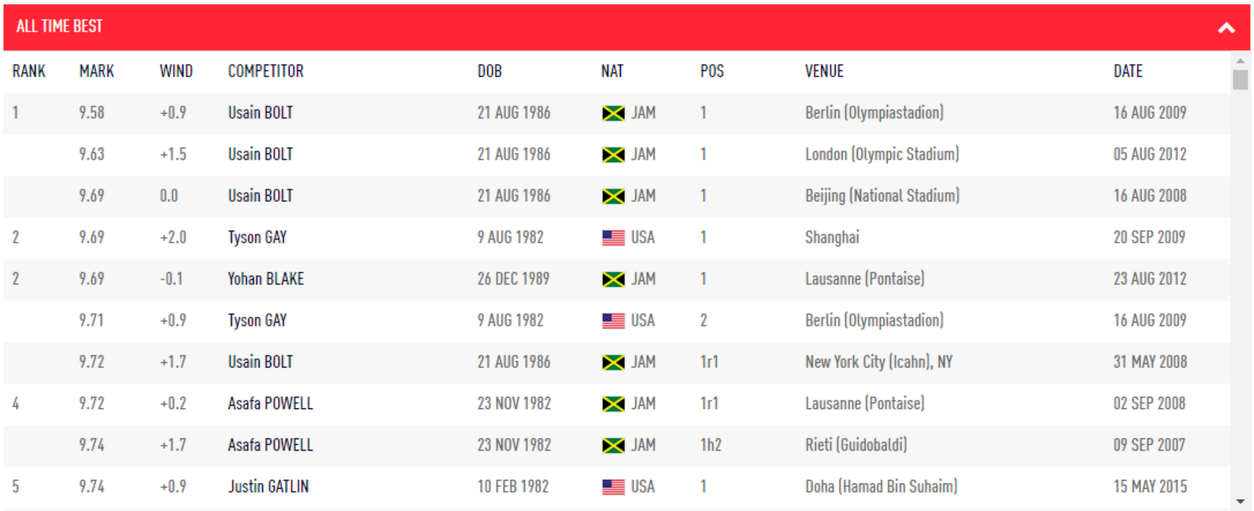
Max Wagner

Data 608 Final – Status Report

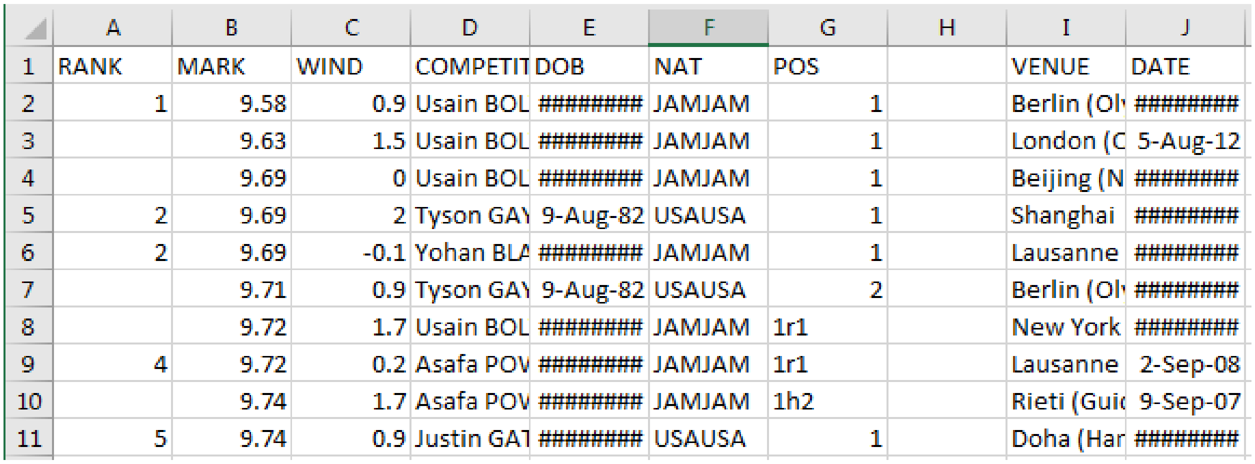
The data was pulled from the source at,www.iaaf.org, which had the initial appearance of a web table.



ALL TIME BEST

RANK	MARK	WIND	COMPETITOR	DOB	NAT	POS	VENUE	DATE
1	9.58	+0.9	Usain BOLT	21 AUG 1986	JAM	1	Berlin (Olympiastadion)	16 AUG 2009
	9.63	+1.5	Usain BOLT	21 AUG 1986	JAM	1	London (Olympic Stadium)	05 AUG 2012
	9.69	0.0	Usain BOLT	21 AUG 1986	JAM	1	Beijing (National Stadium)	16 AUG 2008
2	9.69	+2.0	Tyson GAY	9 AUG 1982	USA	1	Shanghai	20 SEP 2009
2	9.69	-0.1	Yohan BLAKE	26 DEC 1989	JAM	1	Lausanne (Pontaise)	23 AUG 2012
	9.71	+0.9	Tyson GAY	9 AUG 1982	USA	2	Berlin (Olympiastadion)	16 AUG 2009
	9.72	+1.7	Usain BOLT	21 AUG 1986	JAM	1r1	New York City (Icahn), NY	31 MAY 2008
4	9.72	+0.2	Asafa POWELL	23 NOV 1982	JAM	1r1	Lausanne (Pontaise)	02 SEP 2008
	9.74	+1.7	Asafa POWELL	23 NOV 1982	JAM	1h2	Rieti (Guidobaldi)	09 SEP 2007
5	9.74	+0.9	Justin GATLIN	10 FEB 1982	USA	1	Doha (Hamad Bin Suhaim)	15 MAY 2015

It was then pulled directly into a csv file, which had the initial appearance:



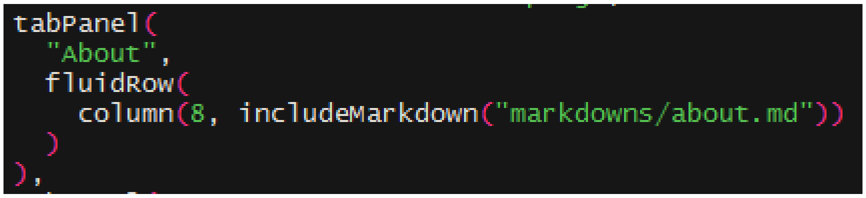
	A	B	C	D	E	F	G	H	I	J
1	RANK	MARK	WIND	COMPETIT	DOB	NAT	POS		VENUE	DATE
2	1	9.58	0.9	Usain BOL	#########	JAMJAM	1		Berlin (Ol	#########
3		9.63	1.5	Usain BOL	#########	JAMJAM	1		London (C	5-Aug-12
4		9.69	0	Usain BOL	#########	JAMJAM	1		Beijing (N	#########
5	2	9.69	2	Tyson GAY	9-Aug-82	USAUSA	1		Shanghai	#########
6	2	9.69	-0.1	Yohan BLA	#########	JAMJAM	1		Lausanne	#########
7		9.71	0.9	Tyson GAY	9-Aug-82	USAUSA	2		Berlin (Ol	#########
8		9.72	1.7	Usain BOL	#########	JAMJAM	1r1		New York	#########
9	4	9.72	0.2	Asafa POV	#########	JAMJAM	1r1		Lausanne	2-Sep-08
10		9.74	1.7	Asafa POV	#########	JAMJAM	1h2		Rieti (Guic	9-Sep-07
11	5	9.74	0.9	Justin GAT	#########	USAUSA	1		Doha (Har	#########

The data was cleaned using the following two R files: 1 and 2.

The first cleaning file removes excess columns, excess times, fixes whitespace issues, trims extra letters and numbers from entries, renames athletes, and finally combines all event csv files into one large csv. The Stringi package was helpful here to fix my capitalizing of names issue.

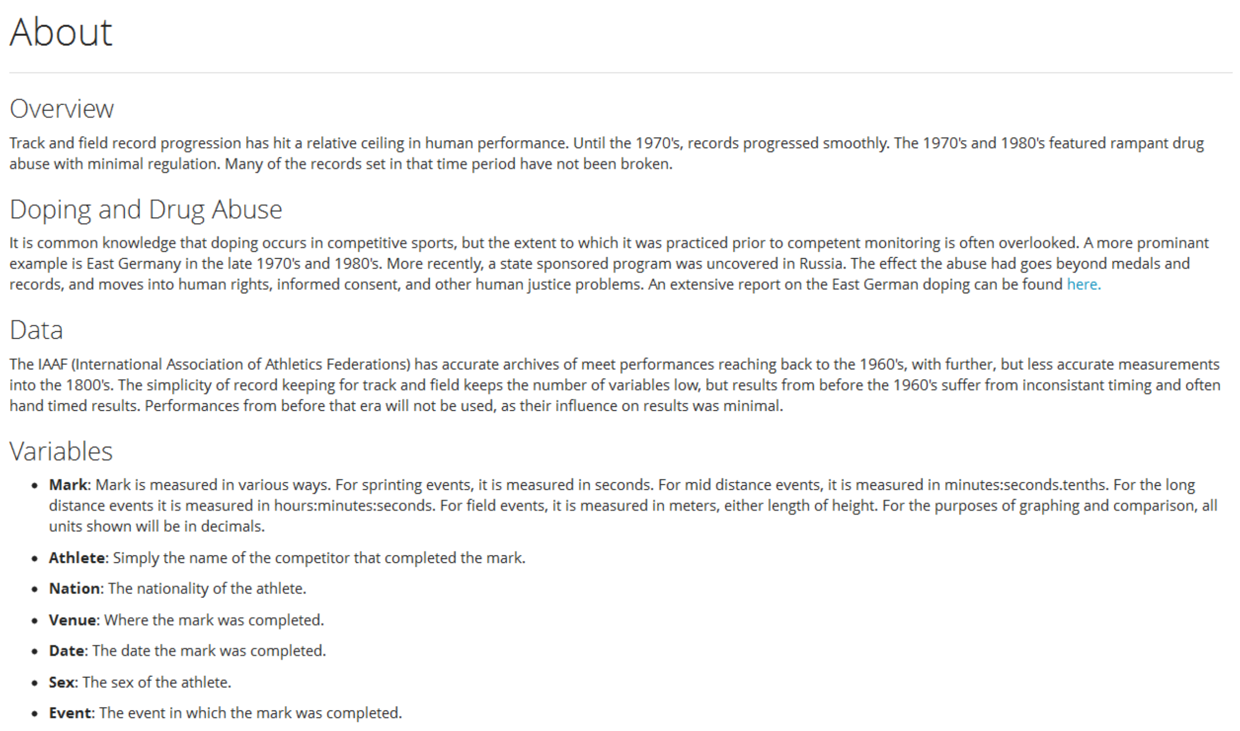
The second cleaning file splits the data set by type of event. This was done because the scoring is done differently depending on the event type. Each event type was then transformed accordingly. For example, the mid distance events are recorded in the format (mm:ss.dd). This was converted to a decimal format. For example, the time (02:30.00) would be converted to 2.5 minutes. This was done for all event types.

The next step was to create a shell of a shiny app to house everything I wanted to display. I knew I had separate pages I wanted to use, so I tried out tab panels in shiny. I outsourced the actual text to different markdown files to keep the main ui.R file cleaner looking.



```
tabPanel(
  "About",
  fluidRow(
    column(8, includeMarkdown("markdowns/about.md"))
  )
),
```

The first page I wrote up was the 2-3 paragraph summary of what was going on in the shiny app and some descriptions of the variables, and what I did to get them to that point.



About

Overview

Track and field record progression has hit a relative ceiling in human performance. Until the 1970's, records progressed smoothly. The 1970's and 1980's featured rampant drug abuse with minimal regulation. Many of the records set in that time period have not been broken.

Doping and Drug Abuse

It is common knowledge that doping occurs in competitive sports, but the extent to which it was practiced prior to competent monitoring is often overlooked. A more prominant example is East Germany in the late 1970's and 1980's. More recently, a state sponsored program was uncovered in Russia. The effect the abuse had goes beyond medals and records, and moves into human rights, informed consent, and other human justice problems. An extensive report on the East German doping can be found here.

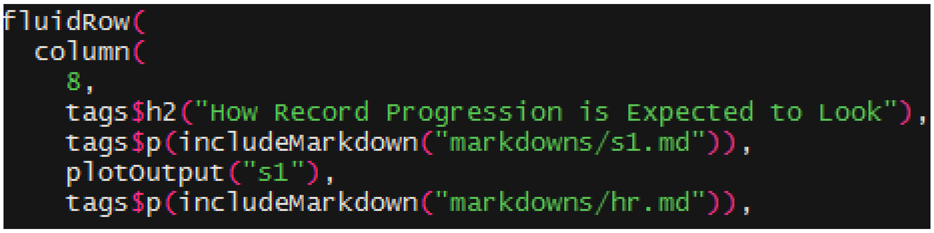
Data

The IAAF (International Association of Athletics Federations) has accurate archives of meet performances reaching back to the 1960's, with further, but less accurate measurements into the 1800's. The simplicity of record keeping for track and field keeps the number of variables low, but results from before the 1960's suffer from inconsistant timing and often hand timed results. Performances from before that era will not be used, as their influence on results was minimal.

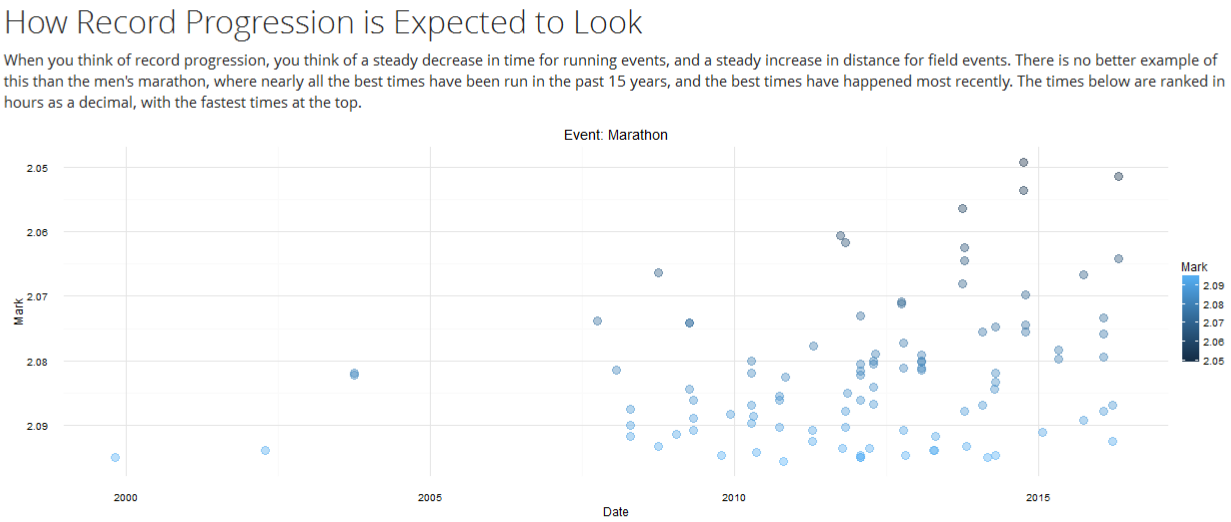
Variables

- **Mark**: Mark is measured in various ways. For sprinting events, it is measured in seconds. For mid distance events, it is measured in minutes:seconds.tenths. For the long distance events it is measured in hours:minutes:seconds. For field events, it is measured in meters, either length of height. For the purposes of graphing and comparison, all units shown will be in decimals.
- **Athlete**: Simply the name of the competitor that completed the mark.
- **Nation**: The nationality of the athlete.
- **Venue**: Where the mark was completed.
- **Date**: The date the mark was completed.
- **Sex**: The sex of the athlete.
- **Event**: The event in which the mark was completed.

I then made a few static graphs to get a feel for the look I was going for on the next page. I ended up using a style sheet and some other options to get ggplot2 to blend into the page and have a “sleeker” look than it does normally. Fluid rows, columns, and the tags function were especially helpful to get everything to fit nicely, and adjust when the page resized.



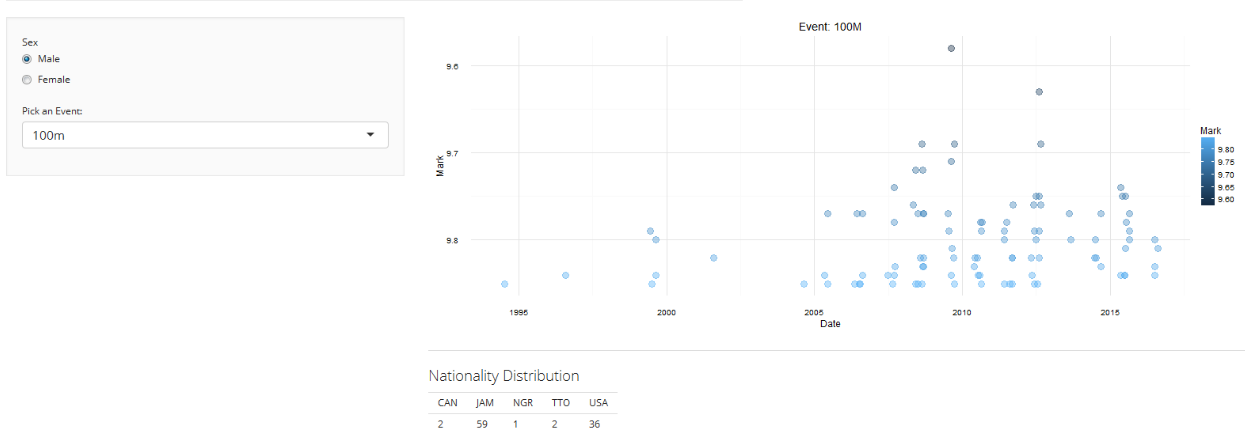
```
fluidRow(
  column(
    8,
    tags$h2("How Record Progression is Expected to Look"),
    tags$p(includeMarkdown("markdowns/s1.md")),
    plotOutput("s1"),
    tags$p(includeMarkdown("markdowns/hr.md")),
```



How Record Progression is Expected to Look

When you think of record progression, you think of a steady decrease in time for running events, and a steady increase in distance for field events. There is no better example of this than the men's marathon, where nearly all the best times have been run in the past 15 years, and the best times have happened most recently. The times below are ranked in hours as a decimal, with the fastest times at the top.

The final part of the project was certainly the hardest. Loading every possible combination of records into a graph and table view. I considered a single dropdown, but it looked cluttered with the male and female choices. I ended up making sex a radial button choice, and the event a dropdown. It cut the clutter in events down by half.



Sex

● Male

○ Female

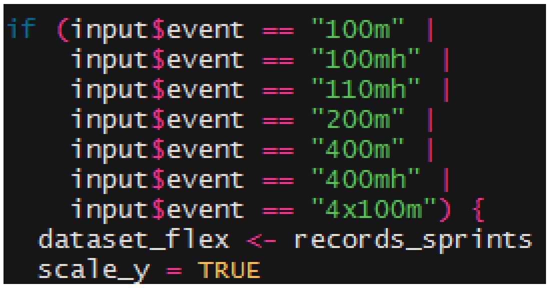
Pick an Event:

100m ▾

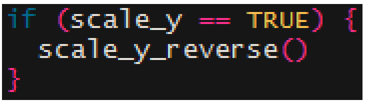
Nationality Distribution

CAN	JAM	NGR	TTO	USA
2	59	1	2	36

The next issue in the graphs I faced was flipping the y-axis for some events, and not for others. I settled on using a Boolean that was chosen in each if statement. Scale_y, and then scale_y_reverse in the graph itself.

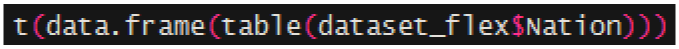


```
if (input$event == "100m" |
    input$event == "100mh" |
    input$event == "110mh" |
    input$event == "200m" |
    input$event == "400m" |
    input$event == "400mh" |
    input$event == "4x100m") {
  dataset_flex <- records_sprints
  scale_y = TRUE
```

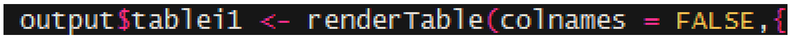


```
if (scale_y == TRUE) {
  scale_y_reverse()
}
```

The small table underneath the graph was a surprisingly annoying bit to produce. The table() function outputs horizontally, but when put into shiny, it was initially viewed vertically, which was a mess to look at. The fix was to wrap the table function in a data frame, and then transpose it. It then printed horizontally, but unfortunately it also printed out ugly variable names. The fix for this was eventually found in the shiny function renderTable, where column and row names can be set to false.



```
t(data.frame(table(dataset_flex$Nation)))
```



```
 output$tablei1 <- renderTable(colnames = FALSE,{
```