Trail Running Race Results Analysis - 2020 Speedgoat 28k

Ryan Hansen

9/22/2020

Introduction

Whether you know me or not, it's probably obvious I love to run in the mountains. Often, I compete in trail running races in Utah and around the country. On July 25th, 2020, I competed in my toughest race yet: Speedgoat 28k at Snowbird Mountain Resort (UT). While there is also a 50k option, the 28k was plenty for me with nearly 6,000 feet of elevation gain.

Sponsored by HOKA ONE ONE, Speedgoat is a well known event in the trail running community. Every year it draws strong competition from around the United States. The race has even attracted international talent such as the sensational, Kilian Jornet. But, due to the COVID-19 outbreak, this year's event included only domestic athletes.

The following is an exploration and analysis of the 2020 Speedgoat 28k race results data (source: http://ultrasignup.com/results_event.aspx?did=72676). Primarily, the goal of this project was to fulfill my curiosity about the distributions of race participant data points and how they may influence performance in the event (ultimately, finishing place). This project also allowed me to practice R skills in a real-world scenario.

It's worth quickly noting that 20 scheduled race participants did not start the race for reasons unknown (likely many due to COVID-19). These instances were not included in the data frame as performing analysis on them was not of interest to me.

Loading required packages and race results CSV file

```
library(tidyverse)
library(psych)

#Loading the CSV file into R and assigning it to the "sg" object
sg <- read.csv("Speedgoat 28k Data Visualization - Speedgoat 28k.csv")
```

Reviewing the basics

First, let's check out the structure of our newly created data frame.

```
str(sg)
```

```
## 'data.frame':    82 obs. of  10 variables:
##  $ Place    : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ First    : chr  "Jeshurun" "Timmy" "Charles" "Adrian" ...
##  $ Last     : chr  "Small" "Parr" "MacNulty" "Gonzalez" ...
##  $ City     : chr  "Golden" "Leadville" "San Francisco" "Pomoma" ...
##  $ State    : chr  "CO" "CO" "CA" "CA" ...
##  $ Age      : int  22 38 46 31 18 29 27 51 24 31 ...
##  $ Gender   : chr  "M" "M" "M" "M" ...
##  $ GP       : int  1 2 3 4 5 1 6 7 8 9 ...
##  $ Time.Rank: chr  "2:50:46" "2:52:12" "3:21:51" "3:27:48" ...
##  $ Rank     : num  88.5 94.3 89.3 94.5 71.3 ...
```

Looks like we have 82 race participants and 10 variables for each.

Let's also review a summary of the data frame.

```
summary(sg)
```

```
##      Place          First               Last              City
##  Min.   : 0.00   Length:82          Length:82          Length:82
##  1st Qu.:18.25   Class :character   Class :character   Class :character
##  Median :38.50   Mode  :character   Mode  :character   Mode  :character
##  Mean   :38.54
##  3rd Qu.:58.75
##  Max.   :79.00
##     State                Age           Gender               GP
##  Length:82          Min.   :14.00   Length:82          Min.   : 0.00
##  Class :character   1st Qu.:29.25   Class :character   1st Qu.: 9.25
##  Mode  :character   Median :34.00   Mode  :character   Median :19.50
##                     Mean   :36.13                      Mean   :20.85
##                     3rd Qu.:42.75                      3rd Qu.:29.75
##                     Max.   :81.00                      Max.   :50.00
##   Time.Rank             Rank
##  Length:82          Min.   : 0.00
##  Class :character   1st Qu.:55.13
##  Mode  :character   Median :67.16
##                     Mean   :66.78
##                     3rd Qu.:76.69
##                     Max.   :96.74
```

Now we'll dive in and make a few adjustments before beginning the analysis.

Removing DNFs from the data frame

When looking at the summary of our 'sg' object above, I notice that the Place variable has a minimum value of 0. Because Place references what order runners finished the race in, this doesn't make sense. But, I know from experience in running that it's common for at least some participants to "DNF" ("Did Not Finish"), particularly in more difficult races such as Speedgoat. These cases are probably being given a value of 0 for Place. Let's verify that.

```
count(filter(sg, Place == 0))
```

```
##   n
## 1 3
```

Perfect. The code above allowed us to filter for these instances and see there were three people who did not finish. We'll remove them from the data frame for this analysis. If we had a bigger dataset, it may be interesting to look into stats of DNF racers vs. finishers.

Based on our data structure review earlier, we know the non-finishers must be the last three instances of our data frame. Let's remove those instances from our main data frame.

```
sg <- sg[-c(80:82),]

#Rechecking the data frame for the change
summary(sg)
```

```
##      Place         First               Last               City
##  Min.   : 1.0   Length:79          Length:79          Length:79
##  1st Qu.:20.5   Class :character   Class :character   Class :character
##  Median :40.0   Mode  :character   Mode  :character   Mode  :character
##  Mean   :40.0
##  3rd Qu.:59.5
##  Max.   :79.0
##     State                Age           Gender                GP
##  Length:79          Min.   :14.00   Length:79          Min.   : 1.00
##  Class :character   1st Qu.:29.50   Class :character   1st Qu.:10.50
##  Mode  :character   Median :34.00   Mode  :character   Median :20.00
##                     Mean   :35.86                      Mean   :21.65
##                     3rd Qu.:42.00                      3rd Qu.:30.50
##                     Max.   :81.00                      Max.   :50.00
##   Time.Rank             Rank
##  Length:79          Min.   :43.00
##  Class :character   1st Qu.:55.95
##  Mode  :character   Median :67.32
##                     Mean   :67.56
##                     3rd Qu.:76.07
##                     Max.   :96.74
```

Perfect, 1 is now our minimum value and 79 is our max value for Place.

Converting some character variables to factor and time variables

By default, R has categorized several variables as character variables. This is fine and preferred for variables like First and Last names, but since we'll want to use a few of these categorical variables for analysis, we'll need to convert them to factor variables (plus a time variable in the case of Time.Rank).

```
sg$City <- factor(sg$City)
sg$State <- factor(sg$State)
sg$Gender <- factor(sg$Gender)
sg$Time.Rank <- strptime(sg$Time.Rank, format = "%H:%M:%S")

#Confirming the conversions above
summary(sg)
```

```
##      Place         First               Last                          City
##  Min.   : 1.0   Length:79          Length:79          Salt Lake City    :17
##  1st Qu.:20.5   Class :character   Class :character   Cottonwood Heights: 3
##  Median :40.0   Mode  :character   Mode  :character   Lehi              : 3
##  Mean   :40.0                                         Denver            : 2
##  3rd Qu.:59.5                                         Ogden             : 2
##  Max.   :79.0                                         Park City         : 2
##                                                       (Other)           :50
##     State          Age        Gender       GP
##  UT     :46   Min.   :14.00   F:29   Min.   : 1.00
##  CA     :11   1st Qu.:29.50   M:50   1st Qu.:10.50
##  CO     : 9   Median :34.00          Median :20.00
##  AZ     : 3   Mean   :35.86          Mean   :21.65
##  WA     : 3   3rd Qu.:42.00          3rd Qu.:30.50
##  TX     : 2   Max.   :81.00          Max.   :50.00
##  (Other): 5
##    Time.Rank                      Rank
##  Min.   :2020-09-23 02:50:46   Min.   :43.00
##  1st Qu.:2020-09-23 04:11:46   1st Qu.:55.95
##  Median :2020-09-23 05:04:31   Median :67.32
##  Mean   :2020-09-23 05:07:34   Mean   :67.56
##  3rd Qu.:2020-09-23 06:00:29   3rd Qu.:76.07
##  Max.   :2020-09-23 07:29:48   Max.   :96.74
##
```

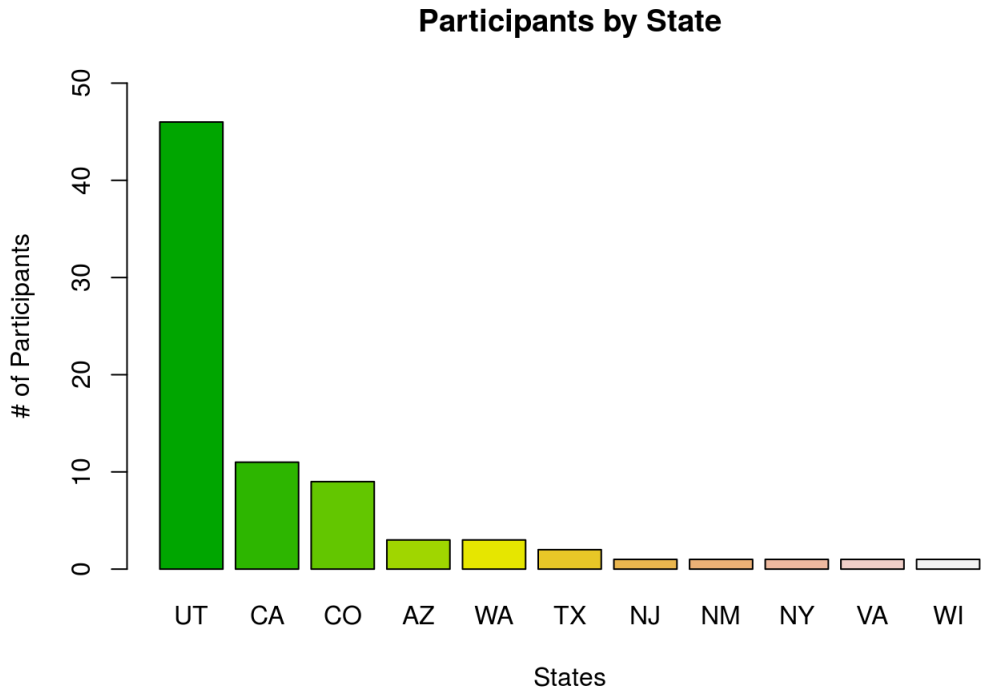
Good. That's a much better looking data frame.

Looking at distributions of variables

Now, let's explore how some of our variables are distributed.

Participants by State

```
counts <- table(sg$State)
barplot(sort(counts,decreasing = TRUE), main = "Participants by State", ylim = c(0, 50),
xlab = "States", ylab = "# of Participants", col = terrain.colors(11))
```

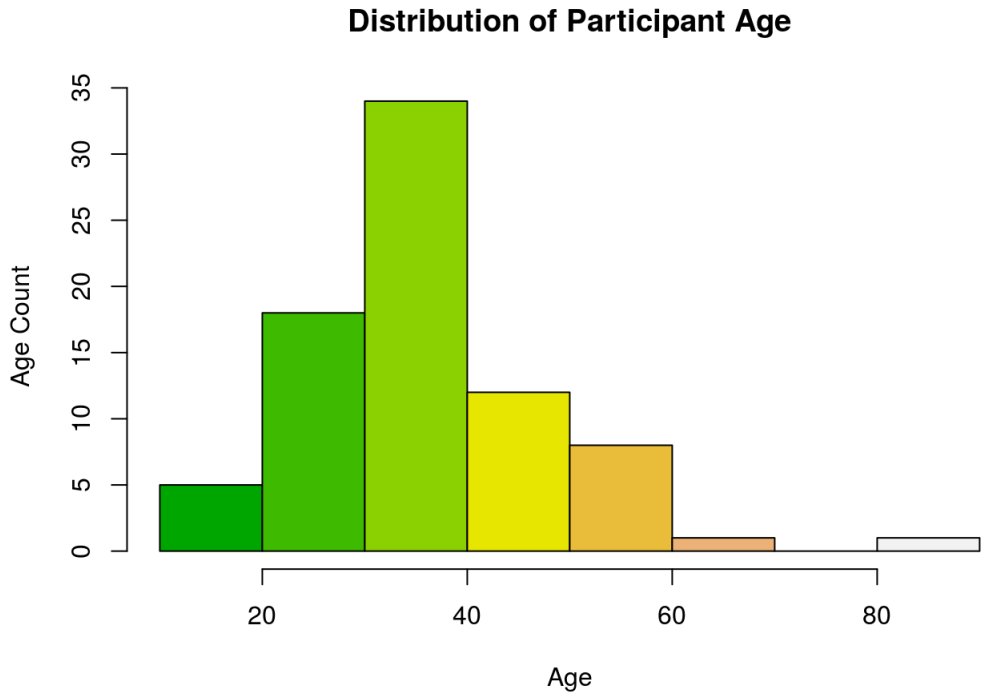


Unsurprisingly, the vast majority of race participants call Utah home. Representation is also higher from nearby states such as California, Colorado and Arizona versus east coast states like New Jersey and New York.

In addition to our dataset being small, the off-balanced nature of it is a very important call out. Any assumptions made or conclusions drawn need to be prefaced by this bias in the dataset.

Distribution of Age

```
hist(x=sg$Age, xlab = "Age", ylab = "Age Count", main = "Distribution of Participant Age",
col = terrain.colors(8))
```



Our participant age data is partially right-skewed but follows a fairly normal distribution. One of the interesting facts about trail running is that it truly can be a lifetime sport. As you can see, there are participants in their teens up to (unbelievably) 81. I can confirm the latter isn't an error because I saw the gentleman during the race! With that outlier, the mean of Age is 35.86 (as seen in the summary of our data set above).

```
#Checking the mean Age of the data set with the outlier removed
mean(sg$Age[sg$Age<81])
```

```
## [1] 35.28205
```

With the outlier removed, the mean is about the same at 35.28.

Finally, let's review the exact breakdown of ages in terms of counts and percentages.

```
#Creating a table displaying counts of each age in the data set
age_table <- table(sg$Age)
age_table
```

```
##
## 14 18 19 22 23 24 26 27 28 29 30 31 32 33 34 35 36 37 38 40 41 42 43 44 45 46
##  1  1  3  2  1  3  2  2  1  4  3  4  6  5  5  3  5  2  2  2  1  2  3  2  1  1
## 47 50 51 52 53 54 55 56 61 81
##  1  1  2  1  1  1  2  1  1  1
```

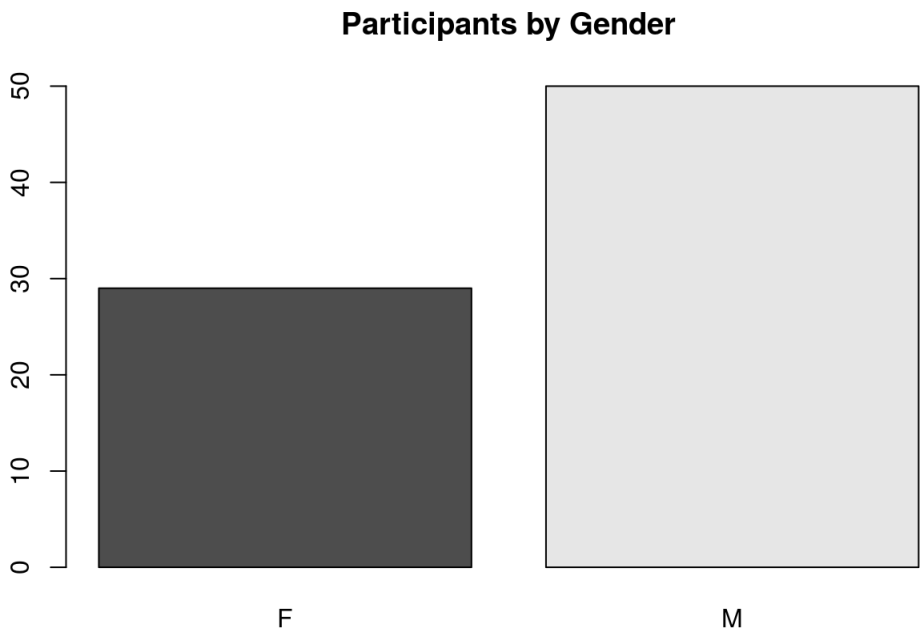
```
#Creating a percentage version of the above table
round(prop.table(age_table), digits = 2)
```

```
##
##   14   18   19   22   23   24   26   27   28   29   30   31   32   33   34   35
## 0.01 0.01 0.04 0.03 0.01 0.04 0.03 0.03 0.01 0.05 0.04 0.05 0.08 0.06 0.06 0.04
##   36   37   38   40   41   42   43   44   45   46   47   50   51   52   53   54
## 0.06 0.03 0.03 0.03 0.01 0.03 0.04 0.03 0.01 0.01 0.01 0.01 0.03 0.01 0.01 0.01
##   55   56   61   81
## 0.03 0.01 0.01 0.01
```

Looks like our most common age is 32 (8% of total participants).

Distribution of Gender

```
barplot(table(sg$Gender), main = "Participants by Gender", col = gray.colors(2))
```




```
gender_table <- table(sg$Gender)
round(prop.table(gender_table), digits = 2)
```

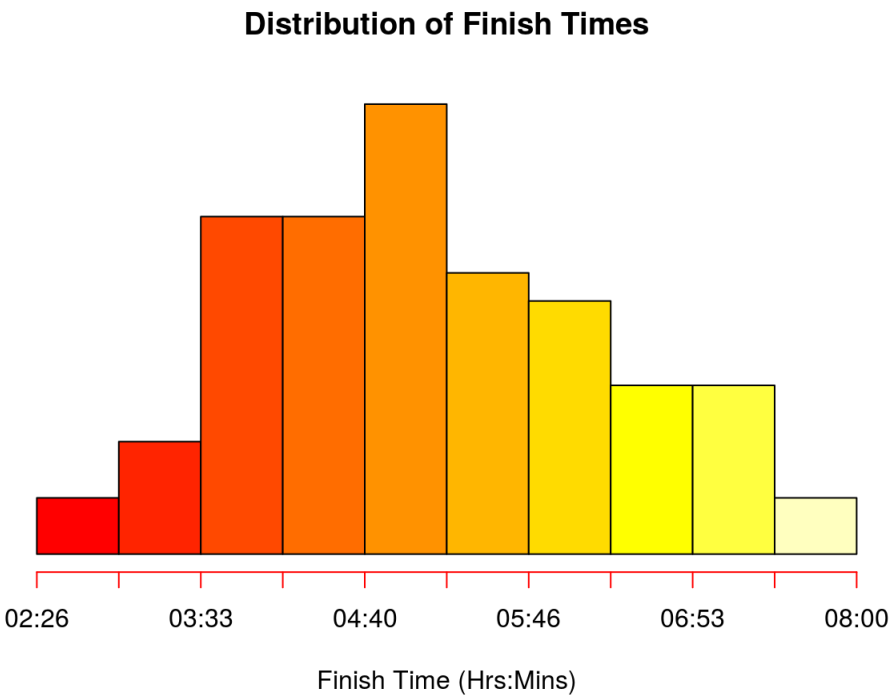
```
##
##    F    M
## 0.37 0.63
```

It appears that around two thirds of race participants were male while a third were female. We'll look at gender a little more later in this analysis.

Distribution of Finishing Times

```
hist(x=sg$Time.Rank, xlab = "Finish Time (Hrs:Mins)", ylab ="", yaxt='n',
main = "Distribution of Finish Times", col = heat.colors(10), breaks = 8)
```

```
## Warning in breaks[-1L] + breaks[-nB]: NAs produced by integer overflow
```

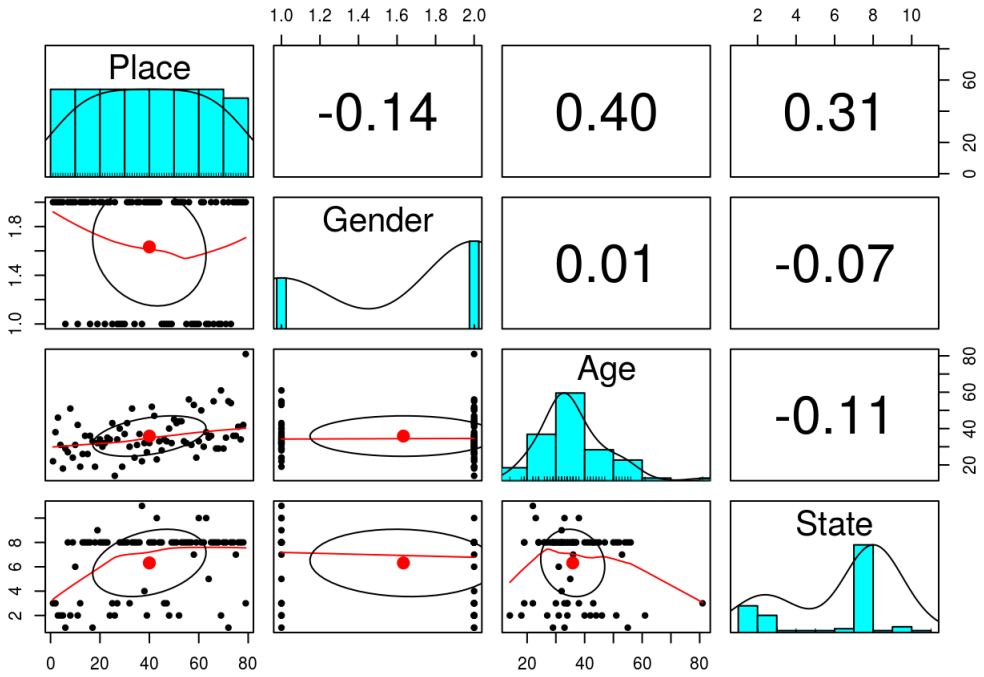


This is a very balanced distribution with some incredible standouts finishing in under three hours. In a demanding and long event like Speedgoat, this type of distribution is quite common with all types of skill/experience levels competing. Top finishers crossed the finish line in less than half the time of those toward the right-tail end of the distribution.

Exploring relationships among variables

Relationships of numeric variables

```
pairs.panels(sg[c("Place","Gender","Age","State")])
```



Above I've used the pairs.panels function which is part of the Psych package. It's a bit overwhelming to take in at first, but it's an incredibly powerful visualization (particularly for small data sets where all crucial variables can be examined at once). Essentially, we're able to see a histogram of each variable plus a bivariate scatter plot and correlation coefficient for each combination of two variables.

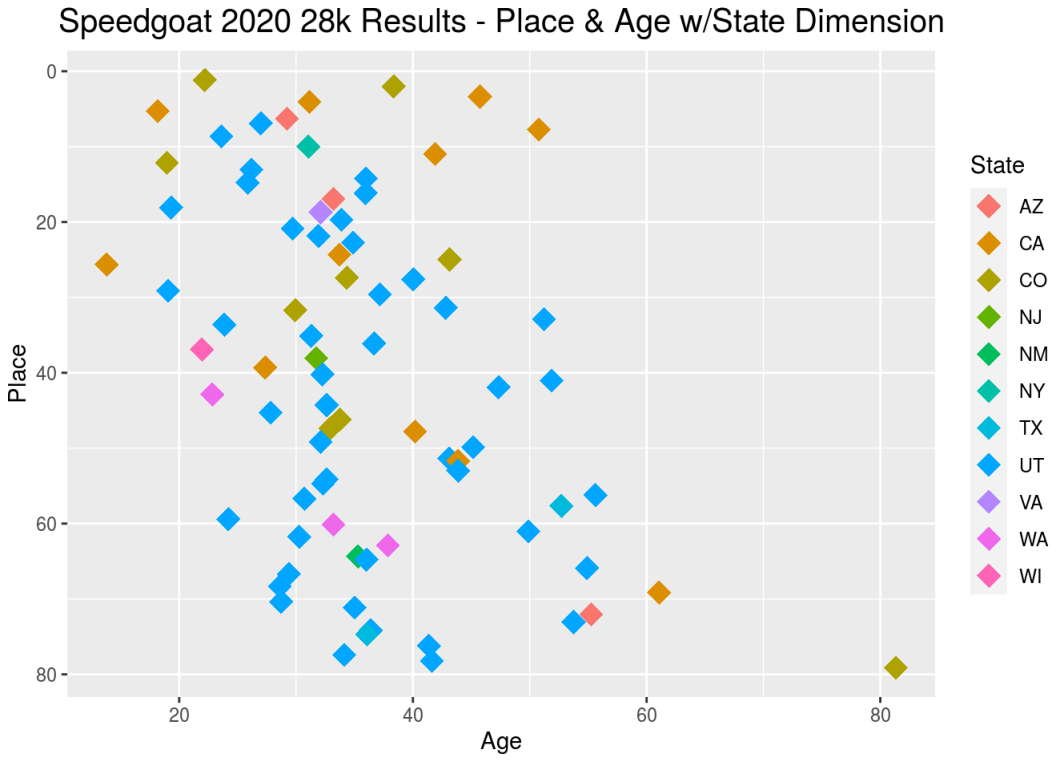
What I want to focus on is two correlation coefficients. Place and Age have the strongest correlation among selected variables of 0.41. This makes complete intuitive sense. As age increases, naturally your finishing place will be higher (worse in this case). However, this is only a moderate correlation. In fact, many trail runners peak well into their 30s, 40s or even later for longer events. This is a great opportunity to examine the relationship between Place and Age further (see plot below).

The next most significant correlation between numeric variables is for Place and State (0.31). This one is particularly interesting for a few reasons. Again, we need to be really careful considering how off-balance our data is in regards to State. Well over half of participants are from Utah, so it's hard to generalize much beyond this analysis.

Nevertheless, my hypothesis of why there may be a moderate correlation is that participants who make the effort to travel out of state to compete are naturally going to invest more time and money into training. On the flip side, Utah participants are likely more evenly distributed in terms of time/money invested in training due to the fact that the event is in their home state. Let's explore this below.

Scatterplot of Age and Place Variables with State Dimension

```
ggplot(sg, aes(x=Age, y=Place, colour = State, fill = State)) +
  geom_jitter(size = 5, shape = 18) +
  theme_grey() + labs(title = "Speedgoat 2020 28k Results - Place & Age w/State Dimension") +
  theme(plot.title = element_text(hjust = 0.5, size = 15)) + scale_y_reverse()
```



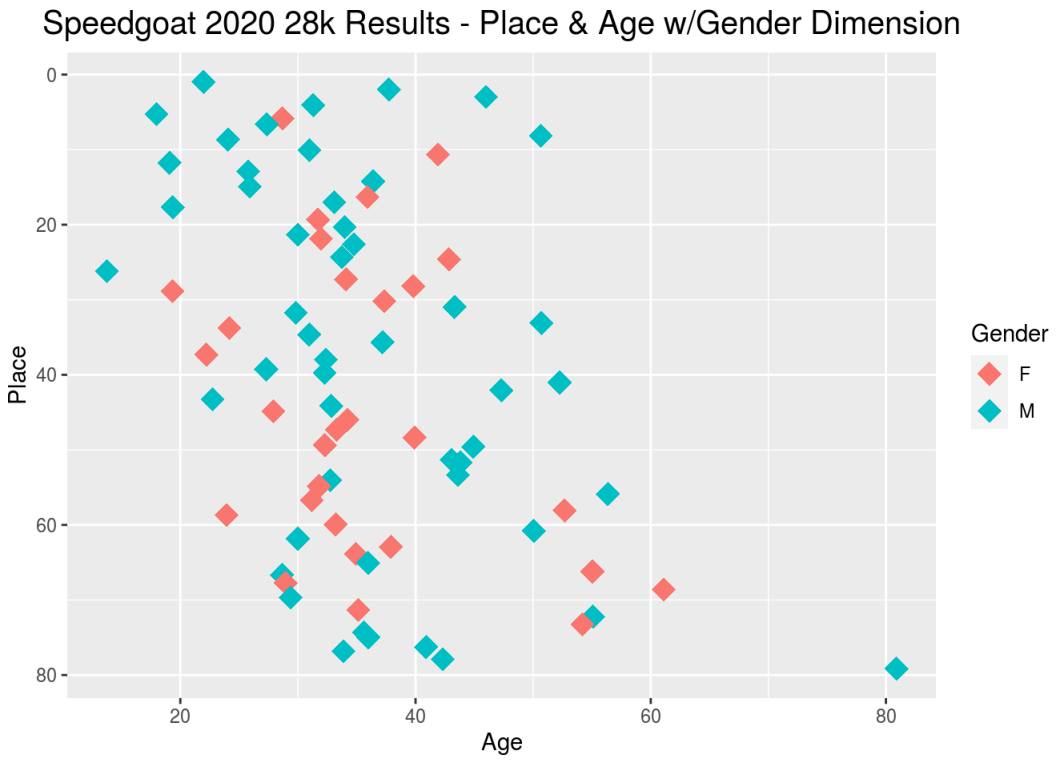
In the scatter plot above, we can clearly see the correlation between Place and Age. Note that I've flipped the y-axis so that lower (better) places are shown at the top of the plot and higher (worse) places are shown near the bottom.

Looking at the top 20 places on the plot, we see two particularly interesting realities. First, among these top finishers, the ages are very diverse. We have three of the five youngest competitors in the top 20 while also having someone more than 50 years old. This really speaks to the nature of the sport and how much is varies from a sport like gymnastics, for example, in terms of individual performance peaks.

Next, let's observe the breakdown by State in the plot above. Despite Utahns comprising 46 out of 79 participants, 12 of the top 20 finishers were from another state. Athletes from California and Colorado appear to have been particularly dominant. This makes sense as both states have strong trail running scenes and mountainous terrain available to train on. Overall, this seems to support the hypothesis that competitors from out of state will, on average, place better due to their greater commitment (as measured through their willingness to travel for an event like Speedgoat).

Scatterplot of Age and Place Variables with Gender Dimension

```
ggplot(sg, aes(x=Age, y=Place, colour = Gender)) +
  geom_jitter(size = 5, shape = 18) +
  theme_grey() + labs(title = "Speedgoat 2020 28k Results - Place & Age w/Gender Dimension") +
  theme(plot.title = element_text(hjust = 0.5, size = 15)) + scale_y_reverse()
```



Finally, I wanted to look at Gender as a dimension as well. The plot above shows how competitive the sport of trail running truly is regardless of gender and particularly at longer distances or over rugged terrain. Look no further than Courtney Dauwalter.

Thanks for your time. If you have any questions, ideas for improvement or just want to connect, please feel free to reach out.

Ryan Hansen ryan.hansen.slc@gmail.com