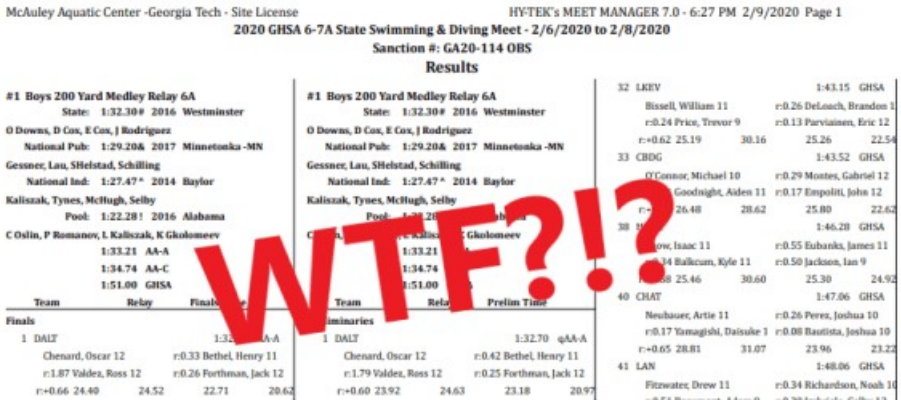
…We'll take define a base url and paste a sequence of numbers from "01" to "24" onto it, which will give us links to the pages for each event. The leading zero is important.

```
base <- "http://www.results.teamunify.com/clov/2019/CIFSTATEMEET/190510F0" # base url
event_numbers <- 1:24 # sequence of numbers, total of 24 evetns across boys and girls
event_numbers <- str_pad(event_numbers, width = 2, side = "left", pad = "0") # add leading
zeros to single digit numbers
CA_Links <- paste0(base, event_numbers, ".htm") # paste together base urls and sequence of
numbers (with leading zeroes as needed)

CA_Results <- map(CA_Links, read_results, node = "pre") %>% # map SwimmeR::read_results
over the list of links
  map(swim_parse) %>%
  bind_rows() %>% # bind together results from each link
  select(Name, School, Finals_Time, Event) %>% # only the columns we need
  mutate(State = "CA") # add column for state since we'll be combining results with GA
```

Georgia (8)

Alright, it's time for us to have a chat. I've been to Georgia several times and would happily go again. I once spent a month living in Atlanta while doing research at Georgia Tech. On another occasion I swam at Masters Nationals when it was hosted by Georgia Tech and got to lose to Cullen Jones. All were great experiences. Georgia is a great state. Georgia Tech is a great university. Georgia's 2020 swimming data is not great. Georgia's 2020 swimming data is atrocious. Georgia's 2020 swimming data makes me feel like I felt the time I threw up in a Waffle House parking lot outside Brunswick, Georgia, while on a training trip. Georgia Tech hosted Georgia's 2020 state meet and generated the 2020 results so I'm blaming them. This is terrible Georgia Tech. This is beneath you. I expect better. Look at it, a three columns? All jacked up on the right border? What a mess! Get in touch Georgia Tech, I can help you out.



GA Header

Anyway, I'm hosting cleaned up data on github. We'll grab that and the State-Off will continue.

```
GA_Link <- "https://raw.githubusercontent.com/gpilgrim2670/Pilgrim_Data/master/GA_States_2020.csv"
GA_Results <- read.csv(url(GA_Link)) %>%
  select(Name, School, "Finals_Time" = Time, Event) %>%
  mutate(State = "GA")
```

Joining Up Results

Having collected results from California and Georgia we just need to join them up, add a column for gender and make sure the event names are consistent across the joined data set.

```
Results <- bind_rows(CA_Results, GA_Results) %>%
  mutate(Gender = case_when(str_detect(Event, "Girls") == TRUE ~ "Girls",
                            str_detect(Event, "Boys") == TRUE ~ "Boys")) %>%
  mutate(Event = case_when(Event == "Girls 1 mtr Diving" ~ "Girls 1m Diving", # make event
names consistent
                           Event == "Boys 1 mtr Diving" ~ "Boys 1m Diving",
                           TRUE ~ Event))
```

Analysis

So here's the thing about reproducible research: it's fantastic. Not only can you follow along with the analysis in a given post and reproduce the results for yourselves, as long as the inputs are structured the same way I can reuse my code across posts. Last week, for New York vs. Pennsylvania, I wrote a bunch of code to split out relays, diving and individual swimming events such that they could each be scored according to their specific requirements. Then I wrote some more code to score the meet, and still more to identify swimmers of the meet. It was work. I read documentation, I visited Stack Overflow, I visited R Bloggers, I tried ideas and experimented, I did all the normal data science development stuff. When the code was done we fed cleaned results, from `SwimmeR`, into that code to do our analysis. Well guess what? We've got cleaned results from `Swimmer` again this week. All that code I wrote last week? It still works!

I've left the code basically unchanged for this week in order to make my point about reusability, but next week, when we do Florida (3) vs. Illinois (6) I'm going to extend reusability even further, by functionalizing pieces of the code.

```
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
```

Relays

Entries have `School` but not `Name`. Point values are doubled.

```
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)

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 our treatment of relays, but we need to handle diving scores differently than swimming 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)

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)

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)
```

Final Results

```
Results_Final <-
  bind_rows(Relay_Results, Diving_Results, Ind_Swimming_Results)
```

One thing I have changed for this week is making the results tables with `gt` rather than `flextable`. Nothing wrong with `flextable`, but I wanted to try out `gt`. I like it, looks good. Anyways, California has won the boys, girls and combined meets by a comfortable margin. Georgia did well though, winning 8 events to California's 16. There's nothing particularly surprising about this outcome. Both California and Georgia have strong swimming traditions, including very successful collegiate programs. Both also have climates that suit the sport. California is just 4x the size of Georgia, population-wise, giving it a much larger population pool to draw talent from.

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

Scores %>%
  arrange(Gender, desc(Score)) %>%
  ungroup() %>%
  gt() %>%
  tab_header(
    title = md("**Meet Scores**"),
  )
```

Meet Scores

State	Gender	Score
CA	Boys	1748.5
GA	Boys	576.5
CA	Girls	1807.5
GA	Girls	517.5

```
Scores %>%
  group_by(State) %>%
  summarise(Score = sum(Score)) %>%
  arrange(desc(Score)) %>%
  ungroup() %>%
  gt() %>%
  tab_header(title = md("**Combined Meet Score**"))
```

Combined Meet Score

State	Score
CA	3556
GA	1094

```
Results_Final %>%
  filter(Place == 1) %>%
```

```
  select(Event, State) %>%
  group_by(State) %>%
  summarise(Total = n()) %>%
    gt() %>%
  tab_header(title = md("**Events Won by State**"))
```

Events Won by State

State	Total
CA	16
GA	8

Swimmers of the Meet

Just like above, all of this code is reusable from last week. Again we'll look for athletes who won two events, thereby scoring a the maximum possible forty points. We'll also grab the All-American cuts to use as a tiebreaker, in case multiple athletes win two events.

```
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 %>% # join Ind_Swimming_Results and Cuts
  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

```
Swimmer_Of_Meet[[1]] %>%
  slice_head(n = 5) %>%
  select(-Gender) %>%
  ungroup() %>%
  gt() %>%
  tab_header(title = md("**Boys Swimmer of the Meet**"))
```

Boys Swimmer of the Meet

Name	Avg_Place	AA_Diff_Avg	State
Hu, Ethan	1.0	-0.05	CA
Aikins, Jack	1.0	-0.04	GA
Magahey, Jake	1.0	-0.04	GA
Dillard, Ben	1.5	-0.04	CA
Lee, Connor	2.5	-0.03	CA

Ethan Hu, from California, is the boys swimmer of the meet. He's still in California, now swimming for the Stanford Cardinal. Jack Aikins and Jake Magahey, both from Georgia, also won two events apiece and where just behind Ethan, pipped on the All-American tie-breaker.

```
Results_Final %>%
  filter(Name == "Hu, Ethan") %>%
  select(Place, Name, School, Finals_Time, Event) %>%
  arrange(desc(Event)) %>%
  ungroup() %>%
  gt() %>%
  tab_header(title = md("**Ethan Hu Results**"))
```

Ethan Hu Results

Place	Name	School	Finals_Time	Event
1	Hu, Ethan	Harker_CCS	1:45.44	Boys 200 Yard IM
1	Hu, Ethan	Harker_CCS	45.72	Boys 100 Yard Butterfly

Girls

```
Swimmer_Of_Meet[[2]] %>%
  slice_head(n = 5) %>%
  select(-Gender) %>%
  ungroup() %>%
  gt() %>%
  tab_header(title = md("**Girls Swimmer of the Meet**"))
```

Girls Swimmer of the Meet

Name	Avg_Place	AA_Diff_Avg	State
Hartman, Zoie	1.0	-0.05	CA
Ristic, Ella	1.0	-0.02	CA
Delgado, Anicka	1.5	-0.02	CA
Tuggle, Claire	2.0	-0.03	CA
Kosturos, Sophi	2.0	-0.02	CA

Zoie Hartman, representing California is the girls swimmer of the meet. Interestingly enough she now represents Georgia, specifically the University of, and won the the 100 and 200 yard breaststrokes for the Dawgs at SECs in 2020. Ella Ristic of California was also a dual event winner. She now swims for Indiana.

```
Results_Final %>%
  filter(Name == "Hartman, Zoie") %>%
  select(Place, Name, School, Finals_Time, Event) %>%
```

```
arrange(desc(Event)) %>%
ungroup() %>%
gt() %>%
tab_header(title = md("**Zoie Hartman Results**"))
```