Final Project

```
years <- 2002:2019
```

```
matches <- readRDS("matches.rds") %>%
  mutate(week = factor(week, c("0", "1", "2", "3", "4", "5", "6", "7", "dcmpd", "dcmp", "cmpd", "cmpf",

matches$week[matches$event_type == "district_championship_division"] <- "dcmpd"
matches$week[matches$event_type == "district_championship"] <- "dcmp"
matches$week[matches$event_type == "championship_division"] <- "cmpd"
matches$week[matches$event_type == "championship_finals"] <- "cmpf"
matches$week[matches$event_type == "festival_of_champions"] <- "foc"
```

```
teams <- readRDS("teams.rds") %>%
  filter(rookie_year != 2020) %>%
  rowwise() %>%
  mutate(years_competed = length(years))
```

```
scores <- matches %>%
  pivot_longer(cols = c("red_score", "blue_score"), names_to = "alliance", values_to = "score") %>%
  mutate(alliance = as.factor(alliance))
scores$alliance <- recode_factor(scores$alliance,
                                 "red_score" = "red",
                                 "blue_score" = "blue")
```

```
matches_by_team <- scores %>%
  pivot_longer(cols = c("red_alliance_1", "red_alliance_2",
                        "red_alliance_3", "red_alliance_4",
                        "blue_alliance_1", "blue_alliance_2",
                        "blue_alliance_3", "blue_alliance_4"),
               names_to = "position",
               values_to = "team") %>%
  filter(!is.na(team)) %>%
  separate(position, sep = "_alliance_", into = c("team_alliance", "position")) %>%
  filter(alliance == team_alliance) %>%
  select(!team_alliance) %>%
  mutate(team_num = as.integer(str_remove(team, "frc")))
```

```
team_matches_per_year <- matches_by_team %>%
  group_by(team, team_num, year) %>%
  summarize(played = n()) %>%
  ungroup()
```

```
team_events_per_year <- matches_by_team %>%
  group_by(team, team_num, year, event_key, event_type) %>%
  summarize() %>%
```

```
  ungroup() %>%
  group_by(team, team_num, year) %>%
  summarize(events = n(),
            non_cmp_events = length(event_key[event_type %in% c("regional",
                                                                "district")]),
            attended_dcmp = "district_championship" %in% event_type |
              "district_championship_division" %in% event_type,
            attended_cmp = "championship_division" %in% event_type,
            in_district = "district" %in% event_type)
```

```
team_avg_by_year <- matches_by_team %>%
  group_by(team, team_num, year) %>%
  summarize(avg_score = mean(score))
```

```
team_percentile_by_year <- data.frame(
  team = character(),
  team_num = integer(),
  year = integer(),
  avg_score = double(),
  percentile = double()
)

for (yr in years) {
  x <- team_avg_by_year %>%
    filter(year == yr)
  x$percentile <- ecdf(x$avg_score)(x$avg_score) * 100

  team_percentile_by_year <- bind_rows(team_percentile_by_year, x) %>%
    arrange(team_num)
}
```

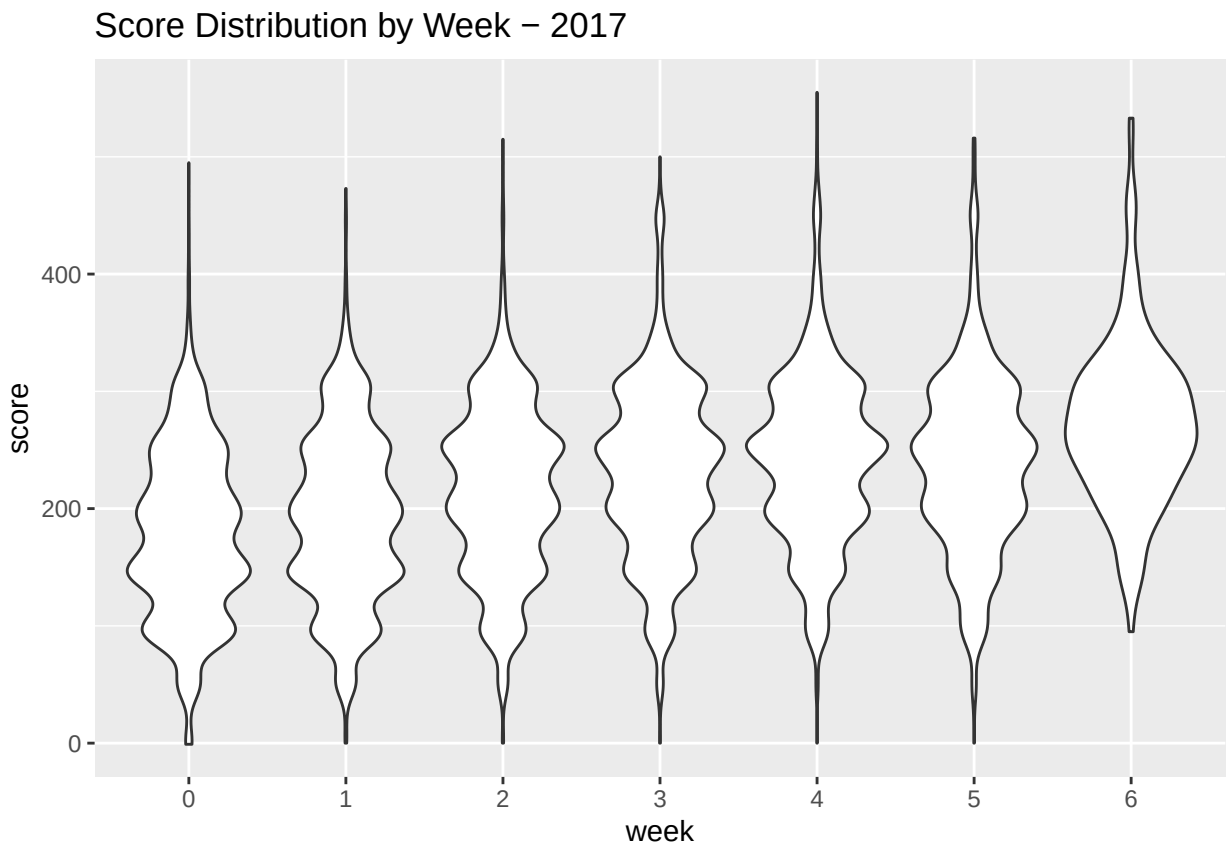
```
team_percentile_avg <- team_percentile_by_year %>%
  group_by(team, team_num) %>%
  summarize(avg_percentile = mean(percentile)) %>%
  ungroup()
```

Performance Throughout a Season

To evaluate performance within a season, we can compare each week of competition. Because the season is multiple weeks long, events happening on the same weekend are classified as being part of that "week" of competition.

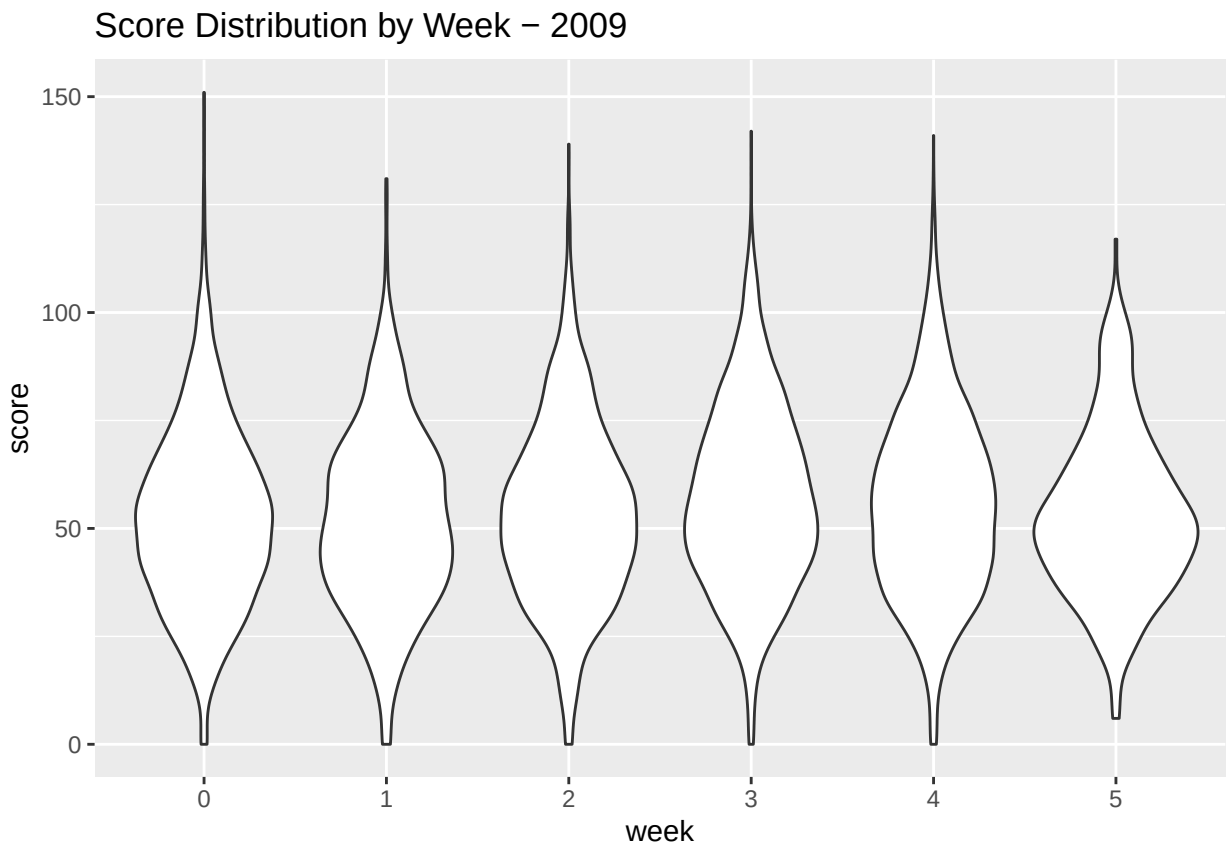
In some years, performance does seem to improve, but in others it does not. For example, in 2017 the performance increased slightly as the weeks of the season progressed:

```
scores %>%
    filter(year == 2017 & week %in% c("0", "1", "2", "3", "4", "5", "6")) %>%
    ggplot() +
    geom_violin(aes(week, score)) +
    labs(title = "Score Distribution by Week - 2017")
```



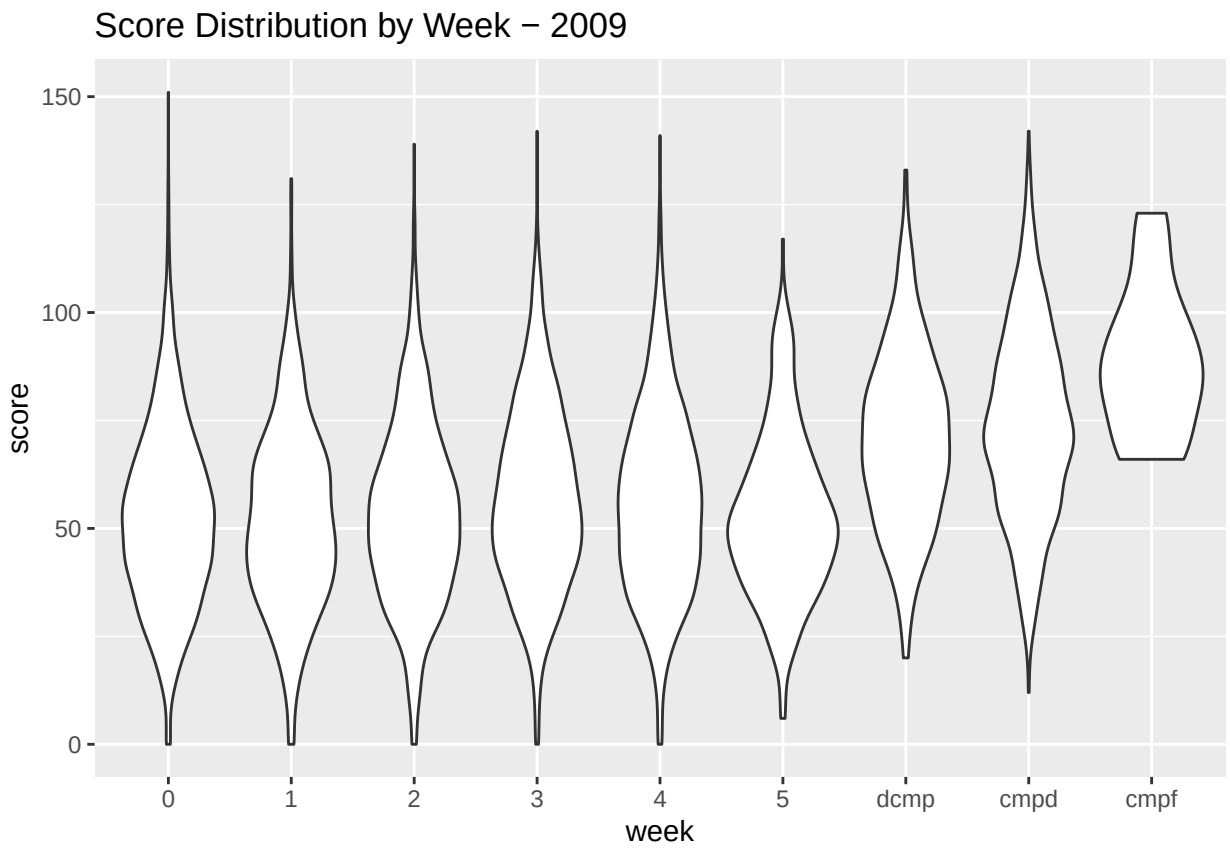
But in the 2009 season, there isn't any obvious improvement:

```
scores %>%
    filter(year == 2009 & week %in% c("0", "1", "2", "3", "4", "5")) %>%
    ggplot() +
    geom_violin(aes(week, score)) +
    labs(title = "Score Distribution by Week - 2009")
```



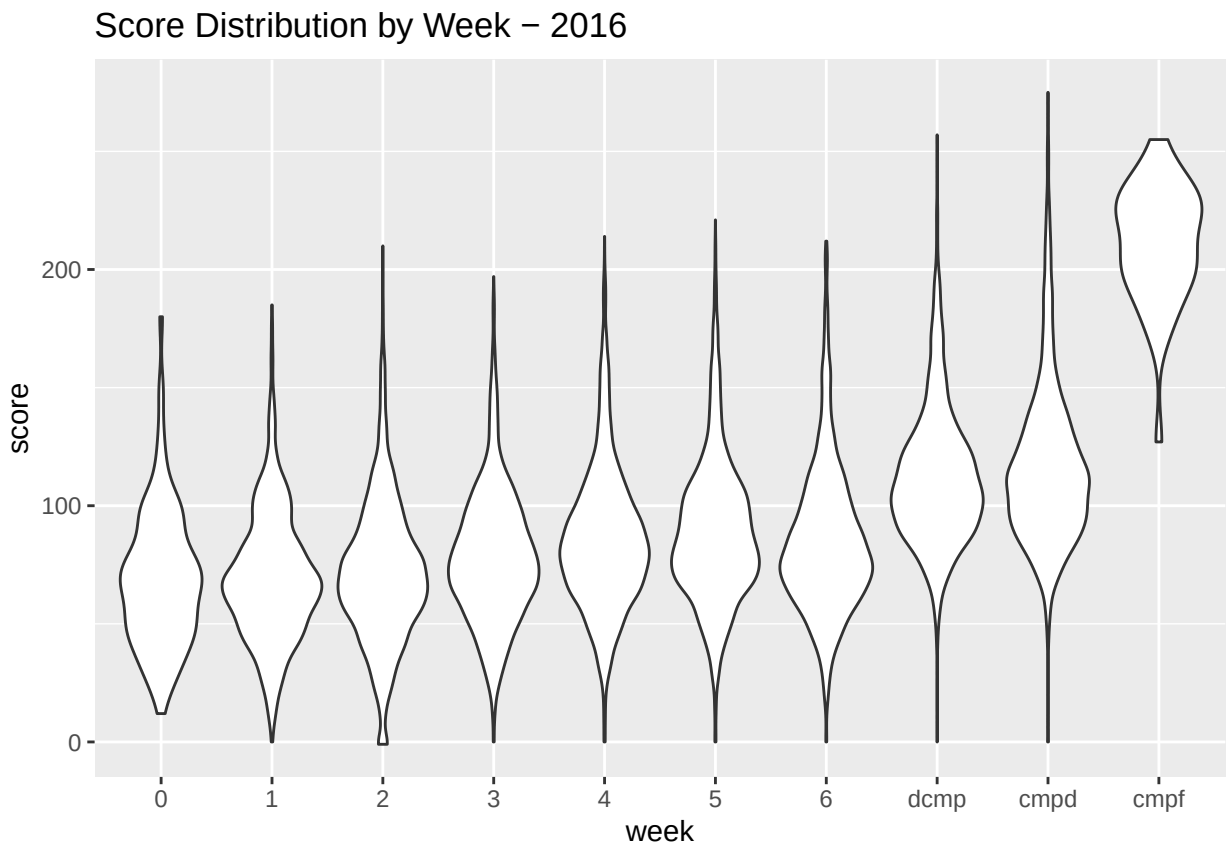
If we include the championship events, we consistently see an improvement over the weeks prior. Using 2009 again, we see that District Championships (`dcmp`), Championship Divisions (`cmpd`) and Championship Finals (`cmpf`) have a very noticeable improvement in scores.

```
scores %>%
    filter(year == 2009) %>%
    ggplot() +
    geom_violin(aes(week, score)) +
    labs(title = "Score Distribution by Week - 2009")
```



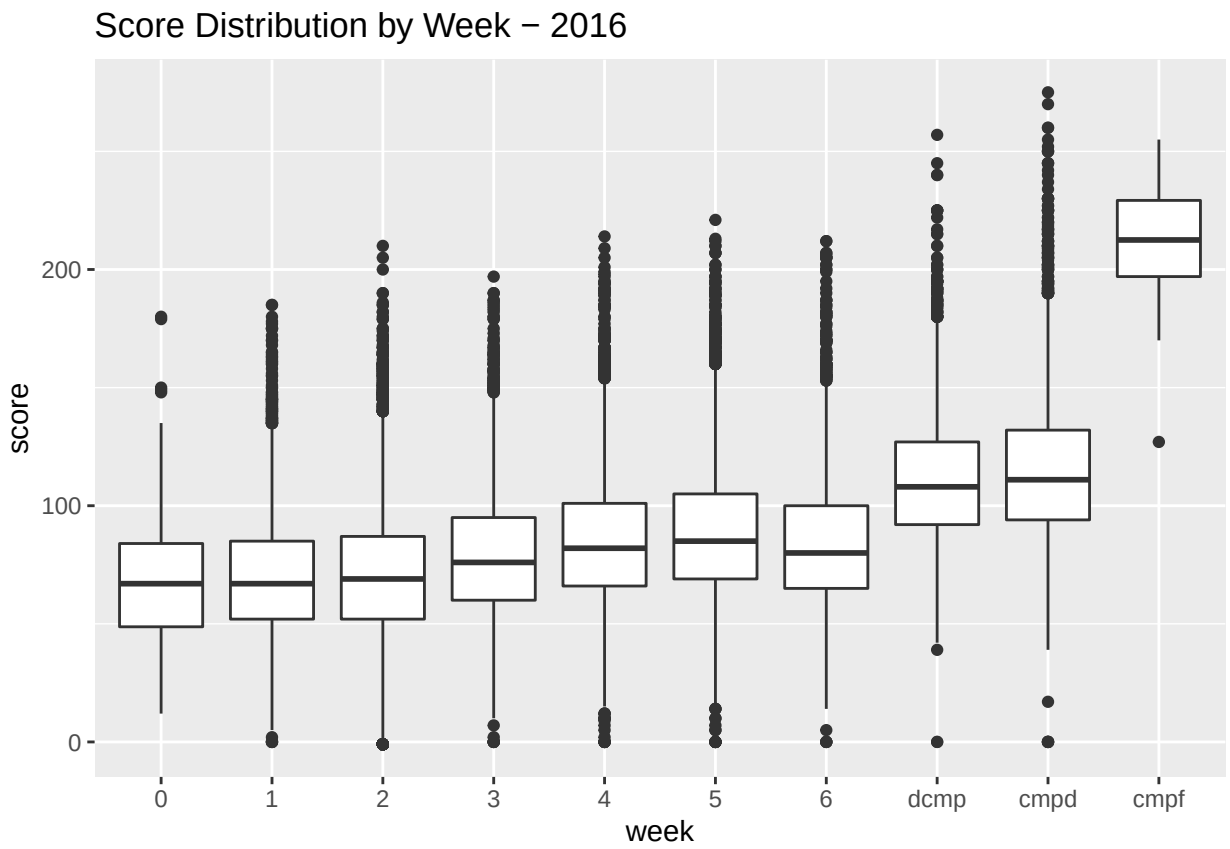
In many years, the difference between the Championship Finals and other levels can be drastic. Take the 2016 season as an example. The weeks progressed with minor improvements, the District Championships and Championship Divisions have an improvement in scores, but the Championship Finals have scores much higher.

```
scores %>%
    filter(year == 2016) %>%
    ggplot() +
    geom_violin(aes(week, score)) +
    labs(title = "Score Distribution by Week - 2016")
```



Using a boxplot, we can see that the average score in a Championship Finals match is almost double that of the average score in the Championship Divisions.

```
scores %>%
    filter(year == 2016) %>%
    ggplot() +
    geom_boxplot(aes(week, score)) +
    labs(title = "Score Distribution by Week - 2016")
```

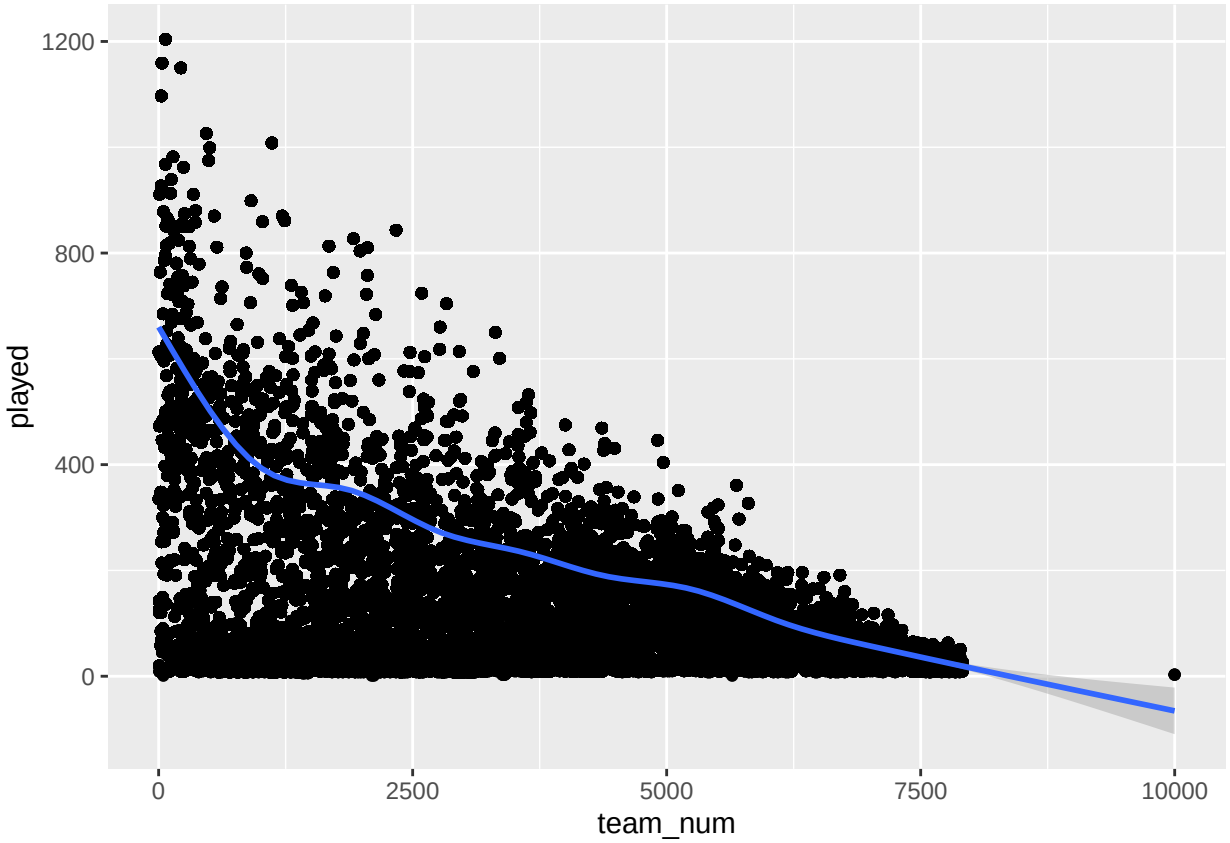


Matches Played by Team (since 2002)

This chart shows how many matches each team has played across the years 2002 to 2019. As expected, teams with lower numbers have played more matches (because they've been around longer). But, teams don't last forever, which is why many teams have very few matches played.

One team to notice is team 9999. This team does not actually exist. The number 9999 is used as a placeholder for a team that has not received a number yet. This usually only happens during preseason and offseason events, though it seems to have happened in a week 0 regional event in 2004.

```
matches_by_team %>%
  group_by(team) %>%
  summarize(team_num = team_num, played = n()) %>%
  ungroup() %>%
  ggplot(aes(team_num, played)) +
  geom_point() +
  geom_smooth()
```



Comparing Performance Across Years

Because the competition changes every year, comparing raw scores from one year to the next is not a good measure of performance. For example, the 2010 competition had an average match score of 4.07, while the 2018 competition had an average match score of 291.90.

```
scores %>%
  group_by(year) %>%
  summarize(avg_score = mean(score)) %>%
  knitr::kable(col.names = c("Year", "Average Score"),
        align = "ll")
```

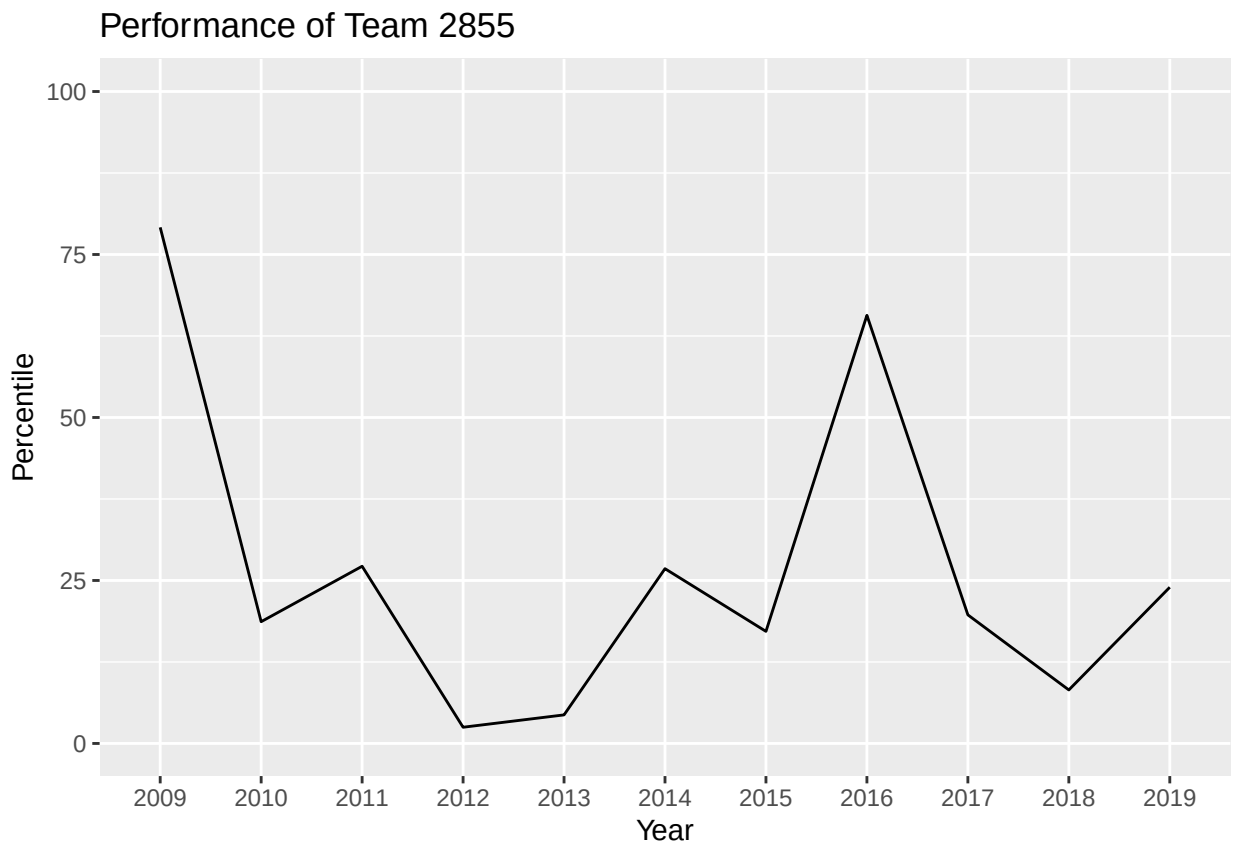
Year	Average Score
2002	29.921029
2003	48.985660
2004	59.851016
2005	28.246132
2006	35.213372
2007	30.064372
2008	42.879916
2009	56.692035
2010	4.079307
2011	38.120657
2012	25.047814

Year	Average Score
2013	64.813903
2014	100.211908
2015	75.583587
2016	85.289039
2017	233.390947
2018	291.902244
2019	54.930552

We found that a good way to compare across years is to compute a team's performance relative to the rest of the teams in the competition that year. By calculating the average score achieved by each team, we can determine what percentile each team achieved in a given year.

For example, the performance of Team 2855 (Max's former team) is shown below:

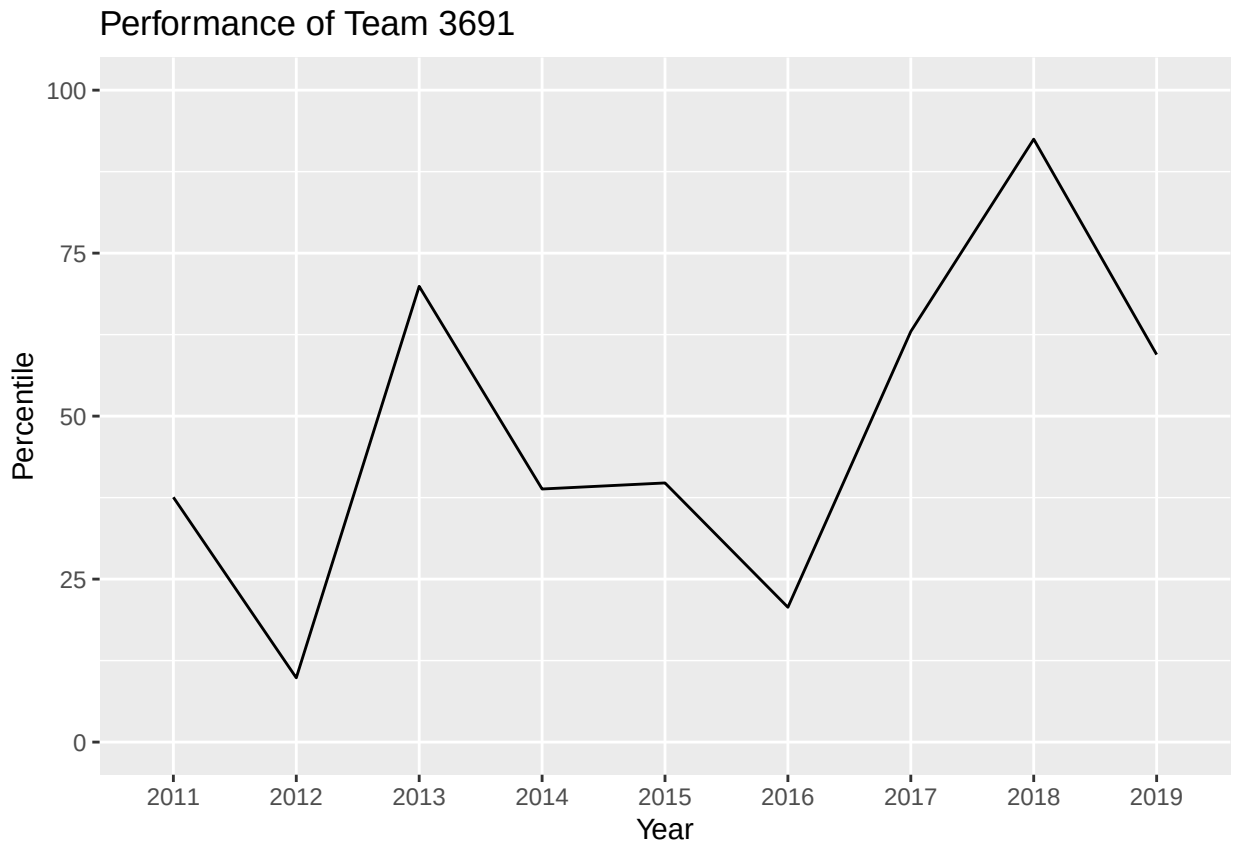
```
team_percentile_by_year %>%
  filter(team_num == 2855) %>%
  ggplot(aes(as.factor(year), percentile, group = 1)) +
  geom_line() +
  ylim(0, 100) +
  labs(title = "Performance of Team 2855",
       x = "Year",
       y = "Percentile")
```



As shown here, Team 2855 has had a couple of good years, but overall has been an ok team at best.

We can compare this to Team 3691, based out of Northfield High School:

```
team_percentile_by_year %>%
  filter(team_num == 3691) %>%
  ggplot(aes(as.factor(year), percentile, group = 1)) +
  geom_line() +
  ylim(0, 100) +
  labs(title = "Performance of Team 3691",
       x = "Year",
       y = "Percentile")
```



At first glance, it seems like Team 3691 has been a better-performing team than Team 2855. We can average all of a team's percentiles to confirm this.

```
team_percentile_avg %>%
  filter(team_num %in% c(2855, 3691)) %>%
  left_join(teams, by = c("team_num" = "team_number")) %>%
  arrange(team_num) %>%
  select(team_num, nickname, rookie_year, years_competed, avg_percentile) %>%
  head(20) %>%
  knitr::kable(col.names = c("Team #", "Nickname", "Rookie Year",
                             "Years Competed", "Average Percentile"),
               align = "lllll")
```

Team #	Nickname	Rookie Year	Years Competed	Average Percentile
2855	BEASTBOT	2009	11	26.67849
3691	RoboRaiders	2011	10	47.94250

This confirms that Team 3691 is a better-performing team than Team 2855.

```
team_nums <- c(16, 33, 111, 118, 148, 254, 330, 900, 1114, 1678, 1816, 2052, 2056, 2220, 2846, 2855, 313
for (i in 1:length(team_nums)) {
  print(team_percentile_by_year %>%
          filter(team_num == team_nums[i]) %>%
          ggplot(aes(as.factor(year), percentile, group = 1)) +
          geom_point() +
          geom_line() +
          ylim(0, 100) +
          labs(title = str_c("Team ", as.character(team_nums[i])),
               x = "Year",
               y = "Percentile"))
}
```

Top Teams by Average Percentile

We can use this metric to determine the top teams of all time:

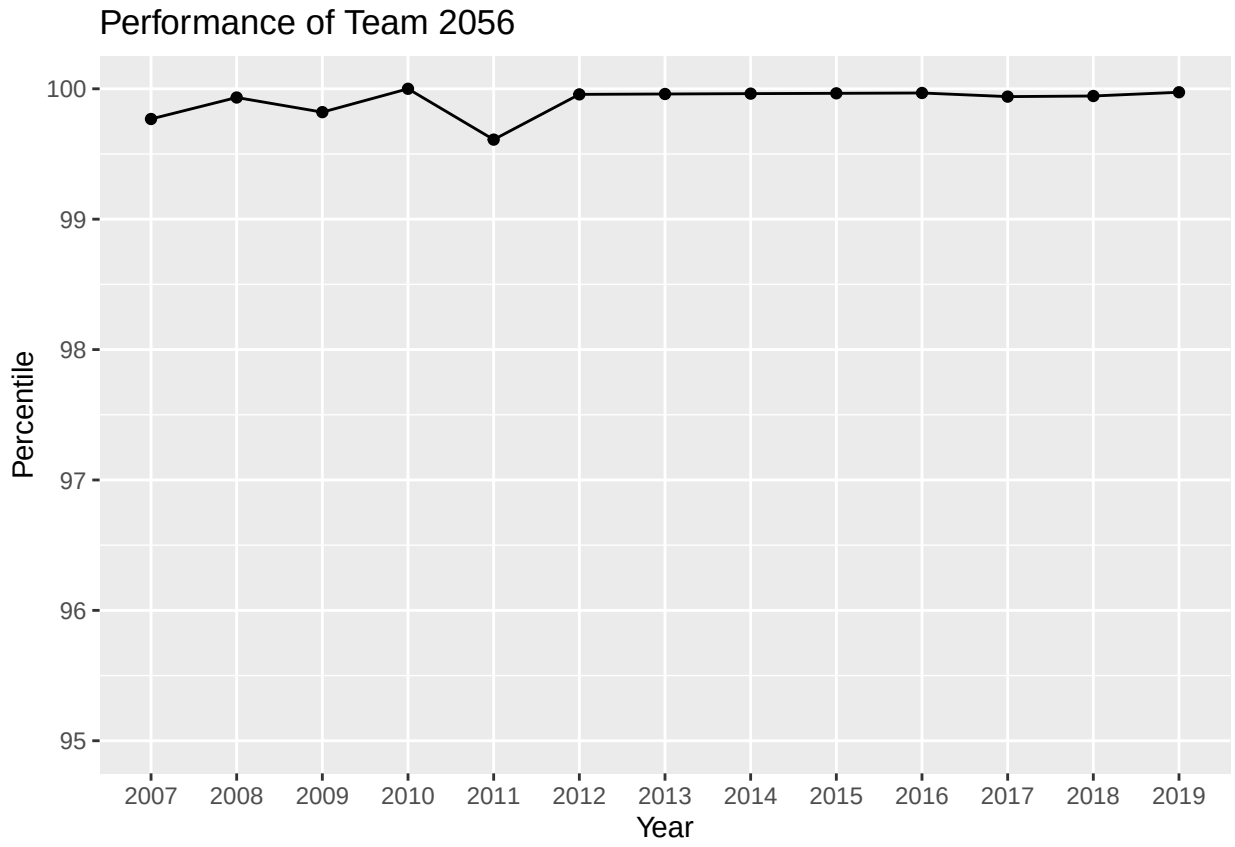
```
team_percentile_avg %>%
  left_join(teams, by = c("team_num" = "team_number")) %>%
  arrange(desc(avg_percentile)) %>%
  select(team_num, nickname, rookie_year, years_competed, state_prov, avg_percentile) %>%
  head(20) %>%
  knitr::kable(col.names = c("Team #", "Nickname", "Rookie Year",
                             "Years Competed", "State/Province", "Average Percentile"),
               align = "llllll")
```

Team #	Nickname	Rookie Year	Years Competed	State/Province	Average Percentile
2056	OP Robotics	2007	14	Ontario	99.90801
2970	eSchool eBots	2009	1	WI	98.86567
5406	Celt-X	2015	6	Ontario	98.64966
2098	Bulldogs	2007	1	GA	98.45560
7457	suPURDUEper Robotics	2019	2	Indiana	97.89894
254	The Cheesy Poofs	1999	22	California	97.10775
1114	Simbotics	2003	18	Ontario	96.99516
3683	Team DAVE	2011	10	Ontario	96.98346
67	The HOT Team	1997	24	Michigan	96.91468
4414	HighTide	2012	2	California	96.80851
2753	Team Overdrive	2009	2	NJ	96.71322
7553	OSTC - SWEET BOTS	2019	2	Michigan	96.67553
5184	TITANICS	2014	1	Alberta	96.55172
4678	CyberCavs	2013	8	Ontario	96.19824
782	Kilowatts	2002	3	CT	96.15072
5172	Gators	2014	7	Minnesota	95.95098
71	Team Hammond	1996	25	Indiana	95.85849

Team #	Nickname	Rookie Year	Years Competed	State/Province	Average Percentile
7021	TC Robotics	2018	3	Wisconsin	95.73704
4917	Sir Lancerbot	2014	7	Ontario	95.12320
27	Team RUSH	1997	24	Michigan	95.11682

A team that stands out here is Team 2056 (aptly named "OP Robotics"), who has been able to perform at an impressive level in all 14 years they've competed, giving them an average percentile of 99.90%. (Notice that the graph only shows the 95th to 100th percentiles, and yet they're still at the top.)

```
team_percentile_by_year %>%
  filter(team_num == 2056) %>%
  ggplot(aes(as.factor(year), percentile, group = 1)) +
  geom_point() +
  geom_line() +
  ylim(95, 100) +
  labs(title = "Performance of Team 2056",
       x = "Year",
       y = "Percentile")
```



Other teams that stand out are teams 2970 and 2098, which only competed for one season but were one of the best teams in that season.

Longest Competing Teams

We can use this metric to compute the performance of some of the oldest teams:

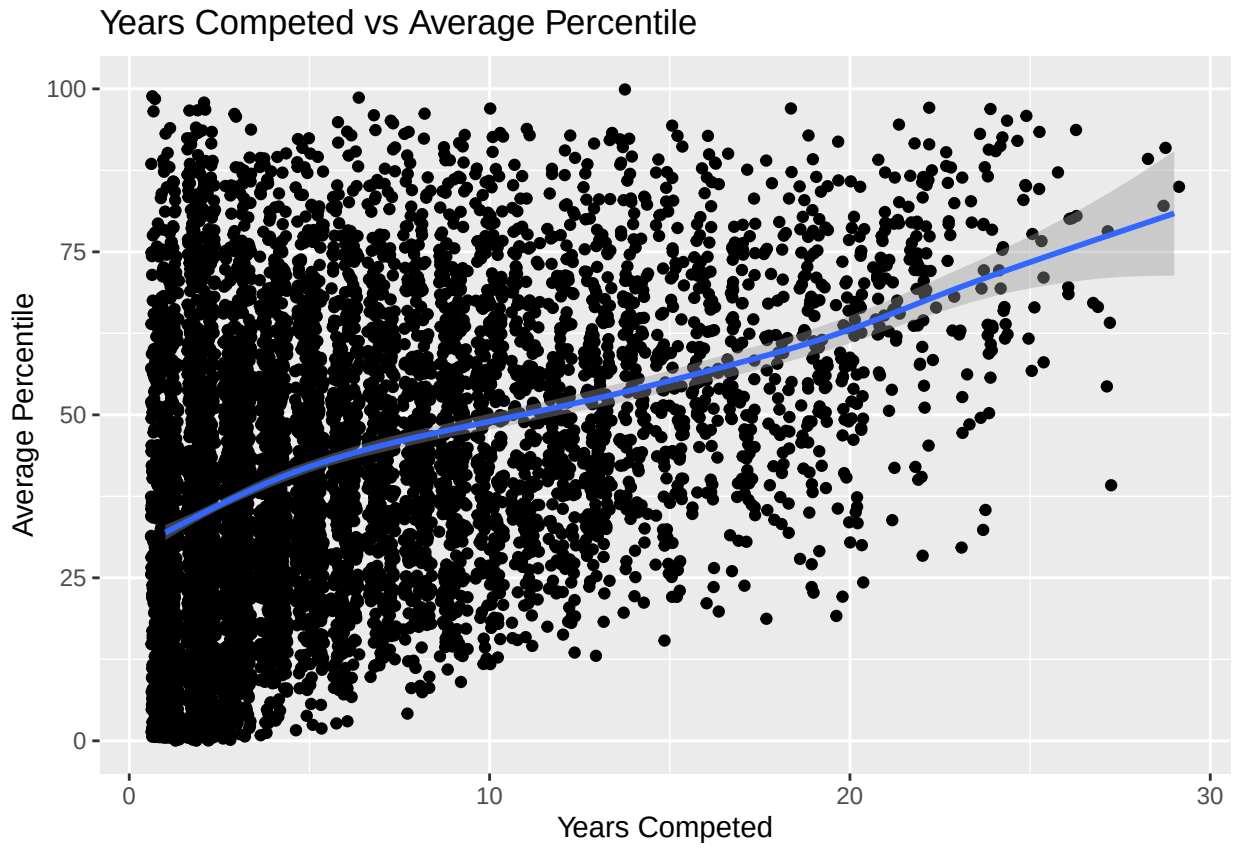
```
team_percentile_avg %>%
  left_join(teams, by = c("team_num" = "team_number")) %>%
  arrange(desc(years_competed), team_num) %>%
  select(team_num, nickname, rookie_year, years_competed, state_prov, avg_percentile) %>%
  head(30) %>%
  mutate(nickname = ifelse(team_num == 173, "RAGE Robotics", nickname)) %>%
  knitr::kable(col.names = c("Team #", "Nickname", "Rookie Year",
                             "Years Competed", "State/Province", "Average Percentile"),
               align = "llllll")
```

Team #	Nickname	Rookie Year	Years Competed	State/Province	Average Percentile
45	TechnoKats Robotics Team	1992	29	Indiana	82.05897
126	Gael Force	1992	29	Massachusetts	90.95443
191	X-CATS	1992	29	New York	84.97691
148	Robowranglers	1992	28	Texas	89.25748
81	MetalHeads	1994	27	Illinois	39.20181
131	C.H.A.O.S.	1992	27	New Hampshire	67.16112
151	Tough Techs	1992	27	New Hampshire	54.35748
155	The Technonuts	1994	27	Connecticut	66.55131
157	AZTECHS	1992	27	Massachusetts	64.12055
190	Gompei and the H.E.R.D.	1992	27	Massachusetts	78.16108
74	Team CHAOS	1995	26	Michigan	80.50757
108	SigmaC@T Robotics Team	1995	26	Florida	68.52366
111	WildStang	1996	26	Illinois	93.69707
141	WOBOT	1995	26	Michigan	80.07415
166	Chop Shop	1995	26	New Hampshire	69.57478
173	RAGE Robotics	1995	26	Connecticut	80.15928
177	Bobcat Robotics	1995	26	Connecticut	87.18504
8	Paly Robotics	1996	25	California	56.74377
28	Pierson Whalers	1996	25	New York	66.48026
33	Killer Bees	1996	25	Michigan	93.39666
58	The Riot Crew	1996	25	Maine	85.25065
69	HYPER	1998	25	Massachusetts	82.95772
71	Team Hammond	1996	25	Indiana	95.85849
85	B.O.B. (Built on Brains)	1996	25	Michigan	85.06791
88	TJ²	1996	25	Massachusetts	77.73869
116	Epsilon Delta	1996	25	Virginia	58.06107
120	Cleveland's Team	1995	25	Ohio	61.69404
121	Rhode Warriors	1996	25	Rhode Island	76.64927
171	Cheese Curd Herd	1995	25	Wisconsin	71.05626
175	Buzz Robotics	1996	25	Connecticut	92.02480

Number of Years Competed vs Average Percentile

```
team_percentile_avg %>%
  left_join(teams, by = c("team_num" = "team_number")) %>%
  ggplot(aes(years_competed, avg_percentile)) +
  geom_jitter(height = 0) +
  geom_smooth() +
  ylim(0, 100) +
```

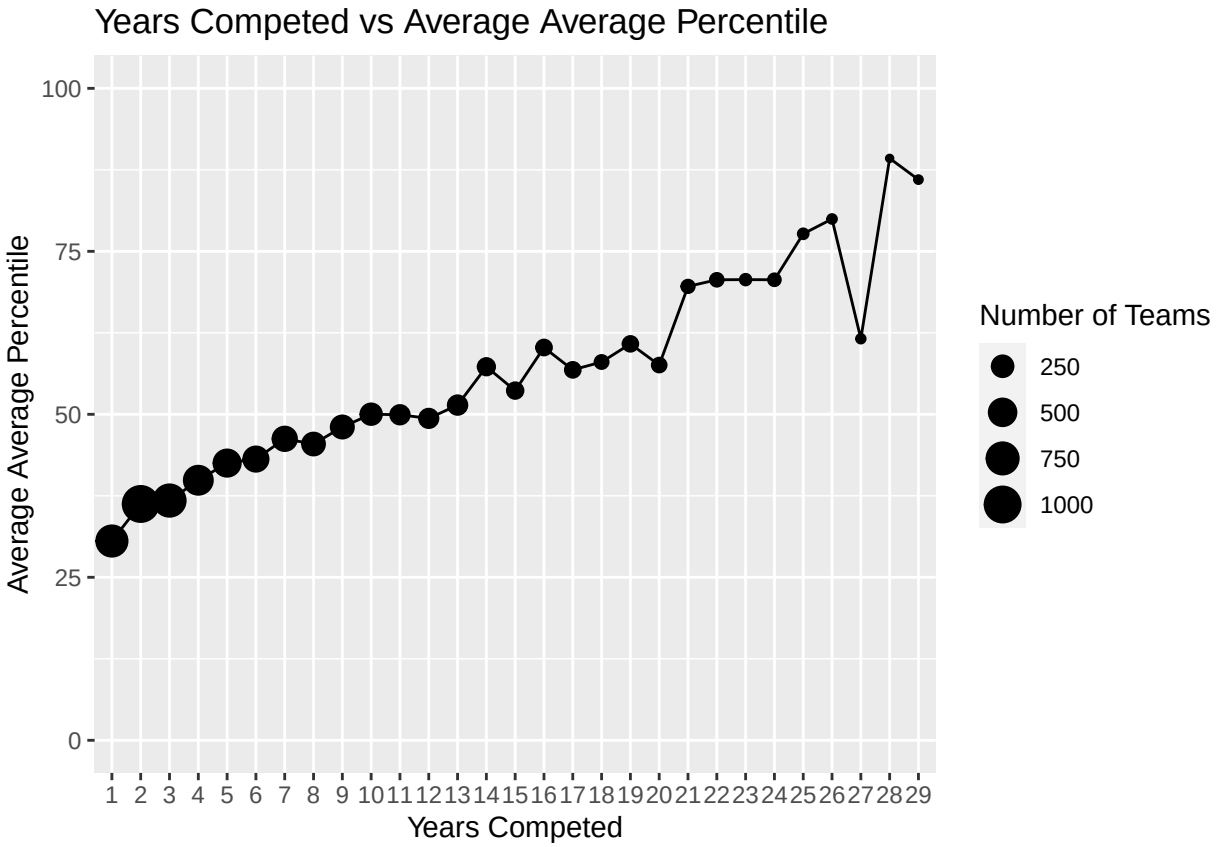
```
  labs(title = "Years Competed vs Average Percentile",
       x = "Years Competed",
       y = "Average Percentile")
```



Numbers of Years Competed vs Average Percentile

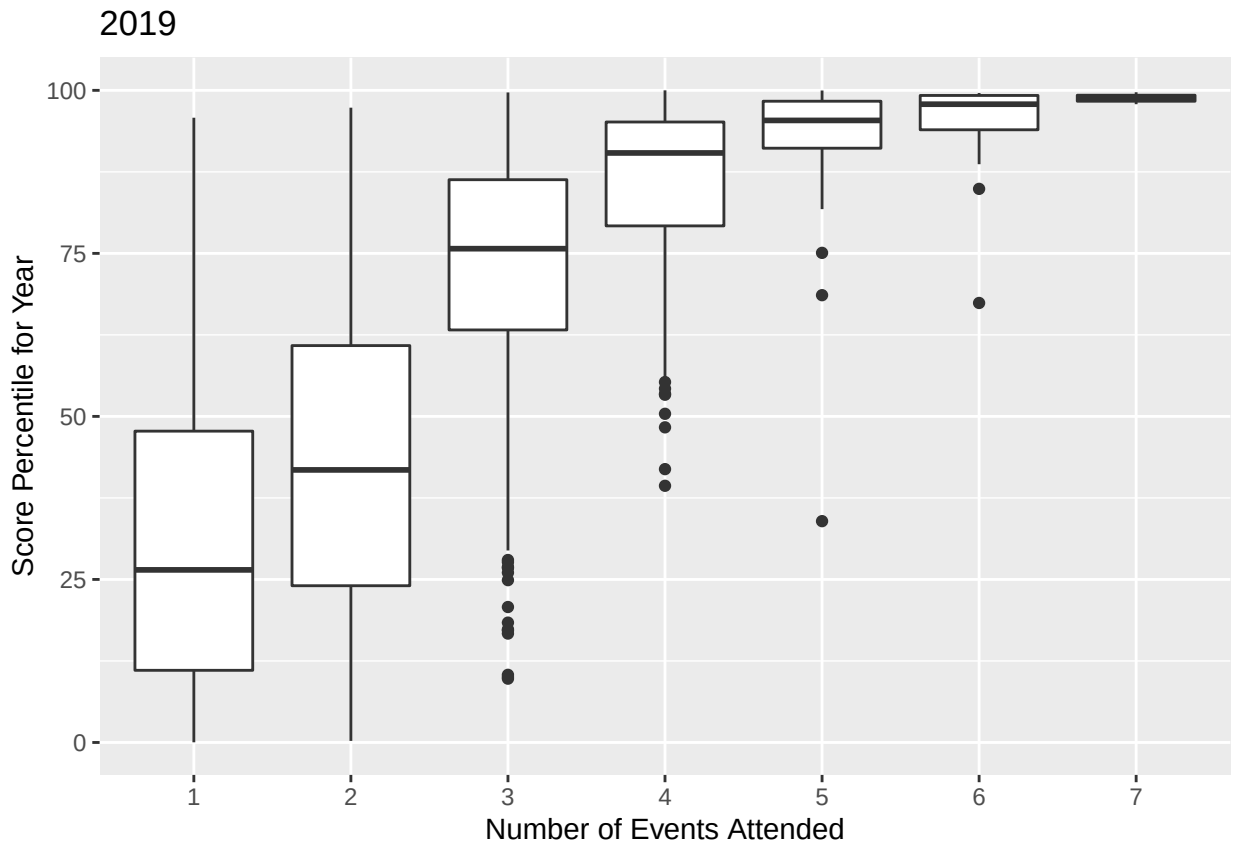
When we average the average percentiles for the teams in each group based on years competed, we see noticeable improvement as the teams compete longer. (Note that there are very few teams that have competed for over two decades, for example there is only one team that has competed for 28 years.)

```
team_percentile_avg %>%
  left_join(teams, by = c("team_num" = "team_number")) %>%
  filter(!is.na(years_competed)) %>%
  group_by(years_competed) %>%
  summarize(avg_avg_percentile = mean(avg_percentile), num_teams = n()) %>%
  ggplot() +
  geom_point(aes(as.factor(years_competed), avg_avg_percentile, size = num_teams)) +
  geom_line(aes(as.factor(years_competed), avg_avg_percentile, group = 1)) +
  ylim(0, 100) +
  labs(title = "Years Competed vs Average Average Percentile",
       x = "Years Competed",
       y = "Average Average Percentile",
       size = "Number of Teams")
```



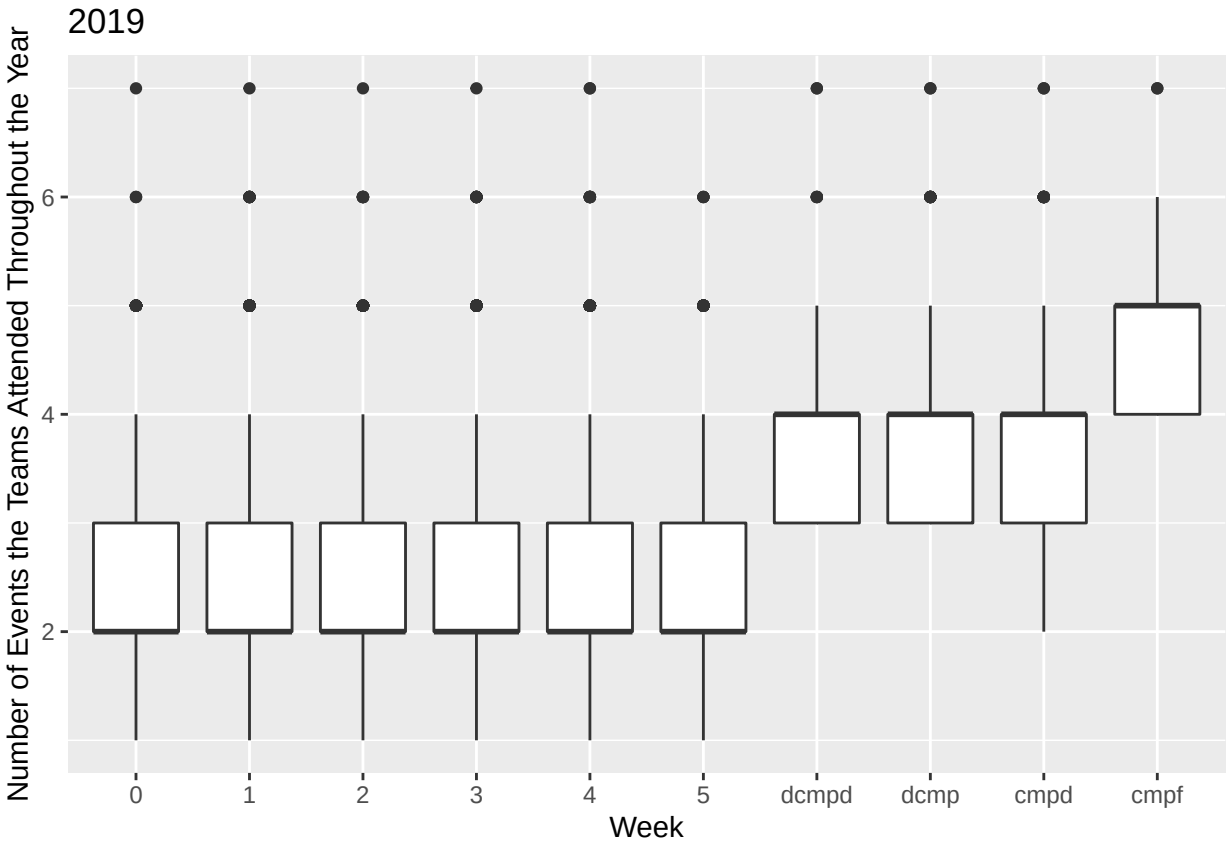
Number of Events Attended and Performance

Our question here is to figure out how attending multiple events can impact your team's performance and development. First, we will divide teams based on how many events they attended in a given season and see how the score percentile compares between number of events teams attended.

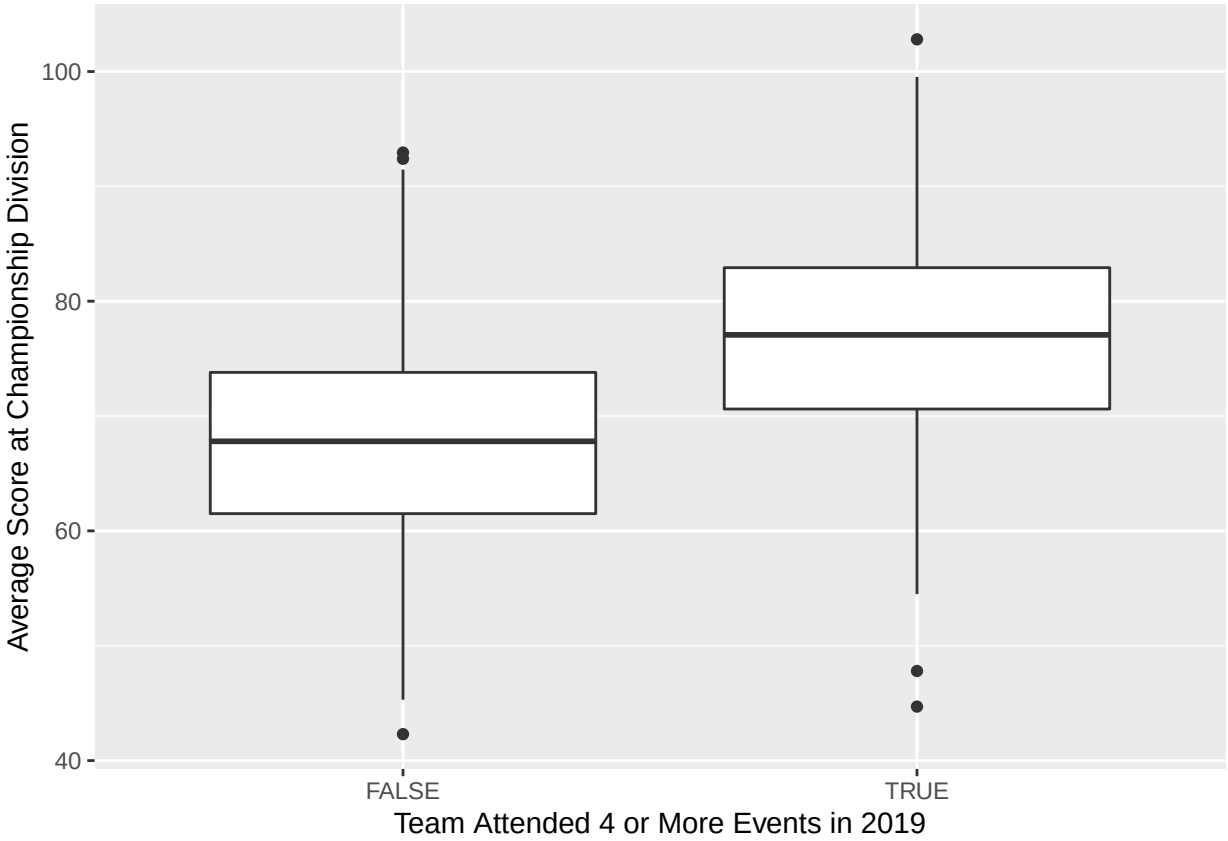


These results aren't too shocking, but they give us some useful information to keep in mind later. Clearly, the average score percentile for a team that attends a lot of events will be higher than a team that attends few since higher scoring teams are able to compete in more events.

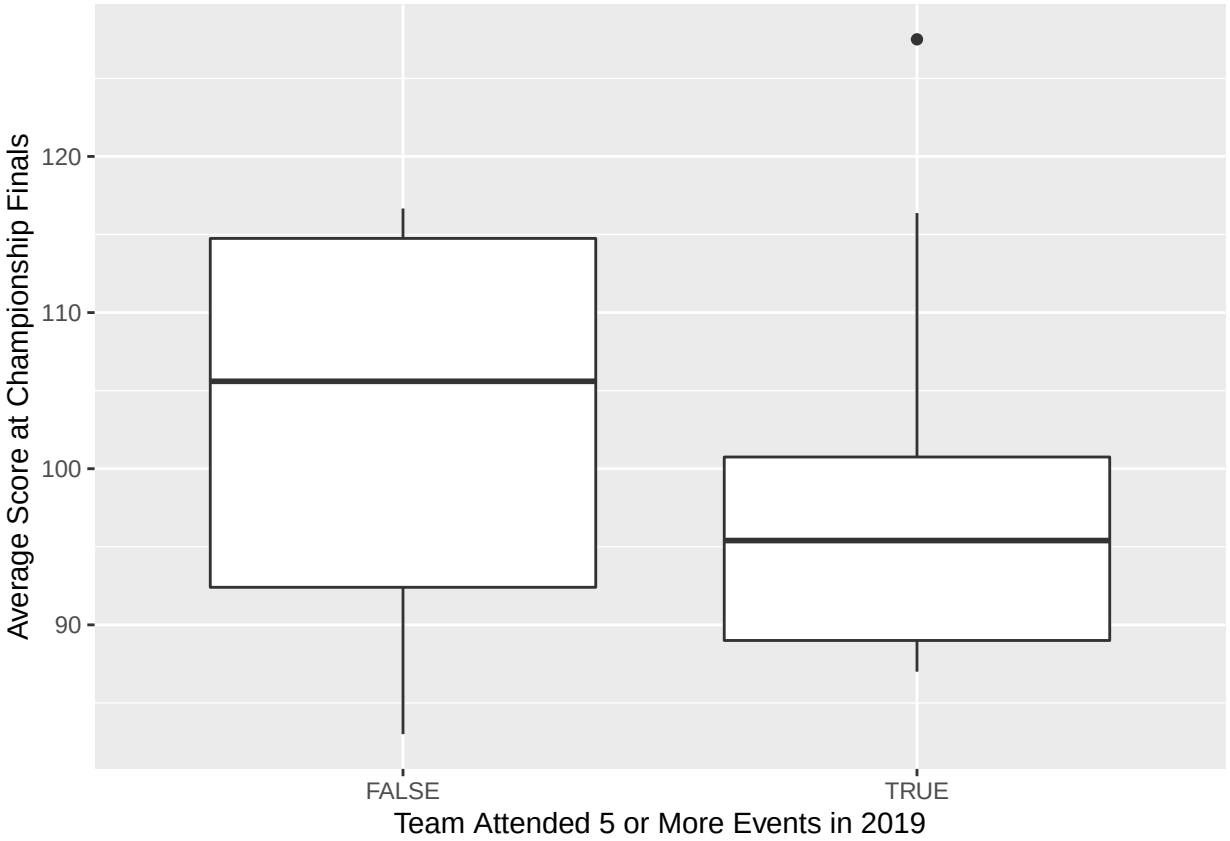
It is interesting to see how high a team's score percentile needs to be in order for them to expect to compete in many events. For example, in 2019 there were only a few teams that were below the 80th percentile that competed in 5 events.



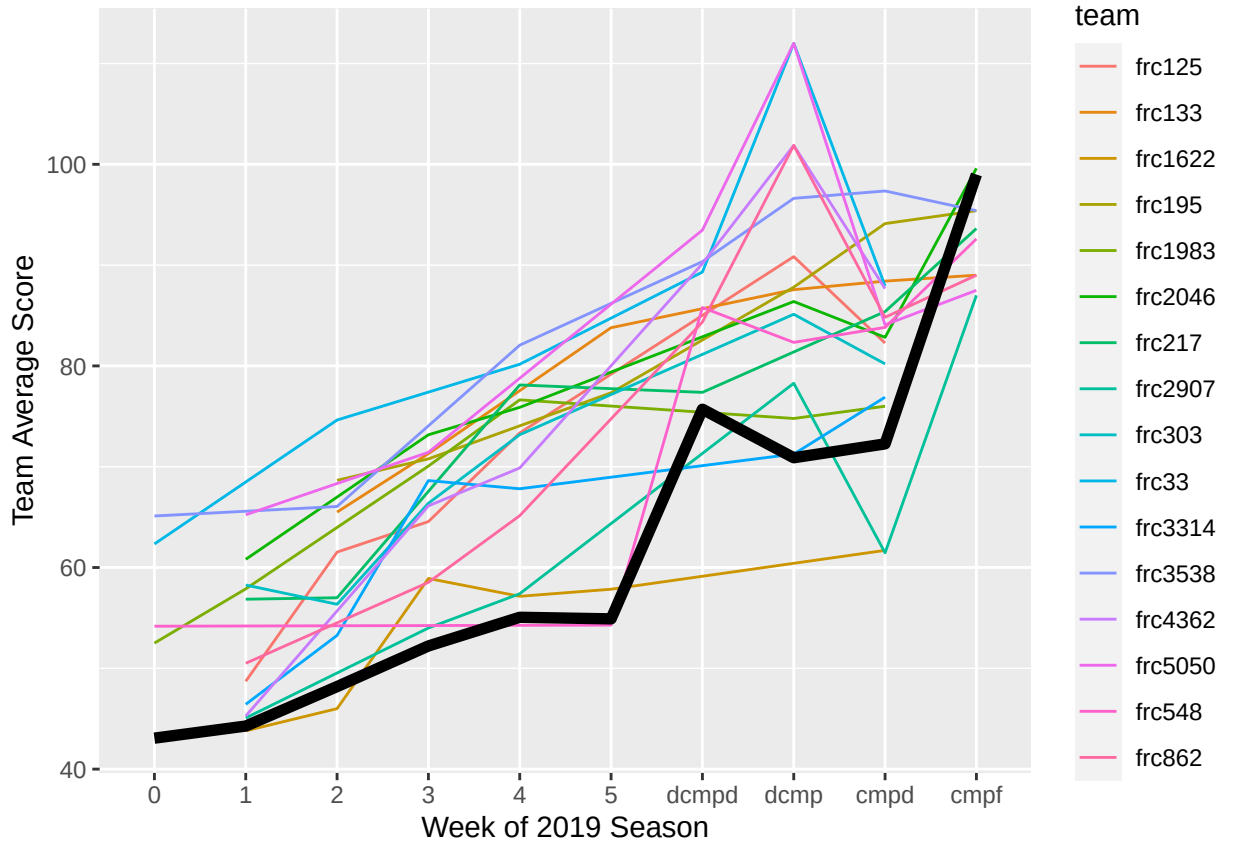
These boxplots don't reveal too many interesting trends, but they give us an understanding of how the average number of events a team participates is based on how far they make it in a season. When we investigate the 2019 scores of the championship division and the championship finals, we can use the median number of events attended 4 and 5, respectively to compare the team scores between teams that attended a lot of events and few events.



This is the breakdown of teams in the championship division based on the 4 events attended cutoff we found previously. This plot is showing that the teams that attended 4 or more events in 2019 score better in this event than those who didn't. We can't use this as evidence that a team that has more experience will score better though since a winning team in the championship division will attend the championship finals, which adds to their event count. A more interesing plot is this same boxplot for the championship finals below.



This plot tells the opposite story from before. The teams that made it to the finals who competed in more events actually performed worse in this case. This shows experience in a single season isn't the most reliable indicator of season performance.



Here is a breakdown of score progression throughout the 2019 season for teams that competed in 6 or more events. These are the teams that competed in the most events of all the teams. All of the teams have positive trends that indicate the teams are still progressing through each of the many events they attend in the 2019 season.

The black line is the team average score each week for all the teams, not just the one's who attended a lot of events. The scores of the more experienced teams are usually higher than this average for each week. The black line has a few sudden jumps, which can be explained by the jump from regular season to postseason, and the jump to the finals. The jumps are caused by the competition level increases that eliminate lower scoring teams.