```
NY_Boys <- "http://www.nyhsswim.com/Results/Boys/2020/NYS/Single.htm"
NY_Girls <- "http://nyhsswim.com/Results/Girls/2019/NYS/Single.htm"

NY_Links <- c(NY_Boys, NY_Girls)
```

The goal is to read in and clean the raw results, which we'll do with `SwimmeR::read_results` and `SwimmeR::swim_parse` respectively. Before doing so however it's useful to look the raw results over for potential issues.

We can see in the New York raw results that federation and state records for the two meets are recorded as "Federation", "NYS Fed", "NYSPHSAA" and "NYS Meet Rec". Those are strings we'll want to tell `SwimmeR::swim_parse` to `avoid`.

```
Event 1  Boys 200 Yard Medley Relay
================================================================================
     NYS Fed: F 1:33.32        2017 St. Anthony's, St. Anthony's
                               C.Rutigliano, M.Chang, A.Stange, J.Meyn
    NYSPHSAA: N 1:33.42        2017 Half Hollow Hills
                               D.Chan, A.Park, L.Tack, J.Meyn
NYS Meet Rec: M 1:33.32        2017 St. Anthony's, St. A
                               C.Rutigliano, M.Chang, A.Stange, J.Meyn
                1:33.21  AA
                1:34.74  AC
                1:40.67  NYS
    School                                      Prelims     Finals       Points
================================================================================
NYSPHSAA Federation Championships
A - Final
  1 Pittsford-5                                       1:34.91    1:34.36 AC      48
     1) Kusch, Aaron JR                  2) r:+0.0 Novozhenets, Jacob SR
     3) r:+0.0 Mortimer, Neil SR         4) r:0.46 Murphy, Liam SR
```

NY_Boys_Header

```
Event 1  Girls 200 Yard Medley Relay
================================================================================
  Federation: F 1:43.21  11/19/2016 Long Beach (8)
                         K Romano, M Aroesty, C Farrell, J Cash
    NYSPHSAA: N 1:43.21        2016 Long Beach (8)
                         K Romano, M Aroesty, C Farrell, J Cash
                1:44.21  AAA
                1:46.21  AAC
                1:52.35  NYS
    School                                      Prelims     Finals       Points
================================================================================
NYSPHSAA Federation Championships
Finals
  1 Sacred Heart Academy-C                            1:43.68    1:42.21FAAA     48
     1) Tess Howley FR                   2) r:0.33 Ariana Brattoli JR
     3) r:0.17 Cavan Gormsen FR          4) r:0.17 Joan Cash JR
     r:+0.58  25.26       54.63 (29.37)     1:18.83 (24.20)     1:42.21 (23.38)
  2 Pittsford-5                                       1:44.58    1:44.63 AAC     42
```

NY_Girls_Header

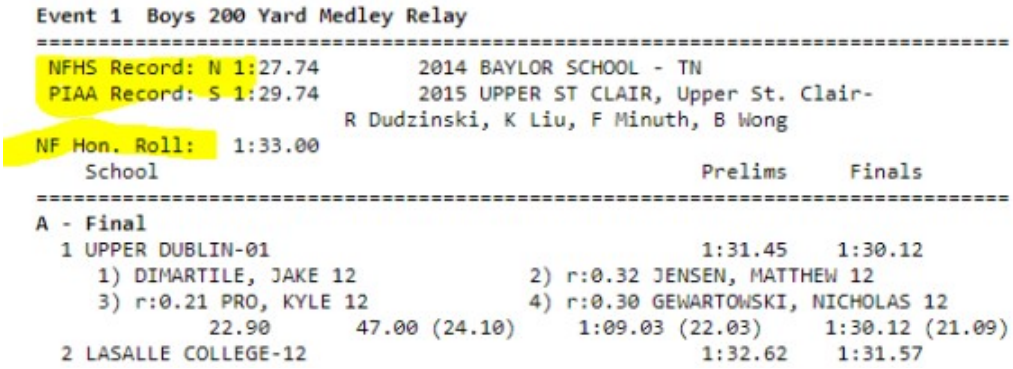
```
NY_Avoid <- c("Federation", "NYS Fed", "NYSPHSAA", "NYS Meet Rec")
```

Pennsylvania is a similar story, with a nice results repository. Unlike New York however Pennsylvania has two different divisions for their state championships, somewhat confusingly called 2A and 3A. The 3A championships were held in 2020 (boys and girls) but the 2A where canceled due to COVID-19. Also diving wasn't included in the Girls 3A 2020 results so as State-Off meet director I'll be subsisting 2019 results for 3A diving and all of 2A. There will be five total links.

```
PA_Boys_3A <- "http://www.paswimming.com/19_20/results/state/PIAA_3_A_boys_states_Results.htm"
PA_Girls_3A <- "http://www.paswimming.com/19_20/results/state/PIAA_3_A_girls_states_Results.htm"
PA_Girls_3A_Diving <- "http://www.piaa.org/assets/web/documents/2019_3a_girls_f_dive_results.htm"
PA_Boys_2A <- "http://www.paswimming.com/18_19/results/states/Results/2_A_Boys_Results_2019.htm"
PA_Girls_2A <- "http://www.paswimming.com/18_19/results/states/Results/2_A_Girls_Results_2019.htm"

PA_Links <- c(PA_Boys_3A, PA_Girls_3A, PA_Girls_3A_Diving, PA_Boys_2A, PA_Girls_2A)
```

Inspecting the Pennsylvania raw results gives us a few more strings to avoid, namely “PIAA” (PA record), plus “NFHS” and “NF Hon. Roll”.



```
Event 1  Boys 200 Yard Medley Relay
================================================================================
 NFHS Record: N 1:27.74        2014 BAYLOR SCHOOL - TN
 PIAA Record: S 1:29.74        2015 UPPER ST CLAIR, Upper St. Clair-
                          R Dudzinski, K Liu, F Minuth, B Wong
 NF Hon. Roll:   1:33.00
    School                                            Prelims     Finals
================================================================================
A - Final
  1 UPPER DUBLIN-01                                   1:31.45    1:30.12
     1) DIMARTILE, JAKE 12              2) r:0.32 JENSEN, MATTHEW 12
     3) r:0.21 PRO, KYLE 12             4) r:0.30 GEWARTOWSKI, NICHOLAS 12
             22.90       47.00 (24.10)     1:09.03 (22.03)     1:30.12 (21.09)
  2 LASALLE COLLEGE-12                                1:32.62    1:31.57
```

```
PA_Avoid <- c("PIAA", "NFHS", "NF Hon. Rol")
```

Reading in Results with the SwimmeR Package

Getting our results is now a simple matter of `map`ping `read_results` and `swim_parse` over our list of links with our avoid lists passed to the `avoid` argument of `swim_parse`.

We’ll then add columns `State` and `Gender` since those are the parameters of our meet – each state is a team, with boys and girls meets, plus a combined total.

```
Results <- map(c(NY_Links, PA_Links), Read_Results, node = "pre") %>%
  map(Swim_Parse, avoid = c(NY_Avoid, PA_Avoid)) %>%
  set_names(c("NY_Boys", "NY_Girls", "PA_Boys", "PA_Girls", "PA_Girls",
"PA_Boys", "PA_Girls")) %>%
  bind_rows(.id = "Source")

Results <- Results %>%
  mutate(
    State = str_split_fixed(Source, "_", n = 2)[, 1],
    Gender = str_split_fixed(Source, "_", n = 2)[, 2]
  ) %>%
  select(-Source) %>%
  filter(str_detect(Event, "Swim-off") == FALSE) # remove swim-offs
```

More Detail on Meet Parameters

We’ll use the National Federation of High School athletics scoring as below. It’s important to specify that 17th place gets 0 points when it comes to dealing with ties.

```
Point_Values <- c(20, 17, 16, 15, 14, 13, 12, 11, 9, 7, 6, 5, 4, 3, 2, 1, 0)
names(Point_Values) <- 1:17
```

In order to score the meet we need to reorder finishes from each state meet in the context of our larger meet. At first glance this is simple, because the fastest (i.e. lowest) time will win, followed by the second fastest/lowest in second etc. There are several complications though.

1. Unique swims: In the New York results there are “Federation” results and “Association” results for each event. Federation is a subset of Association though, so athletes/relay teams in the Federation are listed twice, once in the Federation results and then again (with the same times) in the Association results. The Pennsylvania results include preliminary swims, so athletes/relay teams are also listed twice, once in the finals (which appear first) and again in the prelims, with different times in each instance. We’ll need a way to get only the first instance of an athlete or relay team in a given event.

2. Relays: Relays are different from individual swims for two reasons

- Naming: relays are named by the team/school (Central High), whereas athletes have both a team/school and an individual name (Sally Swimfast from Central High)
- Scoring: point values are doubled for relays

3. Ties: Ties happen, and the procedure (per NFHS rules) is for competitors to be awarded the average of their place and the voided place. For example, if two athletes tie for 9th place then there will be no 10th place finisher (both athletes get 9th, 10th is voided). The point value for 9th place is 9 points and the point value for 10th is 7, so each athlete receives (9 + 7) = 16, divided by two, equals 8 points. Our scoring needs to handle this.

4. Diving: Here at Swimming + Data Science we love diving even if it is a complication. We’re not going to just cut diving out, we’re going to deal with diving on its own terms. Diving results are different from swimming results for two reasons.

- Format: Diving results are scores not times.
- Ordering: The highest score in diving wins, compared to the fastest (i.e. lowest) time winning in swimming.

General Workflow

1. Break up `Results` into relays, diving, and individual swimming using `filter`.
2. Take only the first instance of an athlete/team in an event using `group_by` and `slice`.
3. For relays and individual swims convert times in minutes:seconds.hundreths to seconds with `SwimmeR:sec_format`.
4. Reorder and record finishes on basis of time (or score) across the new NY vs. PA meet using `arrange` and `mutate`.
5. Award points, accounting for ties using a nifty little combo of `rank`, `summarize` and `inner_join`

Relays

```
Relay_Results <- Results %>%
  filter(str_detect(Event, "Relay") == TRUE) %>% # only want relays
  group_by(Event, School) %>%
  slice(1) %>% # select first occurrence of team in each event
  ungroup() %>%
  mutate(Finals_Time_sec = sec_format(Finals_Time)) %>% # convert time to
seconds
  group_by(Event) %>%
  mutate(Place = rank(Finals_Time_sec, ties.method = "min")) %>% # places, low
number wins
  filter(Place <= 16) %>% # only top 16 score
  select(-Points)

Relay_Results <- Relay_Results %>% # deal with ties
  mutate(New_Place = rank(Place, ties.method = "first"),
         Points = Point_Values[New_Place]) %>%
```

```
  group_by(Place, Event) %>%
  summarize(Points = mean(Points)) %>%
  inner_join(Relay_Results) %>%
  mutate(Points = Points * 2) # double point values for relays
```

Diving

Same basic structure as relays, but we need to handle scores differently than times.

```
Diving_Results <- Results %>%
  filter(str_detect(Event, "Diving") == TRUE) %>% # only want diving events
  mutate(Finals_Time = as.numeric(Finals_Time)) %>%
  group_by(Event, Name) %>%
  slice(1) %>% # first instance of every diver
  ungroup() %>%
  group_by(Event) %>%
  mutate(Place = rank(desc(Finals_Time), ties.method = "min"), # again, highest
score gets rank 1
         Finals_Time = as.character(Finals_Time)) %>%
  filter(Place <= 16) %>% #only top 16 score
  select(-Points)

Diving_Results <- Diving_Results %>% # deal with ties
  mutate(New_Place = rank(Place, ties.method = "first"),
         Points = Point_Values[New_Place]) %>%
  group_by(Place, Event) %>%
  summarize(Points = mean(Points)) %>%
  inner_join(Diving_Results)
```

Individual Swimming

Again, very similar to diving and relays.

```
Ind_Swimming_Results <- Results %>%
  filter(str_detect(Event, "Diving") == FALSE,
         str_detect(Event, "Relay") == FALSE) %>%
  group_by(Event, Name) %>%
  slice(1) %>% # first instance of every swimmer
  ungroup() %>%
  group_by(Event) %>%
  mutate(Finals_Time_sec = sec_format(Finals_Time)) %>% # time as seconds
  mutate(Place = rank(Finals_Time_sec, ties.method = "min")) %>% # places, low
number wins
  filter(Place <= 16) %>% #only top 16 score
  select(-Points)

Ind_Swimming_Results <- Ind_Swimming_Results %>% # deal with ties
  mutate(New_Place = rank(Place, ties.method = "first"),
         Points = Point_Values[New_Place]) %>%
  group_by(Place, Event) %>%
  summarize(Points = mean(Points)) %>%
  inner_join(Ind_Swimming_Results)
```

The Final Results

Let's `bind` together the results from our three cases (relays, diving and individual swims) and do a but of cleaning up. Pennsylvania for example has all their results in block capitals. That can be fixed with `str_to_title`.

```
Results_Final <-
  bind_rows(Relay_Results, Diving_Results, Ind_Swimming_Results) %>%
  mutate(Name = str_to_title(Name),
         School = str_to_title(School)) %>%
  mutate(School = str_remove_all(School, "[:punct:]"),
         School = str_remove_all(School, "[0-9]"))
```

Scores

Now we `summarise` and see who won!

```
Scores <- Results_Final %>%
  group_by(State, Gender) %>%
  summarise(Score = sum(Points))

Scores %>%
  arrange(Gender, desc(Score)) %>%
  flextable() %>%
  bold(part = "header") %>%
  bg(bg = "#D3D3D3", part = "header")
```

State	Gender	Score
PA	Boys	1711.5
NY	Boys	613.5
PA	Girls	1524.0
NY	Girls	801.0

```
Scores %>%
  group_by(State) %>%
  summarise(Score = sum(Score)) %>%
  arrange(desc(Score)) %>%
  flextable() %>%
  bold(part = "header") %>%
  bg(bg = "#D3D3D3", part = "header")
```

State	Score
PA	3235.5
NY	1414.5

Pennsylvania wins both meets and the combined in an upset, by quite a wide margin!

It's interesting to think for a moment about why this might be. The State-Off is seeded by population. New York has about 19 million people, but about 8 million of them live in New York City. New York City doesn't have very many swimmers. Swimmers from the new York City Public High School Athletic League have a -P designation after their school name in the raw results. The cleaning we did on `Final_Results` reduced this to a trailing `P`, which we can search for with `str_detect`.

```
Results_Final %>%
  ungroup() %>%
  filter(str_detect(School, "P$")) %>%
  summarise(Count = n())
```

```
## # A tibble: 1 x 1
##   Count
##
```

```
## 1     3
```

```
Results_Final %>%
  ungroup() %>%
  filter(State == "NY") %>%
  summarise(Count = n())
```

```
## # A tibble: 1 x 1
##   Count
##
## 1   134
```

Only three swims out of New York's total of 134 swims are from New York City. Pools take up a lot space so they're difficult to install in cities generally. New York City is also very dense, which makes building pools that much harder. Pennsylvania on the other hand has a total population of 12 million. Philadelphia (1.5 million) and Pittsburgh (300k) are much smaller than New York City, so it's possible that much more of the Pennsylvania population lives in areas conducive to swimming. There's also a racial component. New York City has a higher proportion of African American residents than New York State as a whole, and African Americans have been subjected to segregation and systematic discrimination including specifically with respect to swimming pools to the extent that even today black children drown at a rate 3x that of white children. New York's larger than expected non-swimming population may be reflected in its lower than expected State-Off score.

Swimmers of the Meet

To determine the swimmers of the meet there will be two qualifications:
1. An athlete must have competed in two events – sorry divers. Winner will be the athlete with the lowest average place (winning two events gives an average place of 1). This is an individual award so relays don't count.
2. As a tiebreaker from 1. above, the athlete whose times are fastest across their two events relative to the All-American cuts will be Swimmer of the Meet.

Now if only someone had the All-American cuts readily accessible. Oh wait someone does and that someone is me. Let's grab those cuts and join them to `Ind_Swimming_Results`. Then we can do some math to calculate each athlete's average difference from the All-American cut.

```
Cuts_Link <- "https://raw.githubusercontent.com/gpilgrim2670/Pilgrim_Data/master/State_Cuts.csv"
Cuts <- read.csv(url(Cuts_Link))

'%!in%' <- function(x,y)!('%in%'(x,y)) # "not in" function

Cuts <- Cuts %>% # clean up Cuts
  filter(Stroke %!in% c("MR", "FR", "11 Dives")) %>%
  rename(Gender = Sex) %>%
  mutate(
    Event = case_when((Distance == 200 & #match events
                         Stroke == 'Free') ~ "200 Yard Freestyle",
                      (Distance == 200 &
                         Stroke == 'IM') ~ "200 Yard IM",
                      (Distance == 50 &
                         Stroke == 'Free') ~ "50 Yard Freestyle",
                      (Distance == 100 &
                         Stroke == 'Fly') ~ "100 Yard Butterfly",
                      (Distance == 100 &
                         Stroke == 'Free') ~ "100 Yard Freestyle",
                      (Distance == 500 &
                         Stroke == 'Free') ~ "500 Yard Freestyle",
```

```
                        (Distance == 100 &
                           Stroke == 'Back') ~ "100 Yard Backstroke",
                        (Distance == 100 &
                           Stroke == 'Breast') ~ "100 Yard Breaststroke",
                        TRUE ~ paste(Distance, "Yard", Stroke, sep = " ")),

    Event = case_when(Gender == "M" ~ paste("Boys", Event, sep = " "),
                      Gender == "F" ~ paste("Girls", Event, sep = " ")))

Ind_Swimming_Results <- Ind_Swimming_Results %>%
  left_join(Cuts %>% filter((Gender == "M" &
                               Year == 2020) |
                              (Gender == "F" &
                                 Year == 2019)) %>%
                     select(AAC_Cut, AA_Cut, Event),
            by = 'Event')

Swimmer_Of_Meet <- Ind_Swimming_Results %>%
  mutate(AA_Diff = (Finals_Time_sec - sec_format(AA_Cut))/sec_format(AA_Cut),
         Name = str_to_title(Name)) %>%
  group_by(Name) %>%
  filter(n() == 2) %>% # get swimmers that competed in two events
  summarise(Avg_Place = sum(Place)/2,
         AA_Diff_Avg = round(mean(AA_Diff, na.rm = TRUE), 2),
         Gender = unique(Gender),
         State = unique(State)) %>%
  arrange(Avg_Place, AA_Diff_Avg) %>%
  group_split(Gender) # split out a dataframe for boys (1) and girls (2)
```

Boys

Boys swimmer of the meet is Matt Brownstead from Pennsylvania, the only boy to win two events! He also broke the national high school record in the 50 free. Let's see his results.

```
Swimmer_Of_Meet[[1]] %>%
  slice_head(n = 5) %>%
  select(-Gender) %>%
  flextable::flextable() %>%
  bold(part = "header") %>%
  bg(bg = "#D3D3D3", part = "header")
```

Name	Avg_Place	AA_Diff_Avg	State
Brownstead, Matt	1.0	-0.05	PA
Jensen, Matthew	1.5	-0.04	PA
Faikish, Sean	1.5	-0.03	PA
Newmark, Jake	1.5	-0.02	NY
Guiliano, Chris	2.0	-0.02	PA

```
Results_Final %>%
  filter(Name == "Brownstead, Matt") %>%
```

```
  select(Place, Name, School, Finals_Time, Event) %>%
  arrange(desc(Event)) %>%
  flextable::flextable() %>%
  bold(part = "header") %>%
  bg(bg = "#D3D3D3", part = "header")
```

Place	Name	School	Finals_Time	Event
1	Brownstead, Matt	State College	19.24	Boys 50 Yard Freestyle
1	Brownstead, Matt	State College	43.29	Boys 100 Yard Freestyle

Girls

As for the girls the competition was a bit tighter, with two athletes, Chloe Stepanek and Megan Deuel, both winning two events. Going to our All-American standard tiebreaker gives the win to Chloe Stepanek! Winning here is hopefully some solace for Chloe after Megan won the award at the NYS girls meet.

```
Swimmer_Of_Meet[[2]] %>%
  slice_head(n = 5) %>%
  select(-Gender) %>%
  flextable::flextable() %>%
  bold(part = "header") %>%
  bg(bg = "#D3D3D3", part = "header")
```

Name	Avg_Place	AA_Diff_Avg	State
Chloe Stepanek	1.0	-0.03	NY
Megan Deuel	1.0	-0.02	NY
Catherine Stanford	1.5	-0.01	NY
Cavan Gormsen	2.0	-0.01	NY
Buerger, Torie	2.5	-0.01	PA

```
Results_Final %>%
  filter(Name == "Chloe Stepanek") %>%
  select(Place, Name, School, Finals_Time, Event) %>%
  arrange(desc(Event)) %>%
  flextable::flextable() %>%
  bold(part = "header") %>%
  bg(bg = "#D3D3D3", part = "header")
```

Place	Name	School	Finals_Time	Event
1	Chloe Stepanek	Northport	1:46.15	Girls 200 Yard Freestyle

Place	Name	School	Finals_Time	Event
1	Chloe Stepanek	Northport	48.76	Girls 100 Yard Freestyle

In Closing

That wraps up this match up. Join us next time here at Swimming + Data Science for another Round 1 match up – number 1 seed California vs. number 8 seed Georgia. We'll see you then!