An Analysis of D3 Track Times

Alex Puskaric

2023-12-05

Introduction

In Division III athletics, track & field is unique as a sport ruled entirely by data. If an athlete's times are faster than their opponent's, then they are quantifiably better at their sport than the opposition. It is that aspect of track & field that drives the analysis and questions of this paper.

The following questions are asked in the context of men's Division III outdoor track specifically, as it fits the scope of an exploratory analysis. The results of the following questions can lead to more in-depth analysis involving the field component of track & field, the female side of competition, and the insights that can be gained from track teams at the Division I and II levels.

Through our analysis, we hope to answer the following questions: What region excels in D3 track based on various metrics? What school is the best in D3 track? Based on a runner's 100m time, can we predict their 200m time? Based on a runner's 200m time, can we predict their 400m time?

These questions are worth asking for a variety of reasons. As a track athlete, having a model to compare one's times against gives them new insight into their abilities as a runner. If their true 400m time is far slower than their predicted 400m time, then they know that they may be an issue in their race model, or endurance training. If their 400m time is faster than their predicted time, then they may have an issue with their 200m performance that could be looked into. Coaches would benefit from this too, as they could get an estimate of their athlete's, or a recruit's, potential in the 200m or 400m sprints, if they have only run a 100m or a 200m.

Finding the best region for track events in D3 would give insights into what part of the country is the most competitive in D3 track. Narrowing down the selection to specific event groups like the sprints and distance events would then give further insight into where athletes of specific event types are flourishing the most. These results could be used by high school athletes looking for recruitment to find what regions would benefit their personal growth as an athlete the most. The results can also be referenced by coaches and athletic directors to lead into an investigation of why certain regions are so dominant, and what they can change on their end to improve.

Investigating the overall potency of colleges would have similar implications to the investigation of the best region in men's D3 track. Athletes could use the results of the analysis to make informed decisions on where they want to run in college, and other schools will have examples to follow when making changes to their own running programs.

Data

Collection

The data for this project was sourced from TFFRS.com, the official database for reporting collegiate track & field times. Specifically, each region performance list for D3 was web scraped, the data was then cleaned and passed into a data frame that was then saved as a csv. The data collection and web scarping was carried out through the use of the rvest package and the read_html() function to obtain the raw html files for the given track events.

The manipulation of the web addresses required looking into the html code of the webpage itself to see what parameters needed to be entered into the web address to receive the desired data. The number after "/lists/" is the ID number of the specific performance list. The limit is set to 500, which is the maximum number of entries able to be displayed on TFFRS. No region has more than 500 participants in a single event, so there is no need to worry about missing any athletes. The event_type must be set equal to specific values to get the data on the desired event. The web scraper must be run for each event. The values for each event are given below as comments in the code chunk.

For the sake of simplicity, we will be following only the Mid-Atlantic performance list as it is scraped and added to the data frame. Some pieces of code have also been modified so that it can fit into the code chunks, and be readable.

```
# 5,6,7,9,11 = 110h,100,200,400h,400
# 12,13,21,22,19 = 800,1500,5000,10k,3000s

mid_atlantic <- read_html("https://tf.tfrrs.org/lists/4214/
                          DIII_Mid_Atlantic_Outdoor_Performance_List?
                          limit=500&gender=m&event_type=19")
```

Once we have the html file, we convert it to a table, where we add a Region column, and then combine it with the tables made from the other regions.

```
# Convert all htmls to tables
table <- mid_atlantic %>%
  html_element("table") %>%
  html_table() %>%
  select(-c(1)) %>%
  mutate(Region = "Mid-Atlantic")

# Combines all the tables into one table
allTab <- rbind(table, table1, table2, table3, table4,
                table5, table6, table7, table8, table9)
```

Then, we convert the Time column to numeric. The raw times taken from TFFRS are either in seconds format or a minutes:seconds.00 format depending on the event. Converting the times to numeric doesn't format them, so we have to run the times through a for loop that checks them for proper formatting. Any time under 1 minute will be less than 100, and properly formatted, while any time that is at least 1 minute will be at least 100. This requires a 40 second subtraction to get the proper time, in seconds.

```
allTab$Time <- as.numeric(gsub("[^0-9.]", "", allTab$Time))

# If times are over a minute, convert them to seconds
for (i in 1:nrow(allTab)){
  if (allTab$Time[i] >= 6000) {
      allTab$Time[i] = allTab$Time[i] - 2400
  } else if (allTab$Time[i] >= 5900) {
      allTab$Time[i] = allTab$Time[i] - 2360
  } else if (allTab$Time[i] >= 5800) {
      allTab$Time[i] = allTab$Time[i] - 2320
  } else if (allTab$Time[i] >= 5700) {
      allTab$Time[i] = allTab$Time[i] - 2280
  }
  # Goes down to if the time is >= 100, subtract 40
}
```

With conversion complete, we can sort the data frame by the Time column, add an Event column to properly designate the given times. Finally, we can use write.csv() to save the new data frame as a file for later use. In this context, we're saving a list of 3000m steeple chase times.

```
allTab <- arrange(allTab, Time)
allTab$Event <- "3000s"
write.csv(allTab, file = "3000s")
```

The final step is combining all the event data frames into one aggregate list of athletes, then cleaning it for exploration and analysis.

We used rbind() to combine all the event data frames, and used subset on the sprints data frames to remove the Wind column, since the distance events had no such metric.

```
finalTab <- rbind(subset(onem, select = -Wind),
                  subset(twom, select = -Wind),fourm,
                  fourmh,subset(onemh, select = -Wind),
                  eightm,fifteenm,fivek,threes,tenk)
```

The data was not useable in this state, and in the interest of protecting the athletes' identities, we had to introduce a numeric designation. The athlete_mapping data frame takes into account the unique combinations of athlete names and their team to create an ID specific to them.

```
# Create a mapping data frame
athlete_mapping <- finalTab %>%
  distinct(Athlete, Team) %>%
  mutate(Athlete_ID = row_number())
```

We joined the athlete_mapping data frame to the table of data, matching the generated ID to its given athlete. We then dropped the Athlete column, so that their name would not be included in any analysis. We grouped the new data frame by the Athlete_ID, Event, Team, Region, and Year. Then, we used the summarize() function to create a Total_Time column that is the sum of the times in each event for a given athlete. Finally, we used the pivot_wider() function to move all the Event column values to row columns, such that we had columns for each running event in track. The values_from argument moved all the times from the Total_Time column to their designated event column, and then the values_fill argument made sure that any event that an athlete didn't run had an NA as its value.

```
# Map the athletes to their ID, include the necessary rows, pivot
test_set <- finalTab %>%
  left_join(athlete_mapping, by = c("Athlete","Team")) %>%  # Join with athlete mapping
  select(-Athlete) %>%  # Remove the original "Athlete" column
  group_by(Athlete_ID, Event, Team, Region, Year) %>%
  summarize(Total_Time = sum(Time)) %>%
  pivot_wider(names_from = Event, values_from = Total_Time, values_fill = NA)
```

The final step of preparing the data for analysis was manually combing through the data frame for any discrepancies. It's a fringe case, but because some schools have different academic schedules, their athletes may be considered different years depending on the point in the season that they achieve their season best. This results in odd circumstances where some athletes have their times split between two rows. It was such a rare case; however, that it was hard to account for in coding, so it was manually taken care of after the fact.

Explanation

The cleaned and compiled dataset has each row representing a unique male D3 track athlete. There are 7,787 athletes in this data, and it represents every male that ran in the D3 Outdoor 2023 track season. The first column is the Athlete_ID, which is a categorical variable that ranges from 1 to 7,787. The Team column is next, and it is also a categorical variable that designates the team that the athlete belongs to. Region is the third column of the dataset, and is a categorical variable that tracks the region that the athlete belongs to. The regions in D3 are: Mid-Atlantic, East, Great Lakes, West, South, North, Niagara, Midwest, Mideast, and Metro. The Year column is also a categorical variable, and it tracks the athlete's current academic standing. This column is slightly inaccurate; however, since the pandemic has resulted in the NCAA giving many competing juniors and seniors an extra of eligibility. So, while some athletes may appear to be sophomores or juniors, in reality, they could be juniors or seniors. The next ten event columns are all continuous numeric variables that hold an athlete's season best in that event for the 2023 outdoor season. It does not take into account any previous personal bests that the athlete may have had in the 2023 indoor season or previous outdoor season. The events that the dataset contains are the: 100m, 200m, 400m, 400m hurdles, 110m hurdles, 1500m, 800m, 5000m, 10000m, and the 3000m steeple chase. Other events are sometimes run in the outdoor season, but these are the standard events that are typically run at every meet.

The first five rows of the dataset have been included as an example:

```
##   Athlete_ID                   Team  Region Year  100m  200m  400m 400h 110h
## 1          1         Wis.-La Crosse   North SO-2 10.16 20.90    NA   NA   NA
## 2          2 Claremont-Mudd-Scripps    West JR-3 10.30 20.80    NA   NA   NA
## 3          3                 Ramapo   Metro SR-4 10.39 20.49 47.01   NA   NA
## 4          4     East Texas Baptist    West JR-3 10.40 21.02    NA   NA   NA
## 5          5             Greenville Midwest JR-3 10.41    NA    NA   NA   NA
##   1500m 800m 5000m 10000m 3000s
## 1    NA   NA    NA     NA    NA
## 2    NA   NA    NA     NA    NA
## 3    NA   NA    NA     NA    NA
## 4    NA   NA    NA     NA    NA
## 5    NA   NA    NA     NA    NA
```

While the dataset represents every male D3 track athlete, it is also a sample from the larger competitive running community. D3 track is unique in that it has an extremely wide range of talent. Low level D3 runners are comparable to middle school and low level high school track athletes, while the top level athletes are capable of competing with high level D1 and D2 talent. It is for that reason that this specific dataset is suitable for the creation of predictive linear regression models that can be used by trained male runners of all skill levels.

Research Questions

The Best Region in Men's D3 Track

The process of finding the best regions in D3 involves comparing times in events against other events, but this requires some form of standardized metric to enable the comparisons. To account for this, the data must be normalized for the purposes of this question.

To accomplish this, we made a copy of the original dataset, and passed it into a for loop, where we went through each column, checking if it was a numeric variable that was not Athlete_ID. The only other numeric variables in the dataset are the event columns. The entire column was then passed into the preProcess() function, which is a function of the caret library. The method was set to "range", such that we now had a the fastest and slowest times in the specific event processed to be 0 and 1. All other times would be somewhere in between 0 and 1, and could be found with the predict() method. Once the column of standardized times was generated, it replaced the column of normal times, such that we now had values from 0 through 1, where 0 was the fastest time, and 1 was the slowest time.

The Best Overall Region in Men's D3 Track, by Average Performance

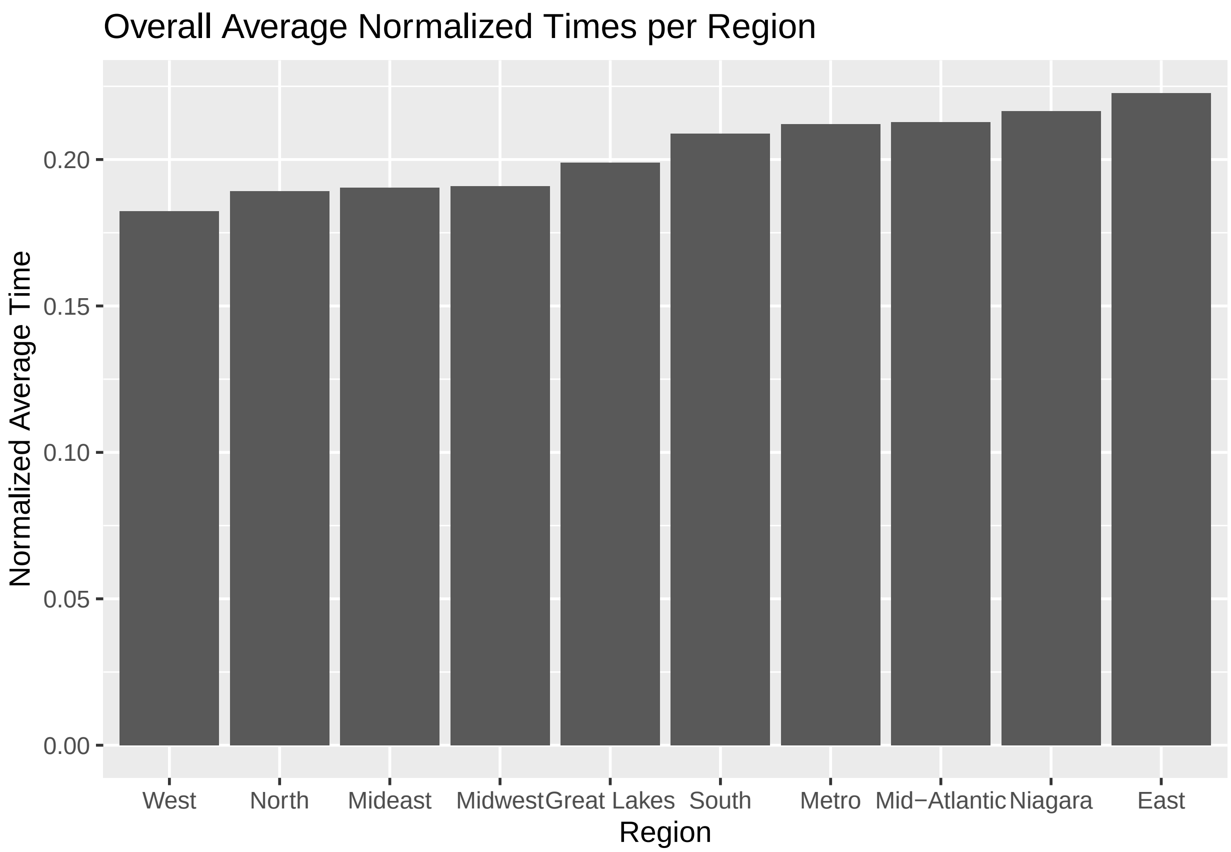
To find the overall best region in D3, we grouped the dataset by region, then found the mean of every single time for every event ran by the athletes of that region. This would give each region an "average performance score" that represented the average running ability of the athletes in the region. Just like running, the lower the score, the better.

```
avg_norm_score_per_region <- norm_data %>% group_by(Region) %>%
  summarize(avg_score = mean(c(`100m`,`200m`,`400m`,
                               `800m`,`1500m`,`5000m`,`10000m`,
                               `110h`,`400h`,`3000s`), na.rm = TRUE)) %>%
  arrange(avg_score)

as.data.frame(avg_norm_score_per_region)
```

```
##          Region avg_score
## 1          West 0.1823221
## 2         North 0.1892817
## 3       Mideast 0.1904356
## 4       Midwest 0.1909830
## 5   Great Lakes 0.1989214
## 6         South 0.2088666
## 7         Metro 0.2120220
## 8  Mid-Atlantic 0.2128074
## 9       Niagara 0.2164931
## 10         East 0.2227533
```

Based on the output, it seems that the West Region has the highest average overall level of competition among men in D3. The North, Mideast, and Midwest Regions are all fairly close together, but noticeably less competitive than the West Region. The Great Lakes, and all other regions have a greater fall off that can be visualized with a bar graph.



Again, the West Region is noticeably more competitive than all the other regions. The North, Mideast, and Midwest Regions are almost indistinguishable from each other in capabilities. Once we get past the first four regions, the fall off is incredibly pronounced. The standouts are the Great Lakes Region which manages to be noticeably better than the bottom five regions in D3, and the East Region which is noticeably lagging behind all other regions in average running ability.

The Best Overall Region in Men's D3 Track, by Top Thirty Times per Event

When investigating what the "best" region is, it is important to consider different metrics. The previous questions found the region with the highest average level of competition, but failed to account for the high level performers in each region. It is with that in mind that we are now only taking into account the thirty fastest times from each region.

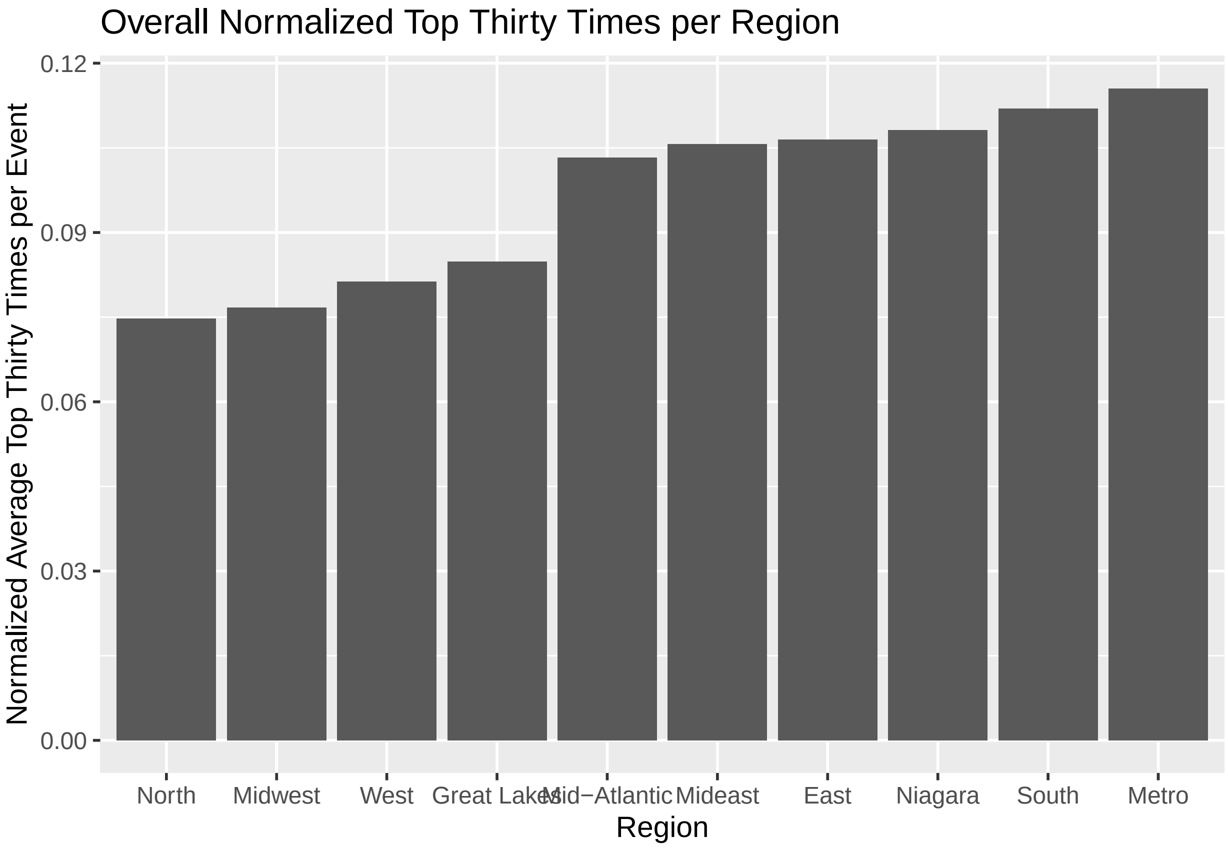
Thirty was chosen because it allows for the given times to fall if there is a noticeable lack of high level talent in the region.

```r
top_thirty_region_data <- norm_data %>% group_by(Region) %>%
  summarize(`100m` = mean(head(sort(`100m`),30),na.rm=TRUE),
            `200m` = mean(head(sort(`200m`),30),na.rm=TRUE),
            `400m` = mean(head(sort(`400m`),30),na.rm=TRUE),
            `110h` = mean(head(sort(`110h`),30),na.rm=TRUE),
            `400h` = mean(head(sort(`400h`),30),na.rm=TRUE),
            `800m` = mean(head(sort(`800m`),30),na.rm=TRUE),
            `1500m` = mean(head(sort(`1500m`),30),na.rm=TRUE),
            `5000m` = mean(head(sort(`5000m`),30),na.rm=TRUE),
            `10000m` = mean(head(sort(`10000m`),30),na.rm=TRUE),
             `3000s` = mean(head(sort(`3000s`),30),na.rm=TRUE)
             ) %>%
   group_by(Region) %>%
   summarize(avg_score = mean(c(`100m`,`200m`,`400m`,
                                `800m`,`1500m`,`5000m`,`10000m`,
                                `110h`,`400h`,`3000s`), na.rm = TRUE)) %>%
  arrange(avg_score)

as.data.frame(top_thirty_region_data)
```

```
##          Region  avg_score
## 1         North 0.07473911
## 2       Midwest 0.07668281
## 3          West 0.08131026
## 4   Great Lakes 0.08488422
## 5  Mid-Atlantic 0.10329637
## 6       Mideast 0.10568881
## 7          East 0.10648160
## 8       Niagara 0.10818012
## 9         South 0.11196617
## 10        Metro 0.11553744
```

When comparing the regions by the top thirty best times, it seems that the North Region is the best in D3, with the Midwest Region, previously listed as the fourth best, taking the second place spot. Again, we can use a bar plot to better visualize the data.



The top four regions are noticeably better than the rest. North, Midwest, West, and Great Lakes are all below 0.09 for the average score. This is in stark contrast to the next best region, Mid-Atlantic, which is slightly above the 0.10 mark. Note that the top three regions for this analysis were in the top four in the previous analysis, and that the Mideast Region fell all the way back to sixth place from its third place spot.

This suggests that while the Mideast Region is as a consistently high level of competition, its best athletes are comparably less competitive than its peers in the top four.

The Best Overall Region in Men's D3 Track, by Top Four Times per Event

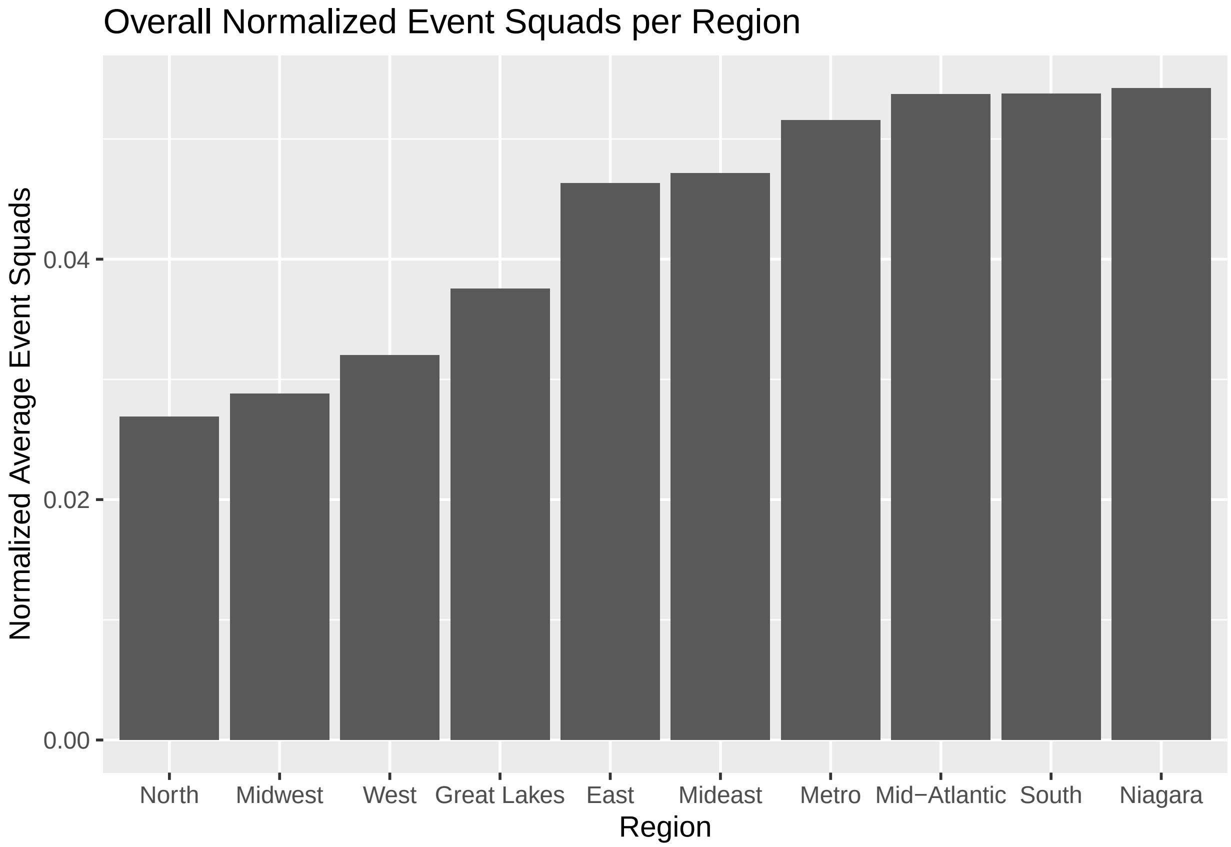
In collegiate track and field, event squads are tracked to measure, and rank a team's potential for one-on-one competition. Event squads consist of the top four athletes of each event on that time. The reason that it's top four is that for collegiate meets, the top eight times score. So, if two teams go head to head with their event squads, a result can easily be deduced by checking the event squads.

That philosophy is being applied to the regions. Their top four best runners in each event are being taken for their average score this time. These are the absolute best of the best in the region, and the answer to this specific question would serve as a predictor for what region would come out on top if they all competed in a track meet against each other.

```
event_squad_region_data <- norm_data %>% group_by(Region) %>%
  summarize(`100m` = mean(head(sort(`100m`),4),na.rm=TRUE),
            `200m` = mean(head(sort(`200m`),4),na.rm=TRUE),
            `400m` = mean(head(sort(`400m`),4),na.rm=TRUE),
            `110h` = mean(head(sort(`110h`),4),na.rm=TRUE),
            `400h` = mean(head(sort(`400h`),4),na.rm=TRUE),
            `800m` = mean(head(sort(`800m`),4),na.rm=TRUE),
            `1500m` = mean(head(sort(`1500m`),4),na.rm=TRUE),
            `5000m` = mean(head(sort(`5000m`),4),na.rm=TRUE),
            `10000m` = mean(head(sort(`10000m`),4),na.rm=TRUE),
            `3000s` = mean(head(sort(`3000s`),4),na.rm=TRUE),
            ) %>%
   group_by(Region) %>%
   summarize(avg_score = mean(c(`100m`,`200m`,`400m`,
                                `800m`,`1500m`,`5000m`,`10000m`,
                                `110h`,`400h`,`3000s`), na.rm = TRUE)) %>%
  arrange(avg_score)
as.data.frame(event_squad_region_data)
```

```
##          Region  avg_score
## 1         North 0.02689337
## 2       Midwest 0.02882396
## 3          West 0.03199483
## 4   Great Lakes 0.03752324
## 5          East 0.04631499
## 6       Mideast 0.04712999
## 7         Metro 0.05155146
## 8  Mid-Atlantic 0.05370218
## 9         South 0.05374845
## 10      Niagara 0.05423729
```

Like the previous result, our top four remains the same, but in a surprising upset, the East Region would be the fifth best region at this hypothetical meet. Compared to its performance in the overall rankings, where it was the worst scoring region, a placement of fifth is a vast improvement. We can use a bar graph to better visualize the improvements.



The top four regions remain dominant in the rankings. The East Region is almost even with the Mideast Region, which in turn is noticeably faster than the bottom four regions. Aside from the bottom six regions, the result is largely the same.

The Best Region for Sprinters in Men's D3 Track, by Average Performance

We will be using the normalized data to answer this question.

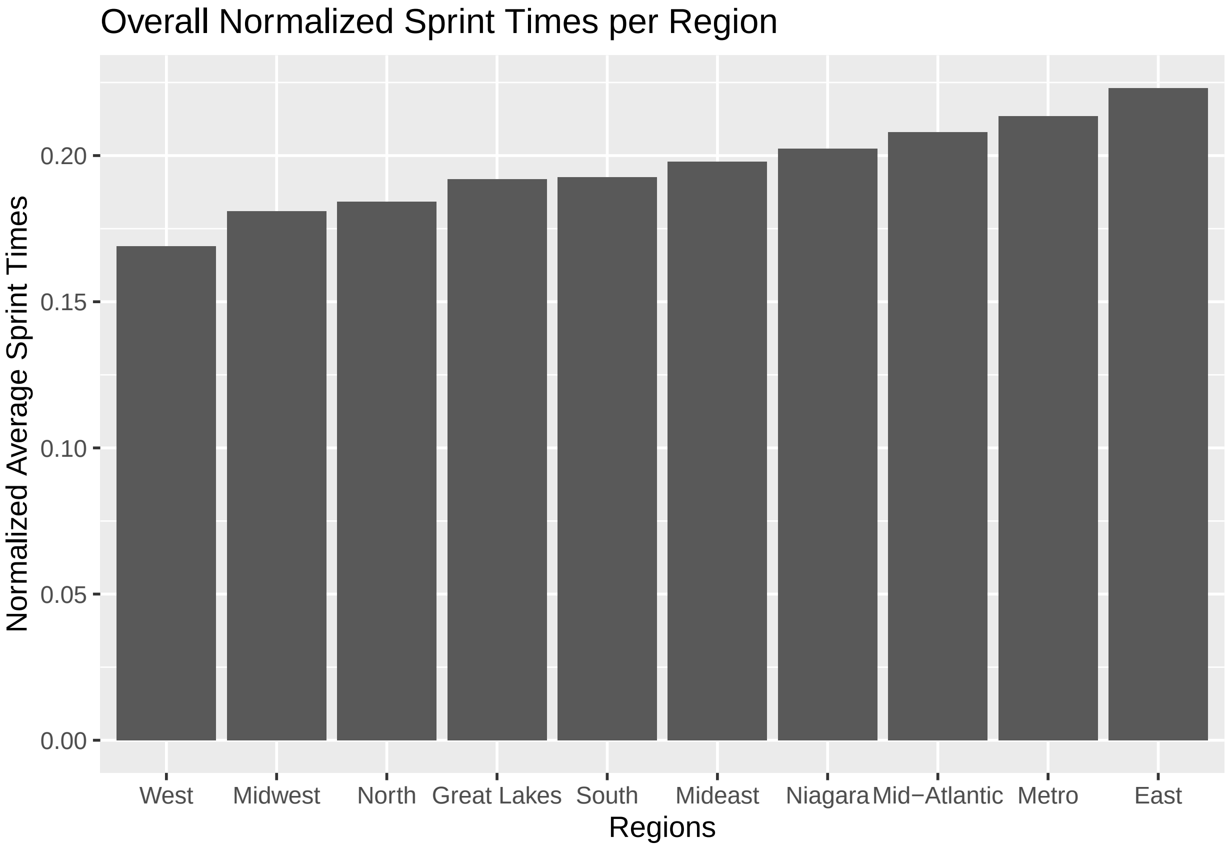
We have investigated the overall abilities of the regions, but it's entirely possible that some regions are heavily biased towards a certain discipline of running, whether it be sprints or distance. It is for that purpose that we will investigate the average performance of all the sprinters in each region.

Like the overall region scores, we are finding the average of all events in the region, except we are only considering sprinting events here. For the sake of simplicity, we are considering the hurdles as a sprinting event. The total list of events is as follows: 100m, 200m, 400m, 110m hurdles, and 400m hurdles.

```r
avg_norm_sprint_score_per_region <- norm_data %>% group_by(Region) %>%
  summarize(avg_score = mean(c(`100m`,`200m`,`400m`,`110h`,`400h`), na.rm = TRUE)) %>%
  arrange(avg_score)
as.data.frame(avg_norm_sprint_score_per_region)
```

```
##          Region avg_score
## 1          West 0.1690350
## 2       Midwest 0.1810151
## 3         North 0.1841696
## 4   Great Lakes 0.1919940
## 5         South 0.1926954
## 6       Mideast 0.1979401
## 7       Niagara 0.2024102
## 8  Mid-Atlantic 0.2079554
## 9         Metro 0.2135499
## 10         East 0.2231527
```

Similar to the overall average times for all events in the regions, the West Region is noticeably better at sprints than the other regions. The East Region remains in last place via this analysis. We can look at a bar graph to get a better understanding of how the regions compare.



The average performance of sprinters in the West Region compared to the rest of D3 is incredibly striking. What is equally striking is the performance of the East Region being noticeably worse than the rest of D3. The Midwest, North, and Great Lakes regions, which have been mainstays in the top five at varying spots remain as strong as ever.

The Best Region for Distance Runners in Men's D3 Track, by Average Performance

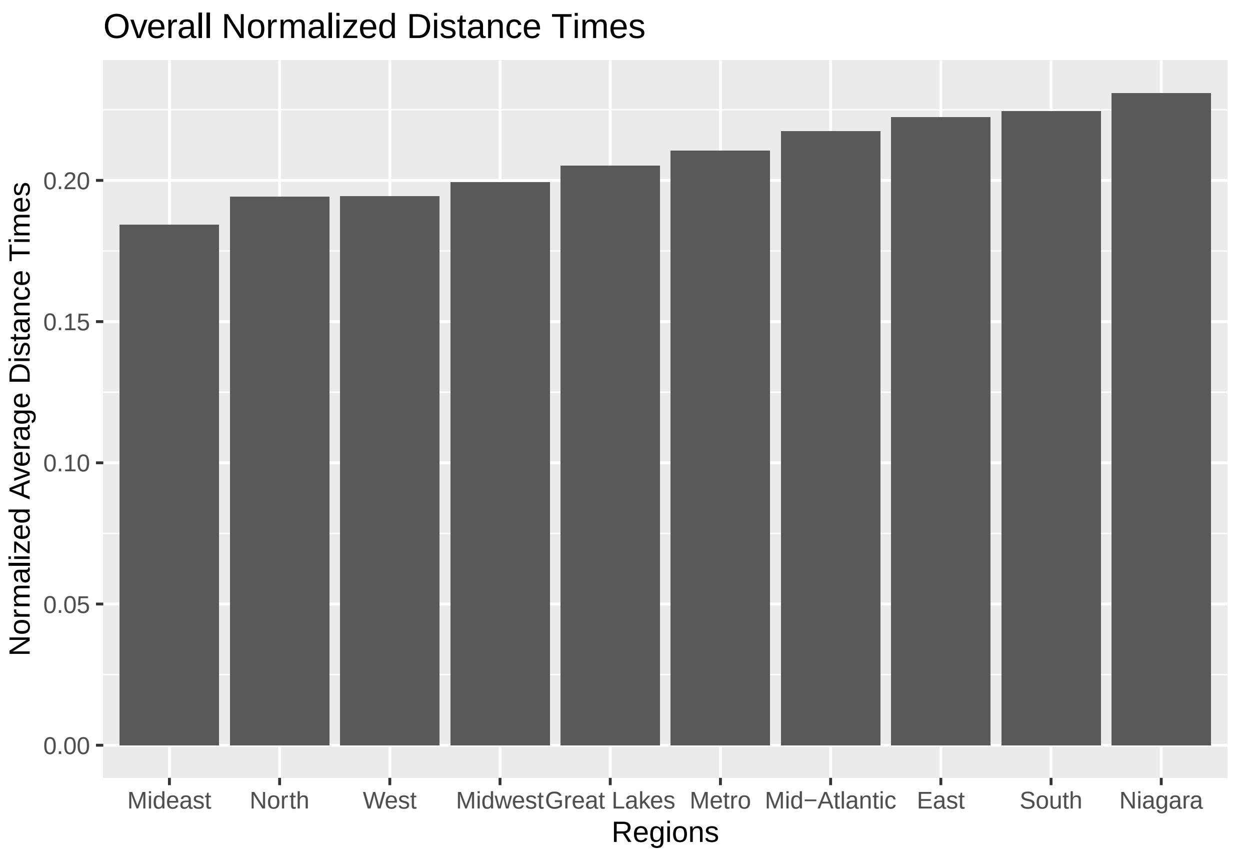
This is similar to the previous question, except we are finding the average of all distance events in the region. These include the: 800m, 1500m, 5000m, 10000m, and the 3000m steeple chase.

```
avg_norm_distance_score_per_region <- norm_data %>% group_by(Region) %>%
  summarize(avg_score = mean(c(`800m`,`1500m`,`5000m`,`10000m`,`3000s`), na.rm = TRUE)) %>%
  arrange(avg_score)
as.data.frame(avg_norm_distance_score_per_region)
```

```
##          Region avg_score
## 1       Mideast 0.1843928
## 2         North 0.1941906
## 3          West 0.1944305
## 4       Midwest 0.1993214
## 5   Great Lakes 0.2052701
## 6         Metro 0.2105324
## 7  Mid-Atlantic 0.2173620
## 8          East 0.2223291
## 9         South 0.2245483
## 10      Niagara 0.2309853
```

The Mideast Region appears to have the best distance runners, on average, in men's D3. The Mideast Region has ranged from middle of the pack to top three in D3 depending on the criteria set for the analysis. So, it makes sense that given a specific category, it could have the chance to excel. Through the use of a bar graph, we can better visualize the differences between the regions.

```
ggplot(avg_norm_distance_score_per_region, aes(x=reorder(Region, avg_score), y=avg_score)) +
  geom_col() +
  xlab("Regions") +
  ylab("Normalized Average Distance Times") +
  ggtitle("Overall Normalized Distance Times")
```



The overall quality of distance runners in the Mideast Region is far more noticeable here. The North, West, and Midwest Regions, which were previously dominant, are noticeably lacking when compared to the Mideast Region. Though, they are still superior to the rest of D3.

The Best Overall School in Men's D3 Track, by Average Performance

This question will be answered with the normalized dataset.

The process is similar to the previous sections where we found the average performance of each region. The main difference is that we are grouping by Team and Region when finding the mean. This is to account for the possibility of there being schools with the same name in D3. We are also only using schools with more than twenty posted times in the season. This is to avoid any schools that have a low number from interfering with the results.

```
avg_norm_score_per_team <- norm_data %>% filter(n() > 20) %>%
  group_by(Team,Region) %>%
  summarize(avg_score = mean(c(`100m`,`200m`,`400m`,
                               `800m`,`1500m`,`5000m`,`10000m`,
                               `110h`,`400h`,`3000s`), na.rm = TRUE)) %>%
  arrange(avg_score) %>%
  head(10)
```

```
## ‘summarise()‘ has grouped output by ’Team’. You can override using the
## ‘.groups‘ argument.
```

```
as.data.frame(avg_norm_score_per_team)
```

```
##                        Team       Region  avg_score
## 1                       MIT         East 0.09189626
## 2            Wis.-La Crosse        North 0.09557828
## 3                 Lynchburg        South 0.11676320
## 4  Claremont-Mudd-Scripps         West 0.12226071
## 5             Washington U.      Midwest 0.12467834
## 6                     Emory        South 0.12524908
## 7                     Rowan        Metro 0.12549745
## 8           Carnegie Mellon Mid-Atlantic 0.12665353
## 9                  Fredonia      Niagara 0.13078361
## 10            John Carroll  Great Lakes 0.13181085
```

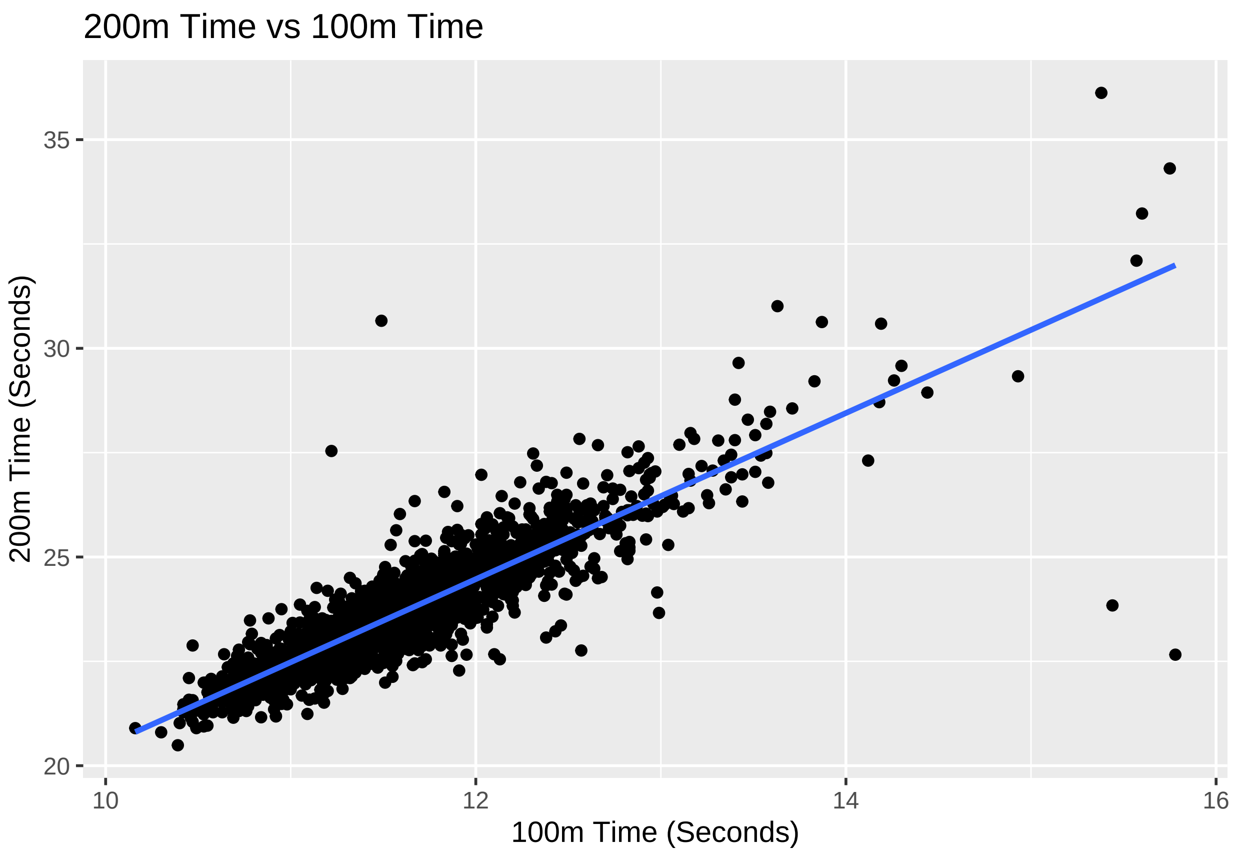
Interestingly, MIT, which is in the East Region, a low performing region, is the best school in D3 when going off of average performances. All regions besides the Mideast Region are represented by the top ten best overall schools in D3, with the South Region being represented twice by Emory and Lynchburg.

We will not be creating a visualization here because the total number of schools in D3 is too large to properly convey in a graph in this format.

Predicting an Athlete's 200m Time from their 100m Time

When considering the relationship between an athlete's 100m time and their 200m time, it is important to first investigate how they interact with each other. So, let's make a graph of all 200m times vs their corresponding 100m times to visualize that relationship.

```
data %>% filter(!is.na(`100m`) & !is.na(`200m`)) %>%
  ggplot(aes(x=`100m`, y=`200m`)) +
  geom_point() +
  geom_smooth(method = "lm", se=FALSE) +
  ggtitle("200m Time vs 100m Time") +
  xlab("100m Time (Seconds)") +
  ylab("200m Time (Seconds)")
```



The resulting graph shows that there is in fact some kind of positive correlation between a runner's 100m time and their 200m time. We have observed that there is a linear relationship between the two events. Thus, we can calculate the correlation coefficient.

```
cor(data$`100m`, data$`200m`, use = "complete.obs")
```

```
## [1] 0.8817864
```

The result is 0.8817, which suggests a significant positive correlation between an athlete's 100m time and their 200m time. So, we can move forward, and create a linear regression model of the 200m time as a function of 100m time.

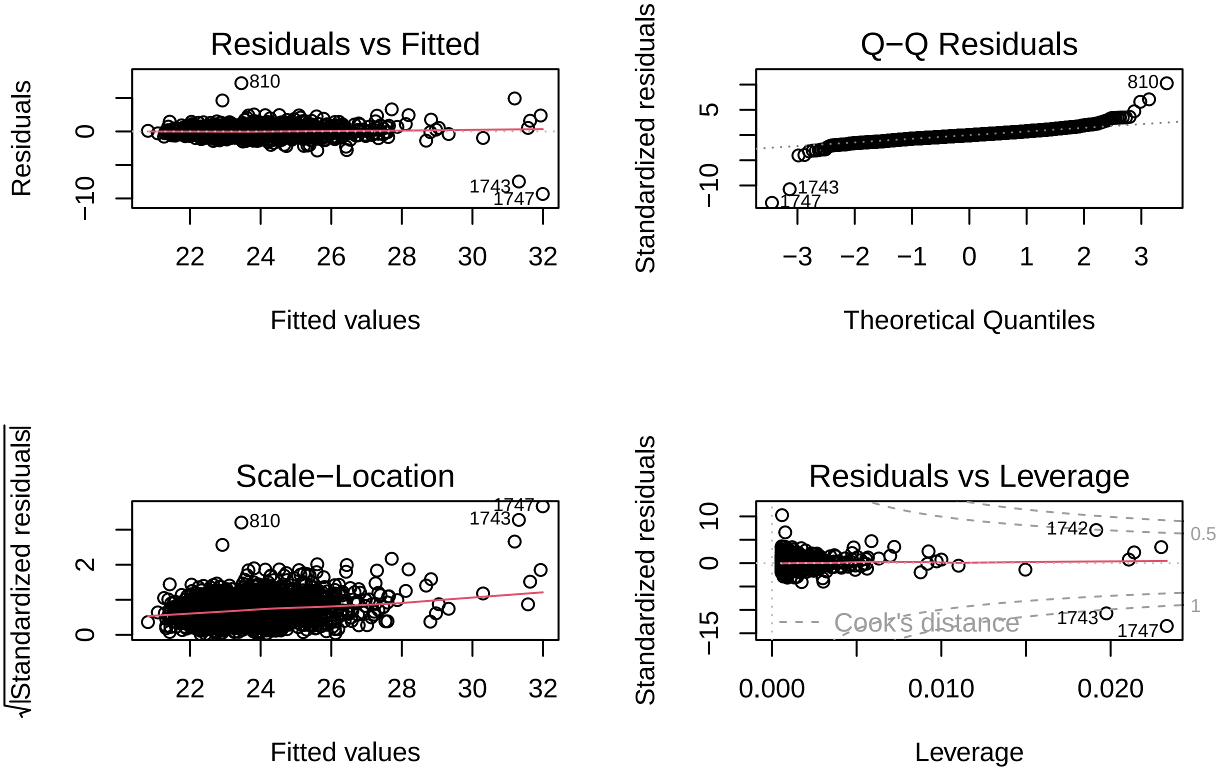
```
lm_100m_to_200m <- lm(`200m` ~ `100m`, data %>% filter(!is.na(`100m`) & !is.na(`200m`)))
lm_100m_to_200m
```

```
##
## Call:
## lm(formula = `200m` ~ `100m`, data = data %>% filter(!is.na(`100m`) &
##     !is.na(`200m`)))
##
## Coefficients:
## (Intercept)       `100m`
##      0.5871       1.9902
```

The resulting equation is $y = 0.5871 + 1.9902x$, where y is the predicted 200m time, and x is the given 100m time.

Since this is an inferential question, where we are using the D3 data as a sample for the entire running community. We need to analyze the residuals to determine if this model is valid for predictions.

```r
par(mfrow=c(2,2))
plot(lm_100m_to_200m)
```



The line is flat for the Residuals vs Fitted graph, with most of the fitted values being spread between 22 and 28. The model passes the linearity test. The qq-plot shows that the model follows the line very closely, only diverting at the very ends, implying that it may be lightly tailed to a slight degree. The Scale-Location graph has the line trending slightly upwards, but the residuals appear to be randomly clustered between 22 and 28. The Residuals vs Leverage plot shows three striking outliers. For future iterations of this model, these outliers would be removed, but for now let's continue with them intact.

The assumption of linearity holds, and while the residuals slightly deviate from a normal distribution at the extremes, exhibits slight indications of heteroscedasticity, and has outliers with high leverage, we will continue with the current model to make inferences on the greater running community. The problems in the plots are not egregious enough to completely shift course. So, we will proceed with the observed problems in mind.

Before making predictions, let's check the model one more time. We can use the summary() function to find the r^2 value of the model to determine how good of a fit it is for the data.

```r
summary(lm_100m_to_200m)
```

```
##
## Call:
```

```
## lm(formula = ‘200m‘ ~ ‘100m‘, data = data %>% filter(!is.na(‘100m‘) &
##     !is.na(‘200m‘)))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.3325 -0.3534 -0.0294  0.3326  7.2054
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.58711    0.29665   1.979    0.048 *
## ‘100m‘       1.99020    0.02548  78.098   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7026 on 1745 degrees of freedom
## Multiple R-squared:  0.7775, Adjusted R-squared:  0.7774
## F-statistic:  6099 on 1 and 1745 DF,  p-value: < 2.2e-16
```

The r^2 value is 0.7774, indicating that the model is indeed a good fit for the data. Additionally, our p-value is 2e-16. So, given an α of 0.05, we can conclude that the 100m time variable is a significant predictor for the 200m time.

Confident that our model is a good fit for the data, we can now make some prediction intervals for the possible 200m times given any 100m time.

```r
predict(lm_100m_to_200m, data.frame(`100m` = 12, check.names=FALSE),
           interval="prediction", level=.95)
```

```
##        fit      lwr      upr
## 1 24.46956 23.09105 25.84806
```

Given a confidence interval of 0.95, and a 100m time of 12 seconds, the resulting 200m times range from [23.09, 25.85]. The prediction itself is a time of 24.47 seconds. This range of values is wider than expected, but the given interval could be used to make some assumptions about an athlete's capabilities and race model.

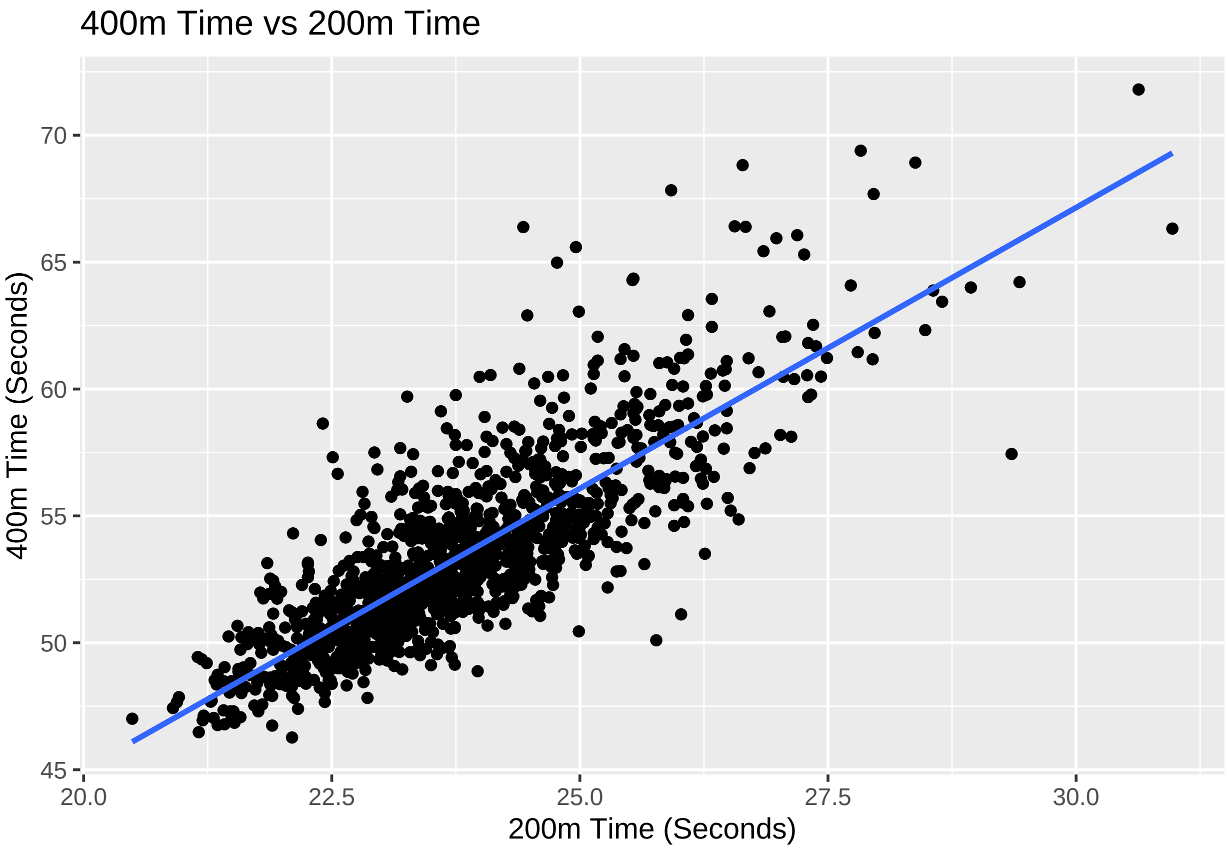
For example, given a time of 12 seconds, we now know that an athlete is likely to fall within the range of [23.09, 25.85] for the 200m time. If their true 200m time falls between the fit and the upper range, then it raises flags that the athlete's speed endurance could be lacking, and they can't maintain their speed in the later half of their race. If the inverse happens, where their true 200m time falls between the fit and the lower range, then it could mean that the athlete has impressive top end speed, but struggles with their acceleration.

So, given our linear regression model, it can be used to predict an athlete's potential 200m time as the fit. Then, via the lower and upper ranges, it gives a range of times that an athlete has a chance of hitting, given their 100m time. This range of values can be used to better understand the relationship between an athlete's 100m time, and a preexisting 200m time. Doing so can lead to to a better understanding of the athlete theirself, and their possible strengths and weaknesses as a runner.

Predicting an Athlete's 400m Time from their 200m Time

Just like the previous section, we will first create a scatter plot of 400m time vs 200m time from the data to see if there is any observable linear relationship between the variables.

```
data %>% filter(!is.na(`200m`) & !is.na(`400m`)) %>%
  ggplot(aes(x=`200m`, y=`400m`)) +
  geom_point() +
  geom_smooth(method = "lm", se=FALSE) +
  ggtitle("400m Time vs 200m Time") +
  xlab("200m Time (Seconds)") +
  ylab("400m Time (Seconds)")
```



The plot suggests that there may be a positive linear relationship between 400m time and 200m time. We can investigate this relationship further by calculating the correlation coefficient of the two variables.

```
cor(data$`200m`, data$`400m`, use = "complete.obs")
```

```
## [1] 0.8217493
```

The correlation coefficient is 0.8217, which suggests a significant positive correlation between 400m time and 200m time. So, we can move forward and create a linear regression model of the 400m time as a function of 200m time.

```
lm_200m_to_400m <- lm(`400m` ~ `200m`, data %>% filter(!is.na(`200m`) & !is.na(`400m`)))
lm_200m_to_400m
```

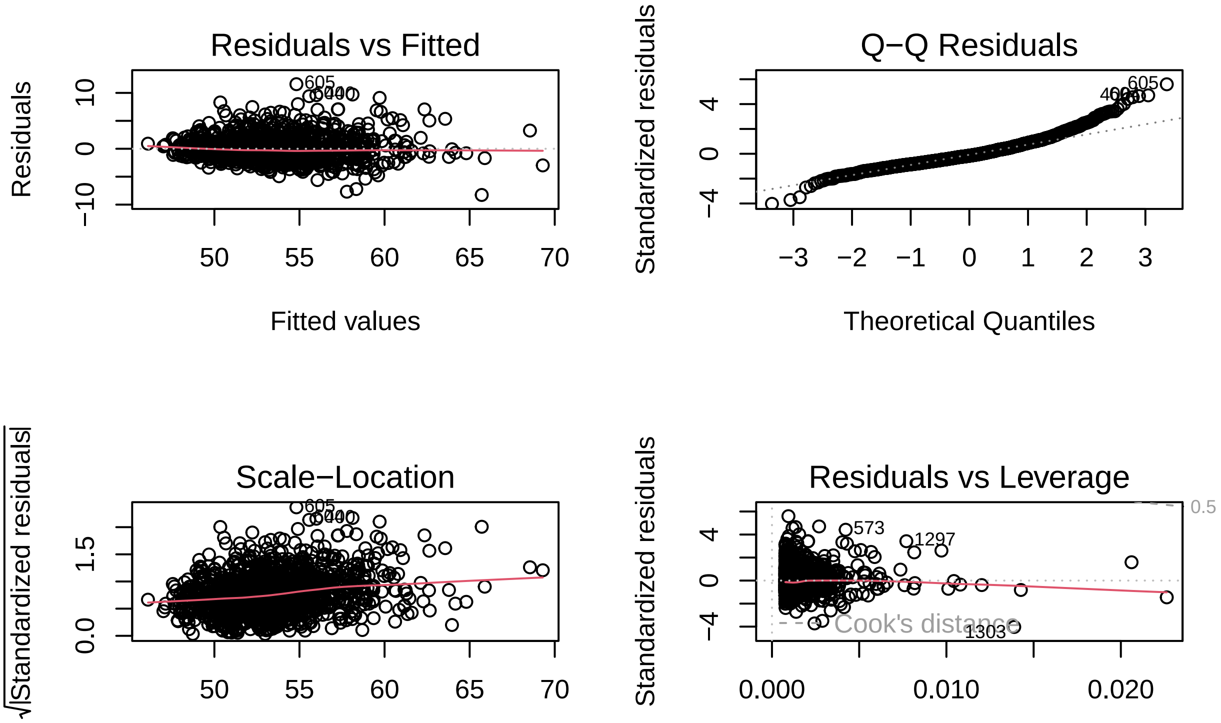
```
##
## Call:
```

```
## lm(formula = '400m' ~ '200m', data = data %>% filter(!is.na('200m') &
##     !is.na('400m')))
##
## Coefficients:
## (Intercept)       '200m'
##      0.7429       2.2135
```

The resulting equation is $y = 0.7429 + 2.2135x$, where y is the predicted 400m time, and x is a given 200m time.

Since this is an inferential question, where we are using the D3 data as a sample for the entire running community. We need to analyze the residuals to determine if this model is valid for predictions.

```
par(mfrow=c(2,2))
plot(lm_200m_to_400m)
```



The line is flat for the Residuals vs Fitted graph, with most of the fitted values being spread between 50 and 60. The model passes the linearity test. The qq-plot shows that the model follows the line closely for the most part, but it is diverting from normalcy more noticeably after the theoretical quantile of 1. This implies that the residuals may not be normal. The Scale-Location graph has the line trending slightly upwards, but the residuals appear to be randomly clustered between 50 and 60. The Residuals vs Leverage plot shows no egregious outliers like the previous model.

The assumption of linearity holds. The residuals are observably not normal. There might be some slight heteroscedasticity in the model. There are no major outliers that have been observed, however. The problems with this model have been noted, and with them in mind, we will proceed.

Before making predictions, let’s check the model one more time. We can use the summary() function to find the r^2 value of the model to determine how good of a fit it is for the data.

```
summary(lm_200m_to_400m)
```

```
##
## Call:
## lm(formula = ‘400m‘ ~ ‘200m‘, data = data %>% filter(!is.na(‘200m‘) &
##     !is.na(‘400m‘)))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.2704 -1.3055 -0.3193  0.9760 11.5603
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.74294    1.01287   0.734    0.463
## ‘200m‘       2.21354    0.04252  52.054   <2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 2.068 on 1303 degrees of freedom
## Multiple R-squared:  0.6753, Adjusted R-squared:  0.675
## F-statistic:  2710 on 1 and 1303 DF,  p-value: < 2.2e-16
```

The r^2 value is 0.675, indicating that the model is a good fit for the data, albeit less so than the previous one. Additionally, our p-value is 2e-16. So, given an α of 0.05, we can conclude that the 200m time variable is a significant predictor for the 400m time.

Confident that our model is a good fit for the data, we can now make some prediction intervals for the possible 400m times given any 200m time.

```
predict(lm_200m_to_400m, data.frame(`200m` = 24.57, check.names=FALSE),
           interval="prediction", level=.95)
```

```
##        fit      lwr      upr
## 1 55.12964 51.07152 59.18775
```

Given a confidence interval of 0.95, and a 200m time of 14.57 seconds, the resulting 400m times range from [51.07, 59.19] seconds. The prediction itself is a time of 55.13 seconds. This range of values is even wider than the previous model’s, but the conclusions that can be drawn from it are still worth investigating.

For example, given a time of 24.57 seconds, we now know that an athlete is likely to fall within the range of [51.07, 59.19] for the 400m. In a similar manner to the previous manner, we can assume that if an athlete’s true time falls between the fit and an upper bound, then they are lacking in overall fitness, and possibly cannot maintain their speed for a long period of time. The inverse, where the athlete’s true time is between the lower range and fit, could imply that an athlete has incredible fitness, and that they need to increase their speed.

Based on prior knowledge of 400m runners, mid-distance runners who participate in the 800m would commonly find themselves between the lower range and fit if they were to use their 200m time as the input. Short sprinters using this model would then find themselves on the other end of the spectrum, since they commonly train the 100m and 200m, with little focus on endurance.

Given our linear regression model, it can be used to predict an athlete's potential 400m time as the fit. Then, via the lower and upper ranges, it gives a range of times that an athlete has the potential of hitting, given their 200m time. This range of values can also serve as a source of comparison against an athlete's true 400m time to draw conclusions on the athlete's specific strengths and weaknesses as a runner.

Conclusion

Given the analysis done on the regions of men's D3 track, it is safe to say that there are three distinct regions that are noticeably superior to the other seven regions. The North, Midwest, and West Regions have shown themselves to be competitive at an overall level, and outright dominant when looking at just their event squad and top thirty times in every event. In the sprints, the story was the same. While the three regions lagged behind the Mideast Region in the overall performance for distance events, they were still superior to the rest of the regions. This dominance requires further investigation from a different angle. We know what regions and schools are the best in D3, but not why. Since it is the schools that recruit and train the runners, it is important to gain an understanding of what separates nationally competitive D3 schools from the average ones. Future research of this topic would revolve around that idea, where aspects of the schools themselves, such as acceptance rate, tuition, academic scholarships, and other characteristics to see if there are any non-athletic factors that affect what schools high-tier track athletes attend.

Furthermore, there may be some worth in expanding the analysis of the regions' proficiency in the sprints and distance events to the same "top thirty" and "event squad" analyses that were carried out for every event in the regions simultaneously. It would be interesting to see if the Mideast Region retains its number one rankings in distance events when observing only its highest level competitors. One final route we can investigate is including the field events, like the jumps and throws, into the dataset so that we can find the team that is truly the best in D3.

Moving onto the linear regression models, we have established during the residual analysis for both models that there are some problems that need addressed to improve the validity of both models. For the 100m to 200m model we need to address the extreme outliers that were observed on the Residuals vs Leverage graph. We can create a second model without the outliers, and see if the accuracy of our predictions improve. Additionally, it is clear that after 28 for the fitted values on the Residuals vs Fitted and Scale-Location graphs, that the data points become more scarce, and less clustered. Creating another model where we only take 200m times faster than 28 seconds may lead to an improved model.

For the 200m to 400m model, we observed that there are issues with its normalcy and its heteroskedasticity. A possible angle to go for fixing these flaws of the model is to transform the variables via logarithms to see if we can create a more robust prediction model.

Aside from the observed flaws with the models, we have successfully created two models that when given a time in either the 100m or 200m, can predict not only an athlete's time in the 200m or 400m, but can also generate a range of possible times that they can hit. This range of times, while helpful for showing the potential times for an athlete, also serves as a means for deeper analysis into an athlete's capabilities. When used by an experienced runner, or a coach, one could take their 100m or 200m personal record, and create a range of their potential times in the 200m or 400m. If they have run that event, then they can compare their times to the range of values to see if they are in the lower or upper range of times. Given their placement in the range, they can then consider the possible factors for their placement, and adjust their training plans as they see fit.

Expansion of this model would involve including more data for the model to be trained on. While D3 does have high level competition, the top tier athletes in D2 and D1 are noticeably faster. Given this fact, introducing the athletes from these divisions into the model may create a prediction model tailored more towards high performance athletes that are looking for insights into their running ability. An entirely new model can also be created for the female athletes of D3, following the same steps taken to create the current models. Linear regression models for predicting event times from other event times can also be made.

Specifically, we could investigate the relationship between distance events, since that is a running discipline that was neglected in the later stages of this report.