need to do is set up our environment to run the analysis

```
library(tidyverse)
library(here)
library(readxl)
library(flextable)
library(ggalt)
```

```
## Warning: package 'ggalt' was built under R version 4.1.2
```

```
hole_scores <- read_csv(here("Data/Hole_Scores.csv"))
round_course_map <- read_excel(here("Data/Round_Course_Map.xlsx"))
course_hole_map <- read_excel(here("Data/Course_Hole_Map.xlsx"))

hole_scores <- hole_scores %>%
  left_join(round_course_map, by = c("tournament_short","round")) %>%
  left_join(course_hole_map, by = c("course","hole")) %>%
  mutate(par_score = score_val - par)
```

Tournament round results were stored in dataset hole__scores, the round and course mapping was stored in round__course__map, and the length and par information for each hole was stored in course__hole__map. round__course__map and course__hole__map were joined to the hole__scores dataset to get course and length information for every hole played. Let's preview the hole__scores dataset to make sure the data was imported and joined properly:

```
hole_scores %>%
  sample_n(3) %>%
  flextable() %>%
  theme_zebra()
```

```
## Warning: Warning: fonts used in ‘flextable‘ are ignored because the ‘pdflatex‘
## engine is used and not ‘xelatex‘ or ‘lualatex‘. You can avoid this warning
## by using the ‘set_flextable_defaults(fonts_ignore=TRUE)‘ command or use a
## compatible engine by defining ‘latex_engine: xelatex‘ in the YAML header of the
## R Markdown document.
```

hole	player	file	score__val	round	tournamer	tournamer	course	distance
16	Chris Dickerson	GMC3.xlsx	3	3	Green Mountain Championship	GMC	Brewster Ridge	365
3	Tristan Tanner	PORTLAND2.xlsx	4	2	Portland Open	Portland	Glendoveer	777
8	Bradley Williams	PRESERVE	2	3	The Preserve Championship	Preserve	Airborn Preserve	406

Everything is looking good! The subset of tournaments included in this analysis are as follows:

```
hole_scores %>%
  select(tournament_short) %>%
  distinct() %>%
  flextable() %>%
  theme_zebra()
```

```
## Warning: Warning: fonts used in ‘flextable‘ are ignored because the ‘pdflatex‘
## engine is used and not ‘xelatex‘ or ‘lualatex‘. You can avoid this warning
## by using the ‘set_flextable_defaults(fonts_ignore=TRUE)‘ command or use a
## compatible engine by defining ‘latex_engine: xelatex‘ in the YAML header of the
## R Markdown document.
```

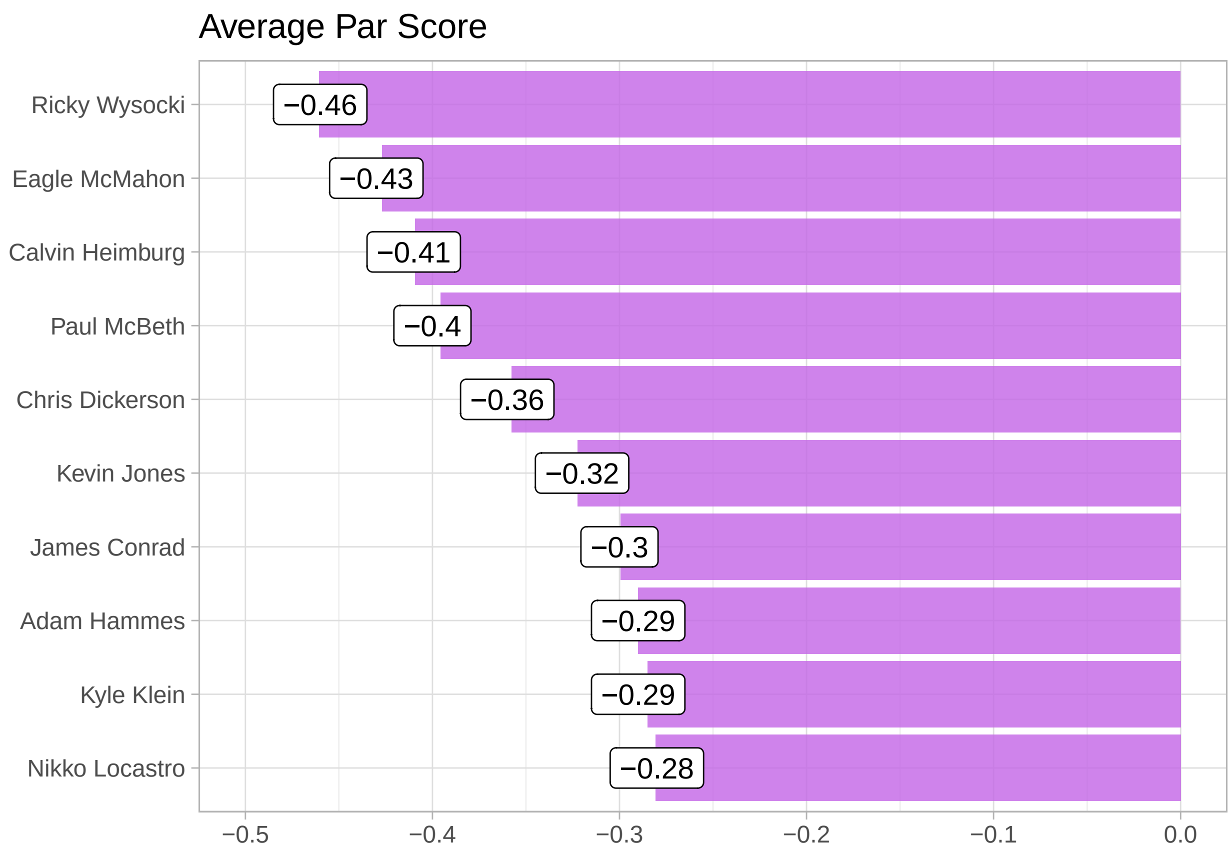
tournament_short
Des Moines
DGLO
GMC
Idlewild
Jonesboro
Ledgestone
LV Challenge
MVP Open
OTB Open
Portland
Preserve
Texas State
Waco

Player Analysis

Average Par Score

The first thing I'd like to look at is who scores the best in relation to par. To do this, let's take a look at the average score in relation to par for some of the best performing players. Let's also limit this to players who have only played in 8 different events out of the 13 included.

```r
hole_scores %>%
  group_by(player) %>%
  summarize(
    events_played = n_distinct(tournament_short)
    ,holes_played = n()
    ,avg_score = mean(score_val)
    ,avg_par_score = mean(par_score)
    ,round_avg_par_score = round(avg_par_score,2)
  ) %>%
  arrange(avg_par_score) %>%
  filter(
    events_played >= 8
  ) %>%
  top_n(10,wt = desc(avg_par_score)) %>%
  ggplot() +
  geom_bar(mapping = aes(x = reorder(player,desc(avg_par_score)), y = avg_par_score), stat = "identity"
  geom_label(mapping = aes(x = player, y = round_avg_par_score, label = round_avg_par_score)) +
  ylim(-.5,0) +
  coord_flip() +
  theme_light() +
  ggtitle("Average Par Score") +
  labs(x = NULL, y = NULL)
```



For those that follow professional disc golf at all, it's not really surprising at all to see these names at the top, with Ricky Wysocki performing the best, averaging 0.46 strokes under par across all holes played. Just so we can use it later to limit our analysis for other statistics, let's get the top 50 players by average_par_score and save it to a separate dataset called top_50.

```
top_50 <- hole_scores %>%
  group_by(player) %>%
  summarize(
    events_played = n_distinct(tournament_short)
    ,avg_par_score = mean(par_score)
  ) %>%
  arrange(avg_par_score) %>%
  filter(
    events_played >= 8
  ) %>%
  top_n(50,wt = desc(avg_par_score))

top_50 %>%
  head() %>%
  flextable() %>%
  theme_zebra()
```

```
## Warning: Warning: fonts used in `flextable` are ignored because the `pdflatex`
## engine is used and not `xelatex` or `lualatex`. You can avoid this warning
## by using the `set_flextable_defaults(fonts_ignore=TRUE)` command or use a
## compatible engine by defining `latex_engine: xelatex` in the YAML header of the
## R Markdown document.
```

player	events_pla	avg_par_score
Ricky Wysocki	12	-0.4605263
Eagle McMahon	12	-0.4269006
Calvin Heimburg	13	-0.4092141
Paul McBeth	13	-0.3956640
Chris Dickerson	8	-0.3577778
Kevin Jones	13	-0.3224932

Bounceback Rate

A statistic I've heard mentioned by commentary teams is bounceback rate, or the propensity for players to score birdie (a stroke under par for a hole) or better after bogeying a hole. However, to my knowledge this statistic has never been calculated, just speculated about. Let's change that.

The first thing we'll have to do is find the score relative to par that a player gets on the next hole they play for any particular hole. To do this, let's join our hole_scores dataset onto itself, while subtracting 1 from the hole number to get the prior hole played.

```
hole_scores_w_prior <- hole_scores %>%
  mutate(prior_hole = hole - 1) %>%
  select(p_player = player
         ,p_tournament_short = tournament_short
         ,p_round = round
         ,bounceback_par_score = par_score
         ,prior_hole)

hole_scores <- hole_scores %>%
  left_join(
    hole_scores_w_prior
    ,by = c("player" = "p_player"
            ,"tournament_short" = "p_tournament_short"
            ,"round" = "p_round"
            ,"hole" = "prior_hole")
  ) %>%
  mutate(bounceback_opp =
           case_when(
             (par_score > 0 & !is.na(bounceback_par_score)) ~ 1
             ,TRUE ~ 0
           )
        ) %>%
  mutate(bounceback_birds =
           case_when(
             (bounceback_opp == 1 & bounceback_par_score < 0) ~ 1
             ,TRUE ~ 0
              )
         )
```

Let's check our output using a single player's round as an example

```
hole_scores %>%
  filter(file == "DESMOINES1.xlsx"
         ,player == "Paul McBeth") %>%
  flextable() %>%
  theme_zebra()
```

```
## Warning: Warning: fonts used in `flextable` are ignored because the `pdflatex`
## engine is used and not `xelatex` or `lualatex`. You can avoid this warning
## by using the `set_flextable_defaults(fonts_ignore=TRUE)` command or use a
## compatible engine by defining `latex_engine: xelatex` in the YAML header of the
## R Markdown document.
```

hole	player	file	score_val	round	tournamer	tournamer	course	distance
1	Paul McBeth	DESMOINE	4	1	Des Moines Challenge	Des Moines	Pickard Park	795
2	Paul McBeth	DESMOINES1.xlsx	3	1	Des Moines Challenge	Des Moines	Pickard Park	753
3	Paul McBeth	DESMOINE	2	1	Des Moines Challenge	Des Moines	Pickard Park	320
4	Paul McBeth	DESMOINES1.xlsx	2	1	Des Moines Challenge	Des Moines	Pickard Park	490
5	Paul McBeth	DESMOINE	3	1	Des Moines Challenge	Des Moines	Pickard Park	330
6	Paul McBeth	DESMOINES1.xlsx	4	1	Des Moines Challenge	Des Moines	Pickard Park	633
7	Paul McBeth	DESMOINE	2	1	Des Moines Challenge	Des Moines	Pickard Park	345
8	Paul McBeth	DESMOINES1.xlsx	3	1	Des Moines Challenge	Des Moines	Pickard Park	540
9	Paul McBeth	DESMOINE	3	1	Des Moines Challenge	Des Moines	Pickard Park	290
10	Paul McBeth	DESMOINES1.xlsx	4	1	Des Moines Challenge	Des Moines	Pickard Park	844
11	Paul McBeth	DESMOINE	4	1	Des Moines Challenge	Des Moines	Pickard Park	810
12	Paul McBeth	DESMOINES1.xlsx	2	1	Des Moines Challenge	Des Moines	Pickard Park	260
13	Paul McBeth	DESMOINE	3	1	Des Moines Challenge	Des Moines	Pickard Park	290

hole	player	file	score_val	round	tournamer	tournamer	course	distance
14	Paul McBeth	DESMOINES1.xlsx	4	1	Des Moines Challenge	Des Moines	Pickard Park	606
15	Paul McBeth	DESMOINE	3	1	Des Moines Challenge	Des Moines	Pickard Park	600
16	Paul McBeth	DESMOINES1.xlsx	4	1	Des Moines Challenge	Des Moines	Pickard Park	315
17	Paul McBeth	DESMOINE	2	1	Des Moines Challenge	Des Moines	Pickard Park	235
18	Paul McBeth	DESMOINES1.xlsx	5	1	Des Moines Challenge	Des Moines	Pickard Park	1,080

Looks like our bounceback_par_score variable is correctly reporting the score relative to par that a player earned on the next hole for any particular round, bounceback_opp is counting instances where a bounceback birdie is possible, and bounceback_birds is showing if a birdie or better was achieved. Now let's see how the players' bounceback rate compares to their standard birdie rate. Additionally, this will be limited to the top 20 players by average score relative to par.

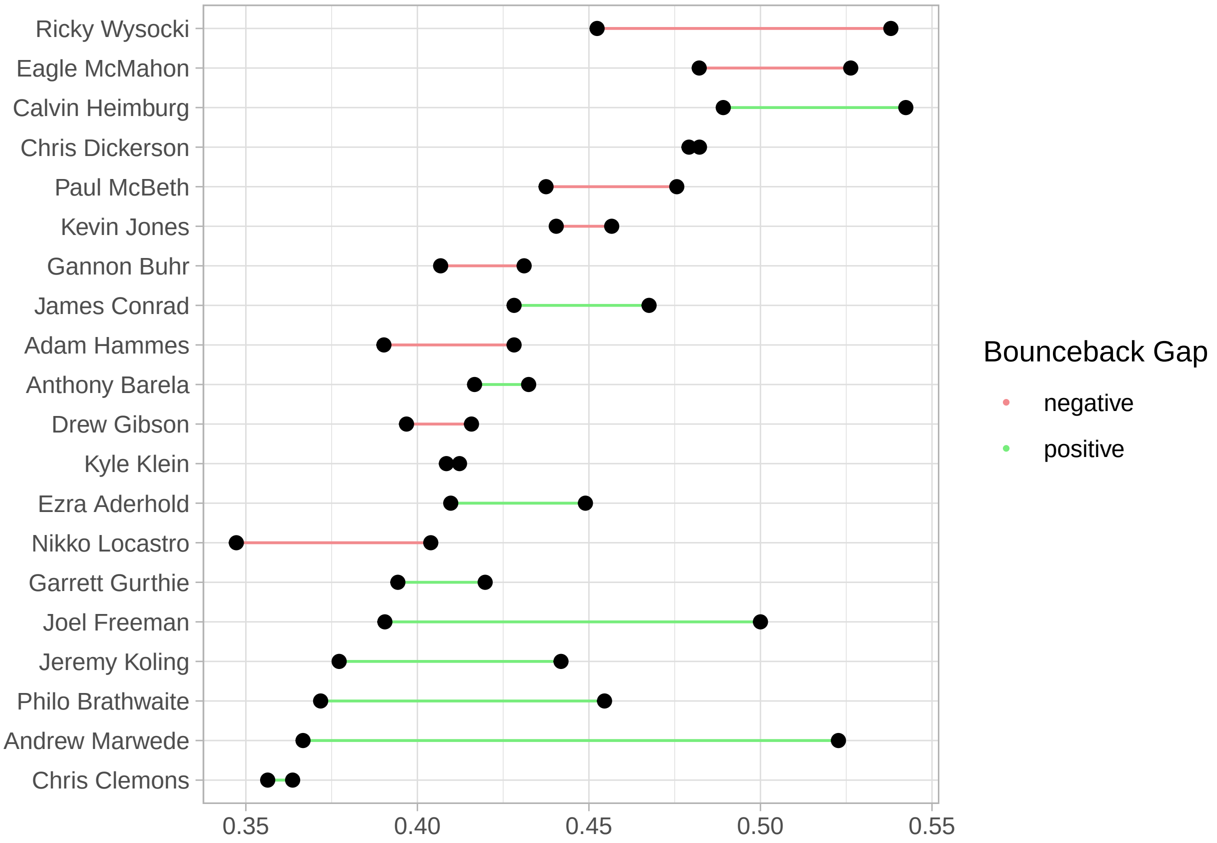
```
bounceback <- hole_scores %>%
  semi_join(top_50, by = "player") %>%
  group_by(player) %>%
  summarize(
    events_played = n_distinct(tournament_short)
    ,holes_played = n()
    ,total_score = sum(score_val)
    ,avg_score = mean(score_val)
    ,median_score = median(score_val)
    ,total_par_score = sum(par_score)
    ,avg_par_score = mean(par_score)
    ,median_par_score = median(par_score)
    ,bird_or_better = sum(if_else(par_score < 0,1,0))
    ,bird_rate = bird_or_better / holes_played
    ,bounceback_birds = sum(bounceback_birds, na.rm = T)
    ,bounceback_opp = sum(bounceback_opp, na.rm = T)
    ,bounceback_rate = bounceback_birds / bounceback_opp
    ,bounceback_gap = bounceback_rate - bird_rate
    ,chart_line_color = case_when(
      bounceback_gap > 0 ~ "positive"
      ,TRUE ~ "negative"
    )
  ) %>%
  top_n(20,wt = desc(avg_par_score))

bounceback %>%
  select(player,bounceback_birds,bounceback_opp,bounceback_rate,bird_rate,bounceback_gap) %>%
  arrange(desc(bounceback_gap)) %>%
  flextable() %>%
  theme_zebra()
```

```
## Warning: Warning: fonts used in ‘flextable‘ are ignored because the ‘pdflatex‘
## engine is used and not ‘xelatex‘ or ‘lualatex‘. You can avoid this warning
## by using the ‘set_flextable_defaults(fonts_ignore=TRUE)‘ command or use a
## compatible engine by defining ‘latex_engine: xelatex‘ in the YAML header of the
## R Markdown document.
```

player	bouncebac	bouncebac	bouncebac	bird_rate	bounceback_gap
Andrew Marwede	23	44	0.5227273	0.3666667	0.156060606
Joel Freeman	30	60	0.5000000	0.3905229	0.109477124
Philo Brathwaite	25	55	0.4545455	0.3717949	0.082750583
Jeremy Koling	38	86	0.4418605	0.3771930	0.064667483
Calvin Heimburg	32	59	0.5423729	0.4891599	0.053212990
James Conrad	36	77	0.4675325	0.4281843	0.039348186
Ezra Aderhold	44	98	0.4489796	0.4097222	0.039257370
Garrett Gurthie	34	81	0.4197531	0.3943089	0.025444143
Anthony Barela	32	74	0.4324324	0.4166667	0.015765766
Chris Clemons	28	77	0.3636364	0.3563686	0.007267800
Chris Dickerson	23	48	0.4791667	0.4822222	-0.003055556
Kyle Klein	29	71	0.4084507	0.4122807	-0.003829998
Kevin Jones	37	84	0.4404762	0.4566396	-0.016163376
Drew Gibson	25	63	0.3968254	0.4157706	-0.018945212
Gannon Buhr	24	59	0.4067797	0.4311111	-0.024331450
Adam Hammes	32	82	0.3902439	0.4281843	-0.037940379
Paul McBeth	21	48	0.4375000	0.4756098	-0.038109756
Eagle McMahon	27	56	0.4821429	0.5263158	-0.044172932
Nikko Locastro	25	72	0.3472222	0.4039039	-0.056681682
Ricky Wysocki	19	42	0.4523810	0.5380117	-0.085630744

The bounceback_gap number shows the difference between bounceback_rate and bird_rate, with a positive number indicating the player is more likely to birdie after carding a bogey or worse, while the opposite is true for a negative number. As shown above, 10 out of the 20 players score better than than they would be expected to, while 10 score worse. There doesn't seem to be any trend indicating the top players "turn it up a notch" after carding a bogey. Just for fun, let's plot what the players' bounceback_gap looks like.

```
bounceback %>%
  ggplot() +
    geom_dumbbell(
      aes(y=reorder(player,bird_rate), x=bird_rate, xend=bounceback_rate,color = chart_line_color)
      ,size_x = 2
      ,size_xend = 2
      ,colour_x = "black"
      ,colour_xend = "black"
    ) +
  scale_color_manual(values=c("#f28a8e", "#77ed7c"),name = "Bounceback Gap") +
  theme_light() +
  labs(x = NULL, y = NULL)
```



Green values indicate a bounceback_rate higher than a player's bird_rate and the opposite is true for red. Again, these values are scattered haphazardly, and the worst bounceback_gap score belongs to Ricky Wysocki, arguably the best player in the world. Bounceback birdies clearly aren't a necessary component of being an elite disc golfer.

Distnace Management

One thing almost everyone who follows disc golf likes seeing is the power throwers chucking bombs down the fairway. Players like Garrett Gurthie, Eagle McMahon, and Ricky Wysocki throw really far and it's fun to watch, but I'm interested in which players are the best at managing overall distance. To do this, let's correlate hole distance and total strokes for each player in the top 50. Additionally, we'll limit to the longer holes for each par value.

```
distance_management <- hole_scores %>%
  semi_join(top_50, by = "player") %>%
  group_by(player) %>%
  filter(
    (par == 3 & distance >= 400)
    | (par == 4 & distance >= 700)
    | (par == 5 & distance >= 1000)
  ) %>%
  summarise(
    dist_cor = cor(x = score_val, y = distance, method = "pearson")
  ) %>%
  arrange(dist_cor)

distance_management %>%
  top_n(10,wt = desc(dist_cor)) %>%
  flextable() %>%
  theme_zebra()
```

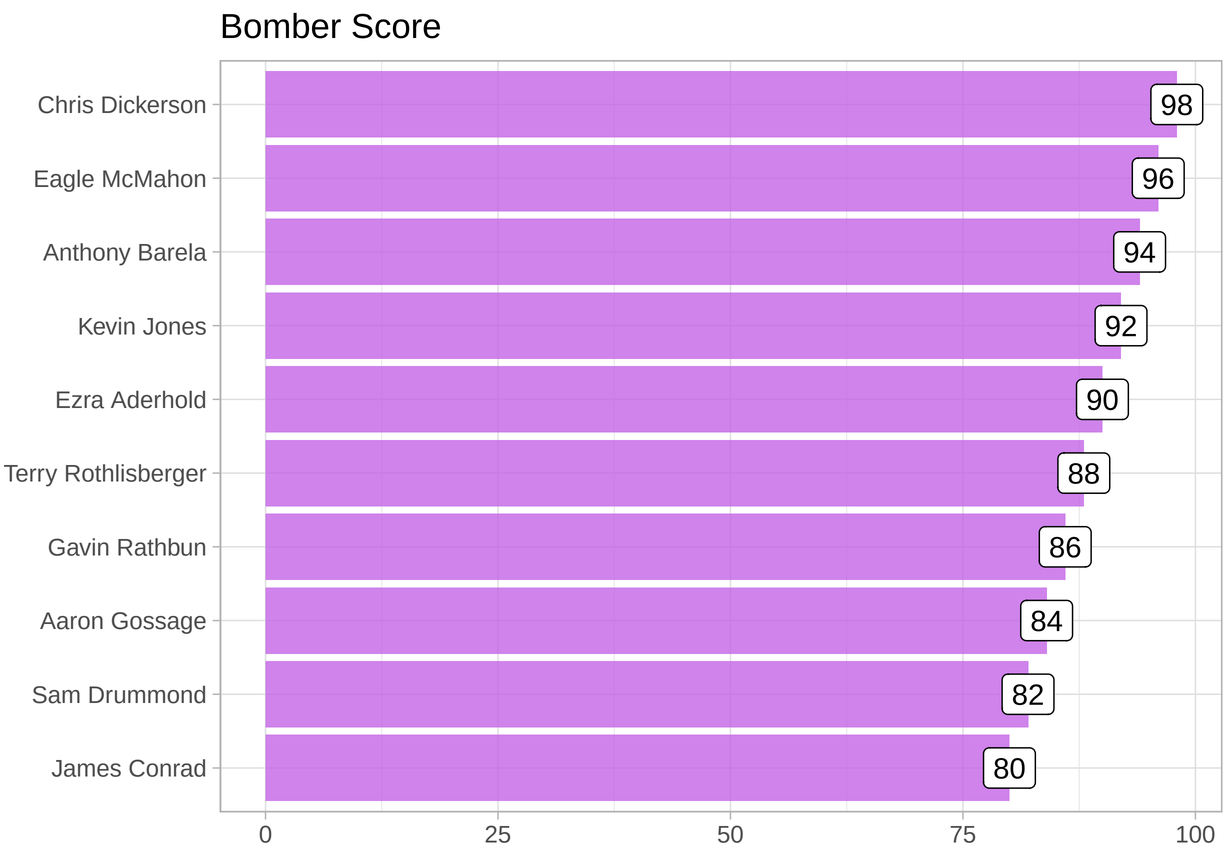
```
## Warning: Warning: fonts used in `flextable` are ignored because the `pdflatex`
## engine is used and not `xelatex` or `lualatex`. You can avoid this warning
## by using the `set_flextable_defaults(fonts_ignore=TRUE)` command or use a
## compatible engine by defining `latex_engine: xelatex` in the YAML header of the
## R Markdown document.
```

player	dist_cor
Chris Dickerson	0.5334475
Eagle McMahon	0.5752453
Anthony Barela	0.5779747
Kevin Jones	0.6079314
Ezra Aderhold	0.6098454
Terry Rothlisberger	0.6142670
Gavin Rathbun	0.6182921

player	dist_cor
Aaron Gossage	0.6224291
Sam Drummond	0.6252791
James Conrad	0.6312405

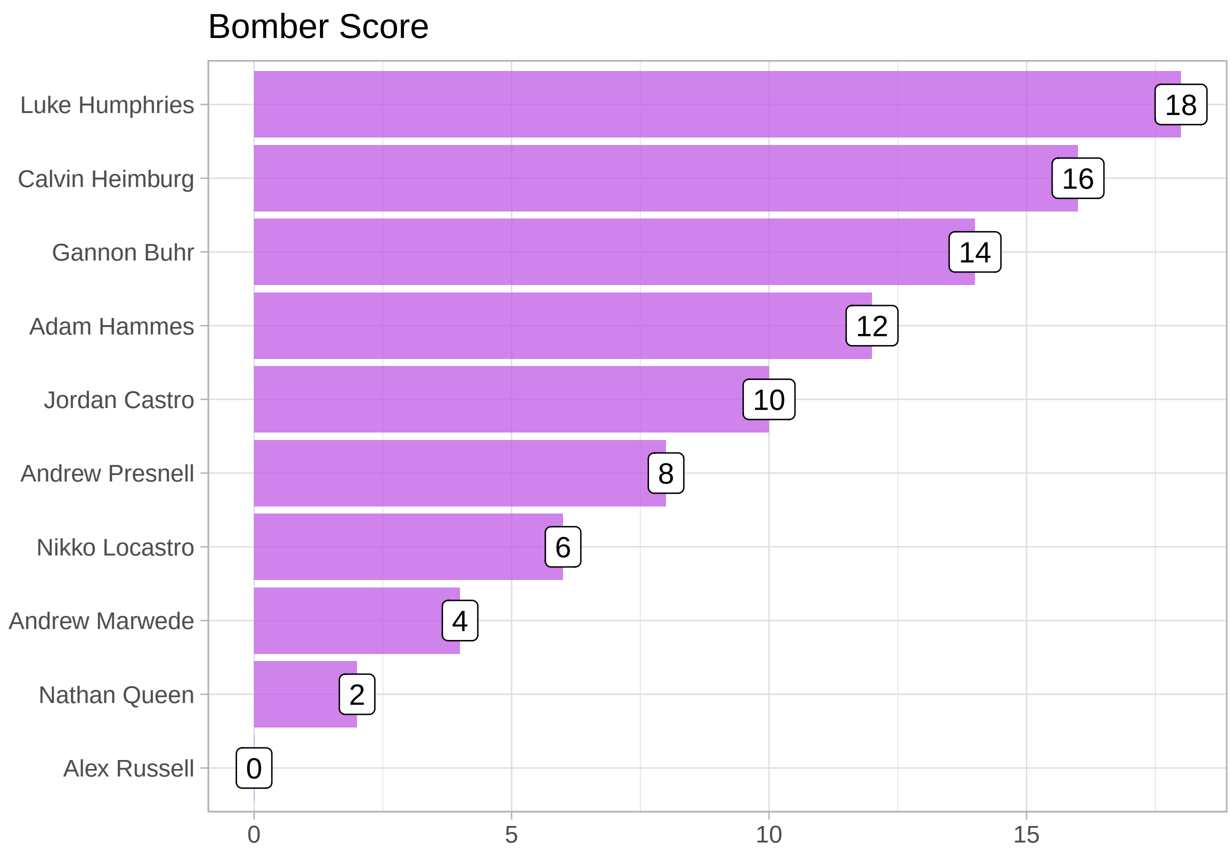
We've got the 10 players with the lowest correlation of score to distance above, and many of the names are not surprising, like McMahon, Brela, Jones, and Aderhold. However, it is pretty surprising to see Chris Dickerson take the top spot, as he isn't really known for being an elite distance thrower. Distance management takes more skills however, like having a good forehand and backhand, throwing accurate rollers, and making smart shot selections. While it's the correlation is helpful, we can turn it into something a little more fun, a bomber score ((1 - dist_cor) * 100).

```
distance_management <- distance_management %>%
  mutate(
    Percentile_Rank= rank(dist_cor)/length(dist_cor)
    ,Bomber_Score = (1 - Percentile_Rank) * 100
    ,round_Bomber_Score = round(Bomber_Score,2)
  )

distance_management %>%
  top_n(10,wt = round_Bomber_Score) %>%
  ggplot() +
  geom_bar(mapping = aes(x = reorder(player,round_Bomber_Score), y = round_Bomber_Score), stat = "identi
  geom_label(mapping = aes(x = player, y = round_Bomber_Score, label = round_Bomber_Score)) +
  coord_flip() +
  theme_light() +
  ggtitle("Bomber Score") +
  labs(x = NULL, y = NULL)
```



The above shows this “Bomber Score” that rates how well the players manage distance. Let’s take a look at the bottom 10 players (of the top 50 of course).

```
distance_management %>%
  top_n(10,wt = desc(round_Bomber_Score)) %>%
  ggplot() +
  geom_bar(mapping = aes(x = reorder(player,round_Bomber_Score), y = round_Bomber_Score), stat = "ident
  geom_label(mapping = aes(x = player, y = round_Bomber_Score, label = round_Bomber_Score)) +
  coord_flip() +
  theme_light() +
  ggtitle("Bomber Score") +
  labs(x = NULL, y = NULL)
```



Again, a lot of the players seen make sense. Players like Russell, Queen, Marwede, and Presnell are not known for throwing very far. The biggest surprise on this list is Heimburg, who has elite level distance on his backhand. His placement on this list is likely due to not being able to rely as strongly on his forehand for distance as some of the other players better at managing total distance.