

Predicting Boston Marathon Cutoff Times

Sam McGee

2022-09-17

Predicting Boston Marathon Cutoff Times

Historically, the Boston Marathon is one of the more fabled marathons for runners to participate in. Part of this is the long lasting legacy of the race, while another part is the selective qualification process. Unlike the other 5 World Major Marathons, Boston does not have a lottery entry system, and participants must complete a prior marathon under a specified time in order to qualify, known as a “Boston Qualifying time” or “BQ.” The following table outlines the current 2022 BQ standards;

Age Group	Men’s BQ	Women’s BQ
18-34	3:00:00	3:30:00
35-39	3:05:00	3:35:00
40-44	3:10:00	3:40:00
45-49	3:20:00	3:50:00
50-54	3:25:00	3:55:00
55-59	3:35:00	4:05:00
60-64	3:50:00	4:20:00
65-69	4:05:00	4:35:00
70-74	4:20:00	4:50:00
75-79	4:35:00	5:05:00
80+	4:50:00	5:20:00

Despite the large field size for Boston Marathon and the level of athleticism required to BQ, in recent years the Boston Marathon has had to limit the accepted entries by imposing an additional “cutoff time” to the BQ standards in order to meet field size requirements. To accomplish this, Boston Marathon admits the fastest marathon entry times first until the allotted field size is at capacity. This means that the BQ standard is made more stringent by shortening the maximum time a successfully admitted participant could have finished their entry-marathon in. For example, in 2014 the cutoff time was 1:38, meaning the maximum accepted time for any age group was 1:38 *faster* than the BQ Standard.

Over the years, this cutoff time has been influenced largely by the number of entrants who had run a BQ time during the qualification window, as well as the allotted field size for the Boston Marathon. Knowing this, we may be able to use data from previous years to develop a model to predict the cutoff time based on the total number of runners with a BQ standard, the field size capacity for the given year, and the number of qualifiers not accepted (as a proxy estimate for the proportion of entrants with BQ standards).

Additional information can be found here;

<https://www.baa.org/races/boston-marathon/qualify>

Previous years' data (2012 through 2020)

The cutoff time was initially implemented in 2012, at 1:14, and prevented 3,228 athletes with the BQ standard from being registered due to the field size of 27,000 that year. The table of all current cutoff time information is shown below;

```
website <- read_html('https://www.baa.org/races/boston-marathon/qualify')
tab <- html_table(website)
Cutoff_info_unformatted <- tab[[2]]
# convert columns 2 & 4 to numeric objects, convert column 3 to date-time object
Cutoff_info <- as.data.frame(matrix(NA,
                                   nrow=nrow(Cutoff_info_unformatted),
                                   ncol=ncol(Cutoff_info_unformatted)))
Cutoff_info[,1] <- Cutoff_info_unformatted[,1]
for (i in 1:nrow(Cutoff_info_unformatted)) {
  Cutoff_info[i,2] <- Cutoff_info_unformatted[i,2] %>%
    str_replace_all(',', '') %>%
    as.numeric()
  Cutoff_info[i,3] <- format(as.POSIXct(as.character(Cutoff_info_unformatted[i,3]),
                                                    format='%M:%OS'),
                            '%M:%OS')
  Cutoff_info[i,4] <- Cutoff_info_unformatted[i,4] %>%
    str_replace_all(',', '') %>%
    as.numeric()
}
```

```
## Warning in Cutoff_info_unformatted[i, 4] %>% str_replace_all(",", "") %>% : NAs
## introduced by coercion
```

```
colnames(Cutoff_info) <- c('Year', 'FieldSize', 'Cutoff', 'NotAccepted')
Cutoff_info
```

##	Year	FieldSize	Cutoff	NotAccepted
## 1	2012	27000	01:14	3228
## 2	2014	36000	01:38	2976
## 3	2015	30000	01:02	1947
## 4	2016	30000	02:28	4562
## 5	2017	30000	02:09	2957
## 6	2018	30000	03:23	5062
## 7	2019	30000	04:52	7248
## 8	2020	31500	01:39	3161
## 9	2021	20000	07:47	9215
## 10	2022	30000	00:00	NA

The field size for 2021 was heavily restricted due to the COVID-19 pandemic. For our analyses, the data for that year will be removed.

Webscrape information for marathon finishers that met BQ standard

Because so many marathons of varying sizes are run each year, we will use a selection of the 30 marathons each year that garnered the most BQs. These data are curated and made available by www.marathonguide.com which will also aid our analyses.

Here, we will pull the years needed for our analyses as well as clean up the tables to coerce the columns into the correct data format.

```
BQ_years <- c('2011','2012','2014','2015','2016','2017','2018','2019','2020','2022')

for (i in BQ_years) {
  website <- read_html(paste0('http://www.marathonguide.com/races/BostonMarathonQualifyingRaces.cfm?Year=', i))
  tab <- html_table(website)
  # assign to temporary table
  temp_BQ_tab <- tab[[15]]
  # clean up "BQers" column to remove commas and assert as numeric
  BQers_cleaned <- temp_BQ_tab
  BQers_cleaned[,4] <- NA
  for(j in 1:nrow(temp_BQ_tab)) {
    BQers_cleaned[j,4] <- temp_BQ_tab[j,4] %>%
      str_replace_all(',', '') %>%
      as.numeric()
  }
  assign(paste('BQ_Total_', i, sep=''), BQers_cleaned, envir = .GlobalEnv)
}
```

```
## Warning in temp_BQ_tab[j, 4] %>% str_replace_all(",", "") %>% as.numeric(): NAs
## introduced by coercion
```

```
## Warning in temp_BQ_tab[j, 4] %>% str_replace_all(",", "") %>% as.numeric(): NAs
## introduced by coercion
```

```
## Warning in temp_BQ_tab[j, 4] %>% str_replace_all(",", "") %>% as.numeric(): NAs
## introduced by coercion
```

Testing the theory that “More BQ Runners” yields “Stricter Cutoff Times”

We should first look to see if there may be a correlation between the number of runners meeting the BQ standard and the ensuing cutoff time for that year.

Bear in mind; the number of runners in a given year are those who would run the following year for the Boston Marathon, so our information between “BQers in a given year” and the “Cutoff Time” needs to be staggered to account for this.

```
BQ_Total_v_Cutoff <- as.data.frame(matrix(nrow=length(BQ_years), ncol=3))
rownames(BQ_Total_v_Cutoff) <- BQ_years
colnames(BQ_Total_v_Cutoff) <- c('BQers', 'Cutoff', 'Seconds')

for (i in 1:length(BQ_years)) {
  cur_year <- BQ_years[i]
  BQ_Total_v_Cutoff[as.character(cur_year), 'BQers'] <-
    sum(get(paste0('BQ_Total_', cur_year, sep=''))[,4])
  if (cur_year == '2022') {
    break
  } else {
    cuttime_formatted <- as.POSIXct(Cutoff_info[(Cutoff_info[,1] == BQ_years[i+1]), 3], format='%M:%OS')

    BQ_Total_v_Cutoff[as.character(cur_year), 'Cutoff'] <- format(cuttime_formatted, '%M:%OS')
```

```

BQ_Total_v_Cutoff[as.character(cur_year), 'Seconds'] <-
  (as.integer(format(cuttime_formatted, '%M'))*60) + as.integer(format(cuttime_formatted, '%OS'))
}
}

# compare cutoff time in seconds to number of BQers
# removing year 2022 because that cutoff time has not been determined yet
BQ_pre2022 <- subset(BQ_Total_v_Cutoff, rownames(BQ_Total_v_Cutoff) != '2022')

ggplot(BQ_pre2022,
  aes(x=BQers,y=Seconds,
      label=rownames(BQ_pre2022))) +
  geom_point(color='blue', size=2) +
  geom_smooth(method='lm', color='gold') +
  xlab('Total Runners Meeting BQ Standards in a Given Year')+
  ylab('Cutoff Time in Seconds') +
  geom_text(vjust=2)

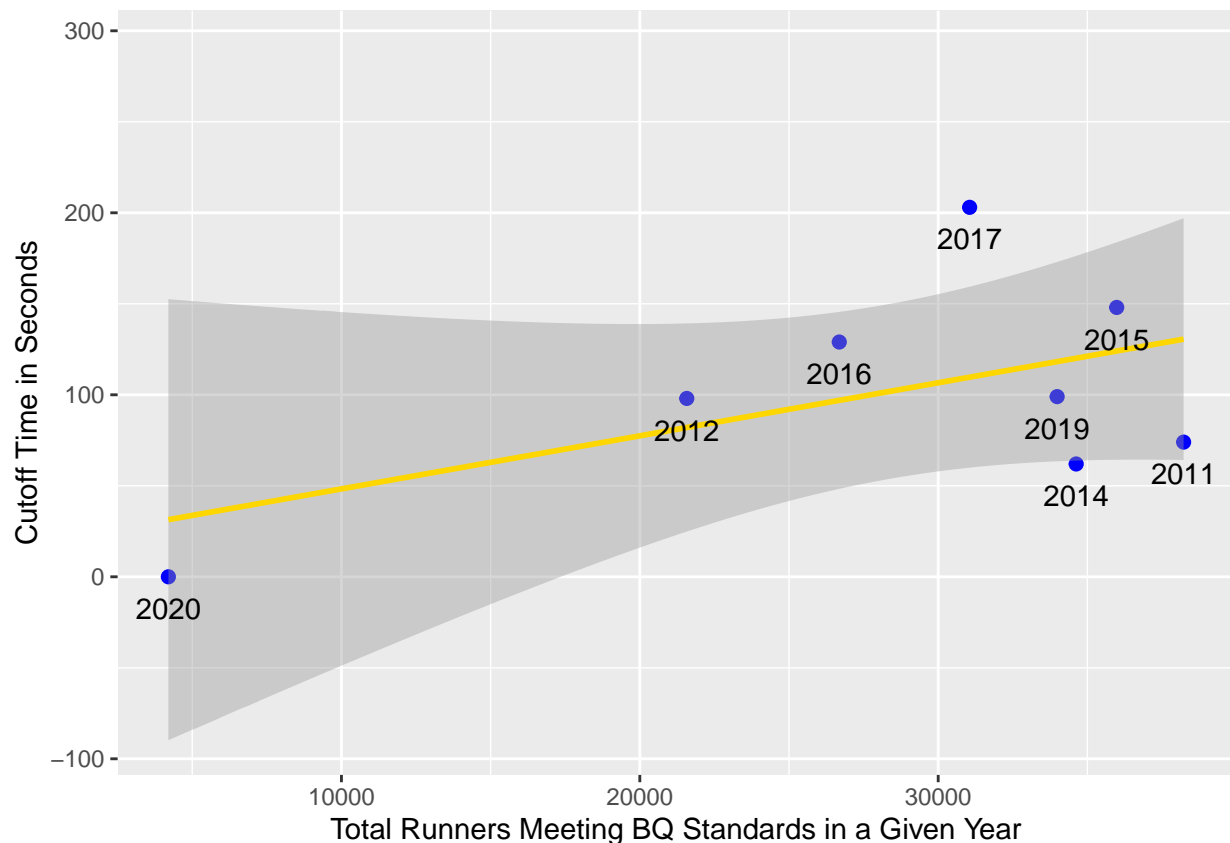
## 'geom_smooth()' using formula 'y ~ x'

## Warning: Removed 1 rows containing non-finite values (stat_smooth).

## Warning: Removed 1 rows containing missing values (geom_point).

## Warning: Removed 1 rows containing missing values (geom_text).

```



Here we can see that there *may* be a moderate relationship between the number of runners with the BQ standard and the stringency of the ensuing cutoff time. The number of BQers and the ensuing cutoff time for 2020 was very likely influenced by the COVID-19 pandemic, as there were not many opportunities to achieve a BQ standard throughout 2020 due to the cancellation of so many races.

```
cor(BQ_pre2022$Seconds,BQ_pre2022$BQers)
```

```
## [1] NA
```

The correlation between these two variables supports our assumption that any relationship between these variables is moderate at best. Given this relationship, our resulting predictions will likely need to be treated with tempered caution.

BQ Cutoff Time, Total BQers Only

Our first model will try to predict the Boston Cutoff time (in Seconds) using only the total number of runners who achieved the BQ standard in the previous year.

```
BQfit <- lm(Seconds ~ BQers, data=BQ_pre2022)
summary(BQfit)
```

```
##
## Call:
## lm(formula = Seconds ~ BQers, data = BQ_pre2022)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -58.088 -37.689  -1.635   25.979   93.322
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  19.146353   56.870061   0.337    0.748
## BQers         0.002915    0.001887   1.545    0.173
##
## Residual standard error: 55.52 on 6 degrees of freedom
## (1 observation deleted due to missingness)
## Multiple R-squared:  0.2847, Adjusted R-squared:  0.1655
## F-statistic: 2.388 on 1 and 6 DF, p-value: 0.1732
```

The F -statistic for this regression is not significant, and therefore the cutoff score is likely influenced by other variables rather than the number of runners with a BQ standard during a given year. The R^2 value is fairly low as well, at about 0.3, so this model doesn't have a whole lot of predictive ability either. However, for the sake of this project, we'll continue to see if we can come close to predicting the 2022 cutoff time.

```
CutoffPredicted <- as.data.frame(predict(BQfit, newdata=BQ_Total_v_Cutoff))

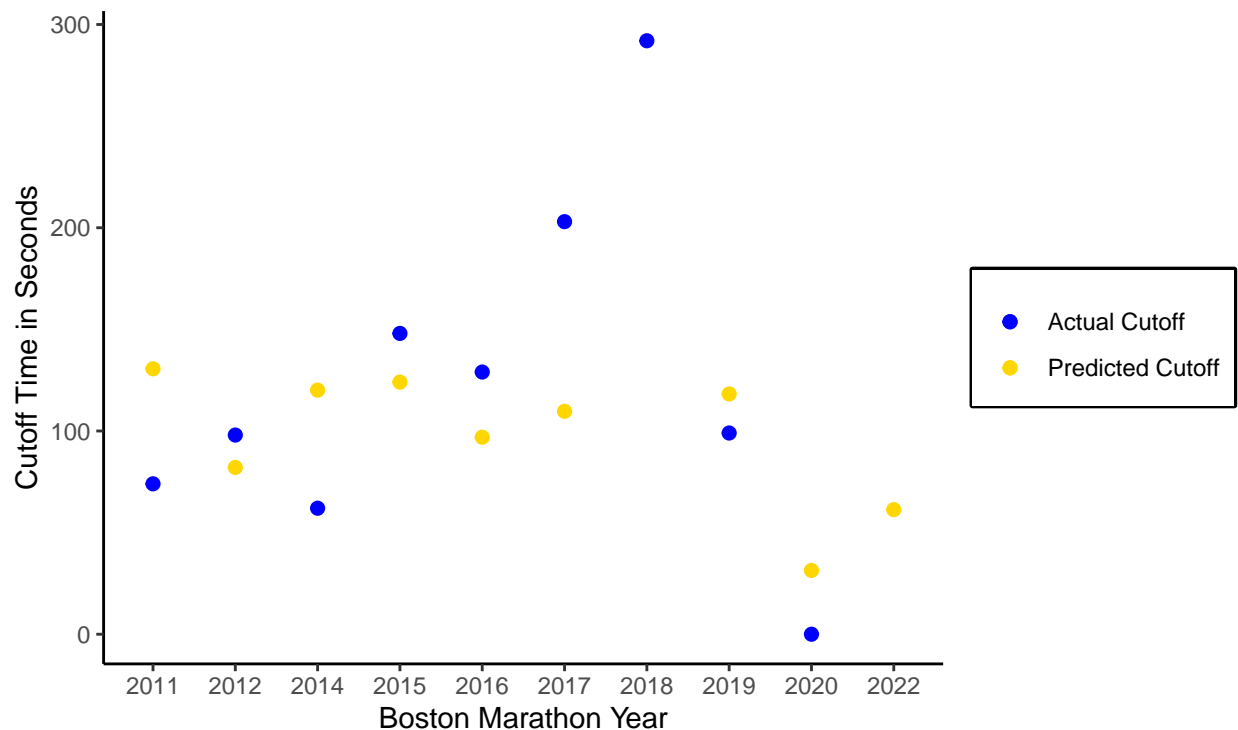
colnames(CutoffPredicted) = 'PredictedCutoff'
CutoffPredicted$ActualCutoff <- BQ_Total_v_Cutoff$Seconds
CutoffPredicted
```

```
##      PredictedCutoff ActualCutoff
## 2011      130.58622         74
## 2012       82.04081         98
## 2014      120.08826         62
## 2015      124.04722        148
## 2016       96.94086        129
## 2017      109.67776        203
## 2018          NA         292
## 2019      118.22830         99
## 2020       31.39057          0
## 2022       61.28395         NA
```

```
ggplot(CutoffPredicted, aes(x=rownames(CutoffPredicted),y=value)) +
  geom_point(aes(y=PredictedCutoff, color='Predicted Cutoff'),size= 2) +
  geom_point(aes(y=ActualCutoff, color='Actual Cutoff'),size= 2) +
  xlab('Boston Marathon Year') +
  ylab('Cutoff Time in Seconds') +
  ggtitle('Predicted versus Actual Cutoff Times Using\nTotal Count of Runners with a BQ Standard') +
  theme(panel.background = element_rect(fill='white', color=NA),
        axis.line = element_line(color='black',size=.5),
        plot.title = element_text(size=18, face='bold'),
        legend.title = element_blank(),
        legend.box.background = element_rect(color='black',size=1),
        legend.key = element_rect(fill='white',color=NA)) +
  scale_color_manual(values = c('blue','gold'))
```

```
## Warning: Removed 1 rows containing missing values (geom_point).
## Removed 1 rows containing missing values (geom_point).
```

Predicted versus Actual Cutoff Times Using Total Count of Runners with a BQ Standard



It looks like this model really doesn't do well, and aside from the 2020 predicted Cutoff value, seems to predict a cutoff score around 110 seconds. Let's see if adding more information about the allotted Field Size can improve our predictions.

Predicting Cutoff Times using Number of Runners with BQ Standard and Boston Marathon Field Size

For this second model, we can include information about the allotted Boston Marathon Field Size. Generally, this number has been near 30,000 runners per year, although certain years have had more or less. Given that a larger Field Size would allow for more runners with the BQ standard to register, we should expect that years with a larger Field Size will have a less stringent Cutoff time.

```
BQ_field_cutoff <- as.data.frame(matrix(NA, nrow = nrow(BQ_pre2022), ncol=3))
BQ_field_cutoff[,1] <- BQ_pre2022$BQers
BQ_field_cutoff[,2] <- Cutoff_info[1:nrow(BQ_field_cutoff),2]
BQ_field_cutoff[,3] <- BQ_pre2022$Seconds
colnames(BQ_field_cutoff) <- c('BQers', 'FieldSize', 'Cutoff')
rownames(BQ_field_cutoff) <- rownames(BQ_pre2022)

BQfit_field <- lm(Cutoff ~ (BQers + FieldSize), data=BQ_field_cutoff)
summary(BQfit_field)
```

```
##
## Call:
## lm(formula = Cutoff ~ (BQers + FieldSize), data = BQ_field_cutoff)
```

```
##
## Residuals:
##      2011      2012      2014      2015      2016      2017      2019      2020
## -33.961 -25.871 -54.858  28.639  26.777  92.724 -24.449  -9.002
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.023e+02  1.395e+02  -0.733   0.496
## BQers        1.843e-03  2.208e-03   0.835   0.442
## FieldSize    5.178e-03  5.425e-03   0.954   0.384
##
## Residual standard error: 55.94 on 5 degrees of freedom
## (1 observation deleted due to missingness)
## Multiple R-squared:  0.3949, Adjusted R-squared:  0.1529
## F-statistic: 1.632 on 2 and 5 DF,  p-value: 0.2848
```

Our F -statistic here is even smaller than before, and thus our model may not be improved with the additional information. The R^2 value is only slightly improved to 0.32, but this model still does not have a great deal of predictive ability and we should take these values with a serious grain of caution. Nonetheless, we'll try to predict the 2022 cutoff time using this model.

```
BQ_Total_w_Field <- BQ_Total_v_Cutoff
BQ_Total_w_Field$FieldSize <- Cutoff_info[,2]

FieldCutoffPredicted <- predict(BQfit_field, newdata = BQ_Total_w_Field)
FieldCutoffPredicted
```

```
##      2011      2012      2014      2015      2016      2017      2018
## 107.960713 123.871205 116.857630 119.360553 102.223460 110.275940      NA
##      2019      2020      2022
## 123.448686   9.001812  79.680572
```

Once again, our model doesn't do a great job of predicting the cutoff times historically, and therefore we should not put too much faith into the predicted 72 second cutoff time for 2022. In terms of pure "gut-check" feeling with these results, 72 seconds or 1:12 does not seem out of place for what we might expect, so the model may be capturing some of the variance and explaining it here, but there are likely other factors we have not considered here that contribute to the final cutoff time. Out of curiosity however, we'll overlay our model's predicted cutoff time over the observed cutoff times for each year.

```
ModelvReality <- as.data.frame(matrix(NA, nrow=10, ncol=2))
ModelvReality[,1] <- BQ_Total_v_Cutoff$Seconds
ModelvReality[,2] <- FieldCutoffPredicted
colnames(ModelvReality) <- c('ActualCutoff', 'PredictedCutoff')
rownames(ModelvReality) <- rownames(BQ_Total_v_Cutoff)

ggplot(ModelvReality, aes(x=rownames(ModelvReality), y=value)) +
  geom_point(aes(y=ActualCutoff, color = 'Actual Cutoff'), size= 2) +
  geom_point(aes(y=PredictedCutoff, color = 'Predicted Cutoff'), size= 2) +
  xlab('Boston Marathon Year') +
  ylab('Cutoff Time in Seconds') +
  ggtitle('Predicted versus Actual Cutoff Time Using\nTotal Runners with a BQ Standard and Field Size') +
  theme(panel.background = element_rect(fill='white', color=NA),
        axis.line = element_line(color='black', size=.5),
```



```

plot.title = element_text(size=18, face='bold'),
legend.title = element_blank(),
legend.box.background = element_rect(color='black',size=1),
legend.key = element_rect(fill='white',color=NA)) +
scale_color_manual(values=c('blue','gold'))

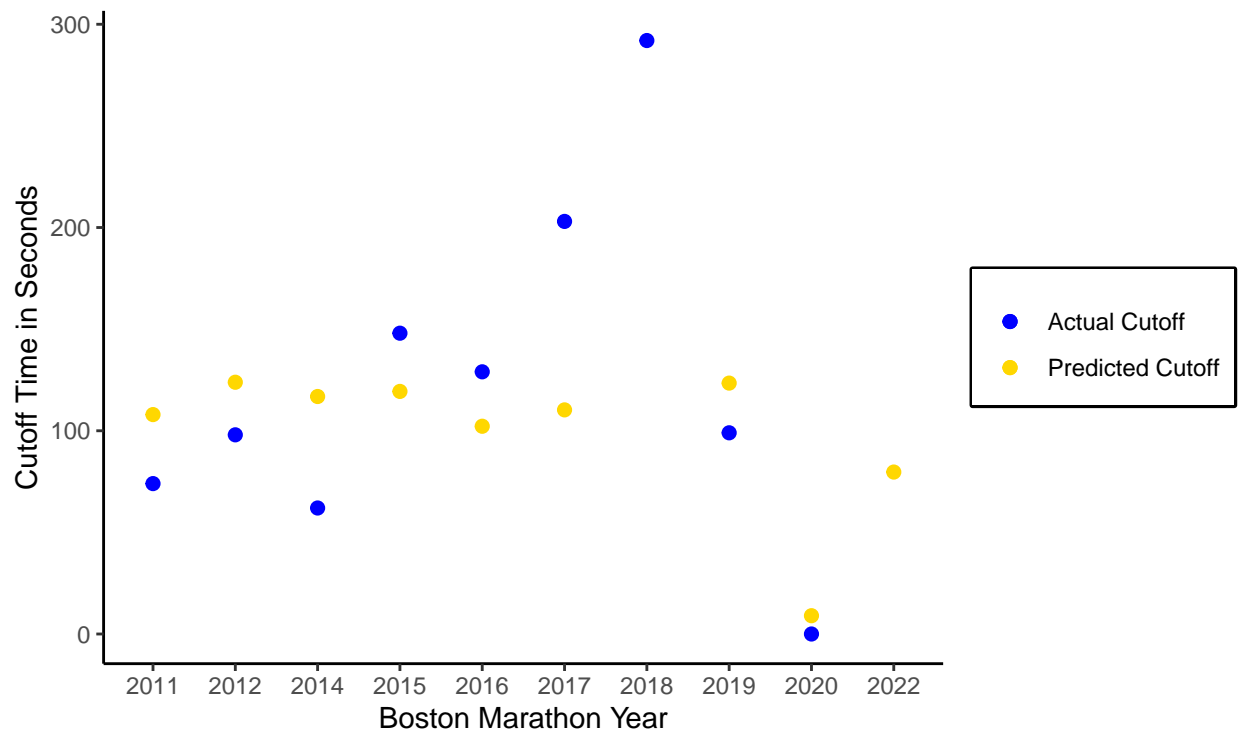
```

```

## Warning: Removed 1 rows containing missing values (geom_point).
## Removed 1 rows containing missing values (geom_point).

```

Predicted versus Actual Cutoff Time Using Total Runners with a BQ Standard and Field Size



Conclusion

In short, this model does not appear to perform very well with the available data, and there are likely other factors not included in these data contributing to the final Boston Cutoff times. Additional methods could be considered to try to better predict the Cutoff time using these data, for instance Bayesian inference may be well-suited to this given the small sample size available. However, for our purposes today with the announcement of the official 2022 Boston Cutoff time coming soon now that registration has closed, *and* given this is a low-stakes analysis, we might consider a Cutoff near 72 seconds, or 1:12, as being one likely 2022 Boston Marathon Cutoff time.