The second ISL season wrapped up a couple of weeks ago, meaning that now we're in the off-season. I loved watching the CBC's coverage of the ISL. The races were exciting, the swimmers seemed like they were having a great time, they were absolutely swimming great times. My other sporting love is the NBA and I massively enjoyed the games in the Orlando bubble, but I never for a second wished I was down at Disney World seeing them in person. The ISL Budapest bubble though – man did I ever want to be there.

In lieu of being on deck though I stayed home and worked on the ISL-centric features of `SwimmeR`, namely the `swim_parse_ISL` function. I'm doing this because I love swimming and I want it to thrive and I want the ISL to succeed. `SwimmeR` is my tiny contribution to that effort. I'm skeptical though, skeptical about the viability of a professional swimming league, and skeptical about this particular incarnation of such a league.

One area that helps a sporting league make money and sustain itself is being able to drive interest even during the off-season. The NBA is great about this, especially over the last few years, as many articles have noted. Within the swimming world the importance of driving interest year round is recognized as well, as, Torrey Hart recent discussed with Mel Stewart, following her article on the business of the ISL. One of the things that makes the NBA work so well in the off-season is all the trades and salary cap machinations and the associated analytics. People love that stuff and they pay attention to it.

My original plan for this post was to do a bit of web scrapping using `swim_parse_ISL` to build data sets of all the International Swimming League results in the two year history of the league. Then I was going to put on my best Zach Lowe disguise to do some slick analytics on that data and make some trenchant points about roster construction for each team. I'm still going to do the first part, building the data set for all of you. The second part though, about the analytics, is premature.

I have a theory about sports analytics: they basically operate in the margins. It's not a terribly revolutionary theory, in fact I suspect most analytics aficionados would agree. Lets examine what what means though in the context of two different leagues, the National Basketball Association (NBA) and the International Swimming League (ISL).

There are two relevant characteristics of the NBA with respect to analytics. One is the totality of the NBA. All, or very nearly all eligible players who are known, or even thought to be capable of contributing to an NBA team are already on NBA rosters. Yes there's the occasional Sergio Llull who decides to play at a high level in other leagues, and there are plenty of guys who are talented by struggle with the professionalism the NBA requires, but it wouldn't be possible to

assemble a competitive NBA roster from eligible players not in the league. So the thinking goes at least. NBA front offices, and their analytics departments, spend their time looking for players that other teams don't think can make it, but actually can – the Duncan Robinsons and Fred VanVleets of the world. The difference between the consensus about Robinson (can't play in the league) and the reality (can excel in the league) is the margin in which the Miami Heat front office is operating. Some strong analytics can make that margin bigger – that's the goal.

The situation in the ISL is quite different. A very plausible plan for any ISL team looking to improve is to sign Simone Manuel, Olympic and world champs gold medalist in the 100 freestyle, or Xu Jiayu, 100 backstroke world championships gold medalist, or any of of the many other top flight swimmers who have never swam in the ISL. Heck since the ISL swimmers don't have contracts like NBA players, just do what London Roar GM Rob Woodhouse suggested on the championship broadcast and sign Caeleb Dressel away from the Condors – that would be great for any team. Tinkering with season 3 roster spots for potential or fringe contributors, just doesn't make sense until we know what top level talent is doing.

The second key difference between the NBA and ISL is the salary cap structure. The NBA cap is extremely complex, but for our purposes it's enough to say that there's a limit to the total salary that each team is allowed to pay its players. Since individual players would like to be paid as much as possible this sets up a market, where teams will offer players different salaries, and the player will chose to play for the team that offers him the largest one. (I realize in practice this doesn't exactly happen, but stick with me.) Teams, and their analytics departments aren't trying to determine if a player is good or not, so much as they're trying to determine how good a player is in relation to his market value. Determining that a player is worth a 10 million dollar/year salary and then paying him that 10 million dollars/year salary isn't a win for an analytics department. A win is getting a 10 mil/year worth of value from a player for less than 10 mil/year – the difference between value and salary is the margin. The salary cap also means that NBA teams can't just go and sign a full roster of A-list players. Those players want the big salaries that they earn with their excellent play, and teams are only allowed to pay out so much salary.

ISL athletes aren't paid in the same way though, and there's no cap. There's nothing to prevent the reigning champion Condors from signing more awesome swimmers, and they don't have to worry about exactly how to value a Simone Manual-level addition, they just need to know if she's better than whoever is in their last roster spot. It's a much simpler decision.

The ISL isn't ready yet for the kind of analytics in use in the NBA. It's not needed. If you're a ISL GM what you can do to improve your team over the off-season is just find the fastest swimmers available and get them to join. No special insights are required to identify them – their medals are easy tell. Once all that's sorted though – then maybe some analytics in the margins.

On a larger level, the uncertainty about the ISL, about what Season 3 will look like, and frankly when and if it will happen is what makes off-season coverage so difficult and makes it difficult to sustain excitement. I watched the excellent championship match, with all of its close races and world records and fantastic performances and when it was over I wasn't sure it would ever happen again. I'm still not, but I am hopeful.

Okay, `R` people – you've waited very patiently – here's how we're going to put together the ISL data sets.

Web Scraping

```
library(SwimmeR)
library(rvest)
```

```
library(stringr)
library(dplyr)
library(purrr)
library(ggplot2)
```

ISL releases their results as .pdf files, and I've collected all of those files on github.

The process here is about the same as in my previous web-scraping post.

1. Get web address for where the stuff of interest is
2. Identify CSS for the stuff
3. Make list of links to the stuff with `rvest` functions `html_nodes` and `html_attr`
4. Clean up list of links
5. Use `SwimmeR` functions to grab contents of list

We'll start with season 2, since it's the most recent one. Season 1 will be handled later with the same methods.

The web address is easy, just copy it out of your browser window.

```
# Part 1 - web address
web_url_season_2 <- "https://github.com/gpilgrim2670/Pilgrim_Data/tree/master
/ISL/Season_2_2020"
```

CSS can be a bit more complicated, but some clicking around with the selector gadget got me what we need.

```
# Part 2 - CSS
selector <- ".js-navigation-open"
```

Now for `rvest`. We'll use `read_html` to get the contents of `web_url_season_2` and then `html_attr` and `html_nodes` to get those contents which are links.

```
# Part 3 - rvest fun
page_contents <- read_html(web_url_season_2)
links_season_2 <- html_attr(html_nodes(page_contents, selector),
"href")
web_url_season_2 <- "https://github.com/gpilgrim2670/Pilgrim_Data/tree/master
/ISL/Season_2_2020"
head(links_season_2, 10)
##  [1] ""
##  [2] ""
##  [3] ""
##  [4] "/gpilgrim2670/Pilgrim_Data/tree/master/ISL"
##  [5] "/gpilgrim2670/Pilgrim_Data/blob/master/ISL/Season_2_2020/
ISL%202020%20Season%202%20Notes.txt"
##  [6] "/gpilgrim2670/Pilgrim_Data/blob/master/ISL/Season_2_2020/
ISL_01112020_Budapest_Match_6.pdf"
##  [7] "/gpilgrim2670/Pilgrim_Data/blob/master/ISL/Season_2_2020/
ISL_05112020_Budapest_Match_7.pdf"
##  [8] "/gpilgrim2670/Pilgrim_Data/blob/master/ISL/Season_2_2020/
ISL_05112020_Budapest_Match_8.pdf"
##  [9] "/gpilgrim2670/Pilgrim_Data/blob/master/ISL/Season_2_2020/
ISL_09112020_Budapest_Match_10.pdf"
```

```
## [10] "/gpilgrim2670/Pilgrim_Data/blob/master/ISL/Season_2_2020/
ISL_09112020_Budapest_Match_9.pdf"
```

The first 5 elements (`R` indexes start at 1) I don't want. Maybe if I was better at CSS selectors I could have avoided them outright, but I didn't. I'll just get rid of them here. The links are also partials, missing their beginnings, so we'll add "http://github.com" to the beginning of each with `paste0`. We'll also need to change "blob" to "raw" so that we just get the .pdfs, rather than the .pdfs and a github landing page. Little github trick there.

```
# Part 4 - cleaning
links_season_2 <- links_season_2[6:17] # only want links 6-17
links_season_2 <- paste0("https://github.com", links_season_2) # add
beginning to link
links_season_2 <- str_replace(links_season_2, "blob", "raw") # replace
blob with raw
```

Now it's just up to `SwimmeR`, with a bit of help from `purrr`. We'll use `read_results` inside of `map` to read in all of the elements of `links_season_2`. Then we'll map again, to apply `swim_parse_ISL`, inside of `safely` to all the results we read in. We'll also set `splits = TRUE` and `relay_swimmers = TRUE` to capture both splits and relay swimmers. After that we'll name each of our results based on their source link and then stick them all together with `bind_rows`.

```
# Part 5 - deploy SwimmeR
season_2 <- map(links_season_2, read_results)
season_2_parse <- map(season_2, safely(swim_parse_ISL, otherwise = NA),
splits = TRUE, relay_swimmers = TRUE)
names(season_2_parse) <- str_split_fixed(links_season_2, "/", n = 10)[,
10]
season_2_parse <- SwimmeR:::discard_errors(season_2_parse)
season_2_df <- bind_rows(season_2_parse, .id = "ID")
```

A Season 2 Demo

Just to show you what we've got in this data set here's a little exercise to get the major contributors to each team's point total. Since the ISL is focusing on points and contributing to one's team rather than times we'll do the same.

```
season_2_df <- season_2_df %>%
  mutate(ID = str_remove(ID, "\\.pdf\\.result")) %>%
  mutate(Start_Date = as.Date(str_split_fixed(ID, "_", 4)[, 2], format
= "%d%m%Y")) %>%
  mutate(Match = str_split_fixed(ID, "_", 4)[, 4]) %>%
  mutate(Season = 2) %>%
  arrange(Start_Date) %>%
  select(-ID)
```

ISL teams have 28 swimmers eligible for individual events. If they all contributed equally to the team score then each would account for about 3.5% of the total. That doesn't happen though, most teams have a few athletes who score most of the points, and the rest are what Shaquille O'Neal refers to as "others" - role players who contribute in different ways. We're going to name all the athletes who scored more than 7% of their team's total as the major contributors - those

athletes are worth twice their share of points.

What we'll do here is collect all athletes scoring less than 7% of their team's total points as "Other" and look at the breakdown for each team. This is also useful because trying to differentiate 28 different colors or point shapes or whatever on a plot, one for each athlete, is real pain.

First we'll sum up each team's total score. Mind you some teams competed in the post season and some did not. Teams that participated in the Semis and Championships had the most opportunity to earn points. By dividing each athlete's total by their team's total we're removing this effect. It's still worth noting though that an athlete scoring 15% of the league leading California Condors' points has scored a lot more points than another athlete who scored 15% of the league trailing DC Trident.

```
team_scores <- season_2_df %>%
  filter(is.na(Name) == FALSE) %>%
  group_by(Team) %>%
  summarise(Team_Score = sum(Points, na.rm = TRUE)) # score for each
team
```

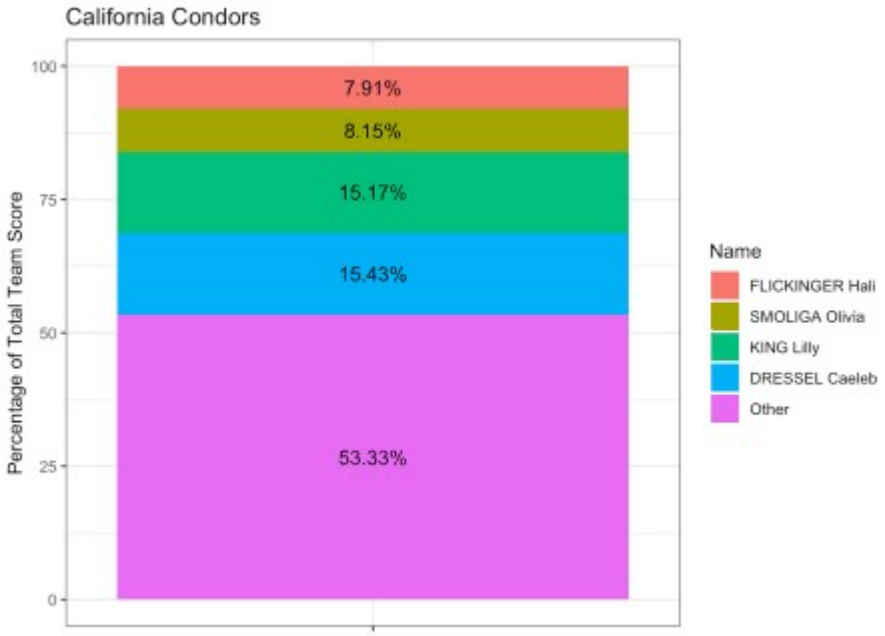
Now we need the total scores for each swimmer as a percentage of their team's total score.

```
individual_scores <- season_2_df %>%
  filter(is.na(Name) == FALSE) %>%
  group_by(Name) %>%
  summarise(
    Score = sum(Points, na.rm = TRUE), # score for each athlete
    Team = unique(Team)
  ) %>%
  group_by(Team) %>%
  left_join(team_scores) %>% # get team scores
  mutate(Score_Perc = 100 * round(Score/Team_Score, 5)) %>% # calculate
percentage of team score for each athlete
  mutate(Name = case_when(Score_Perc < 7 ~ "Other", # rename athletes
with less than 7% as other
                          TRUE ~ Name)) %>%
  group_by(Name, Team) %>%
  summarise(Score = sum(Score, na.rm = TRUE), # collect all "others"
together
            Score_Perc = sum(Score_Perc)) %>%
  group_by(Team) %>%
  mutate(Score_Rank = rank(desc(Score_Perc))) # get numeric rank for
each athlete and other in terms of their percent of the total team
score
```

Here's the Condors. Four swimmers scored almost 50% of their points, including our favorite, the undefeated queen of ISL 100 breaststroke, Lilly King!

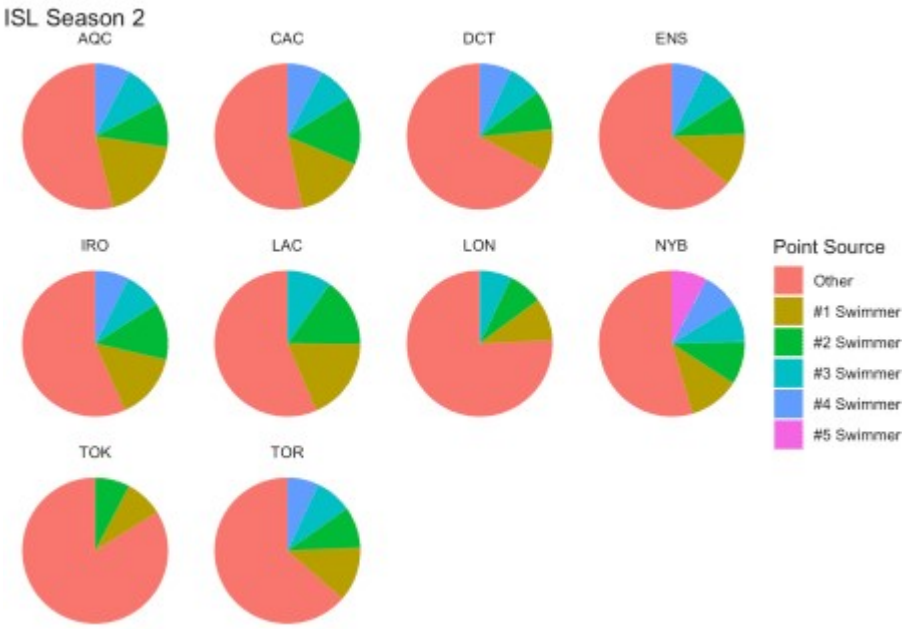
```
individual_scores %>%
  filter(Team == "CAC") %>% # just want cali condors
  arrange(Score_Perc) %>%
  mutate(Name = factor(Name, unique(Name))) %>% # order athlete names
by sore_perc
  ggplot(aes(
```

```
    x = "",
    y = Score_Perc,
    fill = Name,
    label = paste0(round(Score_Perc, 2), "%") # put score_perc values
in plot
  )) +
  geom_bar(width = 1, stat = "identity") +
  geom_text(size = 4, position = position_stack(vjust = 0.5)) +
  theme_bw() +
  theme(axis.text.x = element_blank(),
        axis.title.x = element_blank()) +
  labs(title = "California Condors",
       y = "Percentage of Total Team Score")
```



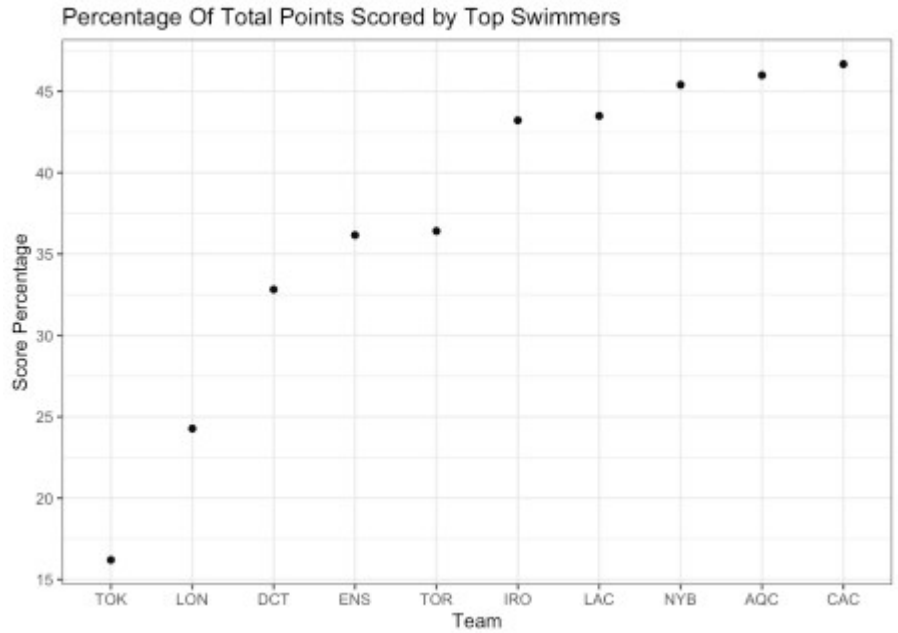
Pie charts aren't terribly well loved in the data-vis community. They're fairly panned as being hard to get values off of, and the colors can make them difficult to read. Pie charts are pretty though - they look like flowers. Just look at this, ten pretty, pretty data flowers, one for each team.

```
individual_scores %>%
  ggplot(aes(x = "", y = Score_Perc, fill = as.factor(Score_Rank))) +
  geom_bar(width = 1, stat = "identity") +
  coord_polar("y", start = 0) + # converts bar chart into pie chart
  scale_fill_discrete(name = "Point Source", labels = c("Other", "#1
Swimmer", "#2 Swimmer", "#3 Swimmer", "#4 Swimmer", "#5 Swimmer")) + #
labels for colors
  facet_wrap(. ~ Team) + # one plot per team
  theme_void() +
  theme(axis.text.x = element_blank()) +
        # legend.position = "none") +
  labs(title = "ISL Season 2")
```



We can convey a similar point with a chart like this - the percent of each team's points scored by their top (non-other) athletes.

```
individual_scores %>%
  filter(Name != "Other") %>%
  group_by(Team) %>%
  summarise(Score_Perc = sum(Score_Perc)) %>%
  arrange(Score_Perc) %>%
  mutate(Team = factor(Team, unique(Team))) %>%
  ggplot(aes(x = reorder(Team, Score_Perc), y = Score_Perc)) +
  geom_point() +
  scale_y_continuous(breaks = seq(5, 50, 5)) +
  theme_bw() +
  labs(title = "Percentage Of Total Points Scored by Top Swimmers",
       x = "Team",
       y = "Score Percentage")
```



Some teams, like the London Roar are more balanced than others, like the Condors are more

top heavy. There's not an obvious lesson to be drawn here though, because LON, CAC, ENS and LAC where the best teams in the league in Season 2 and they're all over this plot. Also there's nothing to be surmised about next season, because we don't know anything about next season. We're just doing this for fun.

Season 1 Data Set

The ISL made a lot of changes between season 1 and season 2, so they're not directly comparable. It's still possible to build a data set from season 1 though. The methods are exactly the same as for season 2 above.

```
web_url_season_1 <- "https://github.com/gpilgrim2670/Pilgrim_Data/tree/master
/ISL/Season_1_2019"
selector <- ".js-navigation-open"

page_contents <- read_html(web_url_season_1)
links_season_1 <- html_attr(html_nodes(page_contents, selector),
"href")
links_season_1 <- links_season_1[5:18]
links_season_1 <- paste0("https://github.com", links_season_1)
links_season_1 <- str_replace(links_season_1, "blob", "raw")

season_1 <- map(links_season_1, read_results)
season_1_parse <- map(season_1, safely(swim_parse_ISL, otherwise = NA),
splits = TRUE, relay_swimmers = TRUE)
names(season_1_parse) <- str_split_fixed(links_season_1, "/", n = 10)[,
10]
season_1_parse <- SwimmeR:::discard_errors(season_1_parse)
season_1_df <- bind_rows(season_1_parse, .id = "ID") %>%
  mutate(ID = str_remove(ID, "\\.pdf\\.result")) %>%
  mutate(Start_Date = as.Date(str_split_fixed(ID, "_", 4)[, 2], format
= "%d%m%Y")) %>%
  mutate(Match = str_split_fixed(ID, "_", 5)[, 3]) %>%
  mutate(Season = 1) %>%
  arrange(Start_Date) %>%
  select(-ID)
```
