# Time Prediction for Track and Field Athletes

Clancy Andrews

## Abstract

The present paper looked to analyze the performance of top Track and Field athletes. The goal was to determine if we could adequately predict the time that a top Track and Field athlete could achieve. We implemented two linear regression models to predict time in seconds using gender and distance, and gender, distance in meters and placement. The following mathematical formulas for the depicted models are $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \epsilon$ and $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \epsilon$, respectively. We found that both models performed well when predicting the time. The model containing predictors gender and distance had an $R^2$ of 0.9951 and a p-value of $2.2 \times 10^{-16}$. The model containing predictors gender, distance, and place had an $R^2$ of 0.9951 and a p-value of $2.2 \times 10^{-16}$. When observing the F-statistic for the models, we found that the model with two predictors had a F-stat of $1.271 \times 10^6$ with 12628 degrees of freedom. The model with three predictors had a F-stat of $8.531 \times 10^5$ with 12620 degrees of freedom. Based on these findings, we determined that both models can adequately predict the time that an athlete can run a race given their gender, distance, and placement.

Keywords: Linear Regression, Statistical Learning, Running, Cross Country

## 1. Introduction

This paper constructs linear regression models on quantitative and qualitative data regarding top Track and Field athletes. We look into two linear regression models to determine if there is a way to adequately predict the times of the athletes based on gender of the runner, distance of the race in meters, and the placement the runners achieved in that race. According to Liu, "…literature leads us to believe that there is no prediction model available to accurately predict the future magnitude of any performance in track and field." (1). Given the complexity of predicting the time one can achieve without knowing the conditions before hand can lead to a precision and accuracy in a statistical learning method. With this in mind, we hope to better understand how the linear regression models can lead to a higher error in prediction.

## 2. Related Works

There has been several different works on the topic of time prediction in Track and Field athletes. The papers referenced here have the same distinctions on time prediction: it is difficult to accurately predict an athletes time based on a few parameters. This can be due to many reasons, some being performance specialization, racing conditions, and potential. As stated in the paper by Blythe and Király, "Any classical method for performance prediction which only takes this information into account will predict that green performs similarly over 1500m to blue and red. However, this is unrealistic, since it does not take into account event specialization…" (3). Often, data collected on running performance lacks the necessary information needed to accurately predict time. Due to this, predictions can have large errors associated with them. We hope to address this further in the following sections.

## 3. Research Questions

Since one of the goals of this analysis is understanding the performance of top runners in Track and Field and use that data to predict future top performers in the sport, we would like to answer the following questions:

- Can we predict the time that a runner can achieve based on their gender and the distance ran?
- Can we predict the time that a runner can achieve based on their gender, the distance ran, and the finishing placement of the competition?

## 4. Data

The source of data we will be using is the Top Running Times data set located on Kaggle: https://www.kaggle.com/data sets/jguerreiro/running. This data set contains the top 1000 performances for each track and field event for both the men and women. The data is comprised of 18244 observations. Of the 18244 observations, 50% are of gender male and 50% are of gender female. Here is a sample of the data set:

```
##    Rank            Time            Name Country Date.of.Birth Place
## 1     1 00:01:40.910000   David Rudisha     KEN    1988-12-17     1
## 2     2 00:01:41.010000   David Rudisha     KEN    1988-12-17     1
## 3     3 00:01:41.090000   David Rudisha     KEN    1988-12-17     1
## 4     4 00:01:41.110000 Wilson Kipketer     DEN    1970-12-12     1
## 5     5 00:01:41.240000 Wilson Kipketer     DEN    1970-12-12     1
## 6     6 00:01:41.330000   David Rudisha     KEN    1988-12-17     1
## 7     7 00:01:41.510000   David Rudisha     KEN    1988-12-17     1
## 8     8 00:01:41.540000   David Rudisha     KEN    1988-12-17     1
## 9     9 00:01:41.730000   Sebastian Coe     GBR    1956-09-29     1
## 10    9 00:01:41.730000 Wilson Kipketer     DEN    1970-12-12     1
##               City       Date Gender Event
## 1           London 2012-09-08    Men 800 m
## 2            Rieti 2010-08-29    Men 800 m
## 3           Berlin 2010-08-22    Men 800 m
## 4             Köln 1997-08-24    Men 800 m
## 5           Zürich 1997-08-13    Men 800 m
## 6            Rieti 2011-10-09    Men 800 m
## 7  Heusden-Zolder 2010-10-07    Men 800 m
## 8      Saint-Denis 2012-06-07    Men 800 m
## 9          Firenze 1981-10-06    Men 800 m
## 10       Stockholm 1997-07-07    Men 800 m
```

The different variables in the data set are as follows:

- Rank: The all-time ranking by time of runner for each event/distance
- Time: The time the runner ran hh:mm:ss.ms
- Name: Name of the runner
- Country: Country they were representing
- Date.of.Birth: Runner's date of birth
- Place: Placement the got in the race that the time was accomplished in
- City: City that the race took place in
- Date: Date of the race
- Gender: Man or Woman
- Event: Distance of race/Type of race (in meters)

### 5. Methods

I plan to using Multiple Linear Regression to achieve a model that can be used to determine the time a top runner would run given the gender of the runner and the distance of the race, and the time a top runner would run given the gender of the runner, distance of the race, and the placement they got in the race. The following are the mathematical models:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \epsilon$$

where $x_1$ evaluates to a 0 if the runner is a man and a 1 if the runner is a woman. Distance, in meters, will be represented by $x_2$. Our response $y$ is the time in seconds.

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \epsilon$$

where $x_1$ evaluates to a 0 if the runner is a man and a 1 if the runner is a woman. The distance, in meters, will be represented by $x_2$. The $x_3$ represents the placement of the runner. Our response $y$ is the distance in meters.

### 6. Results

We cleaned out code and ran linear regression modeling functions and predictions on training and testing data to get the following results. Refer to section 8 for the codes used to get these results.

### 6.1. Model One

Let us analyze the linear model with our response variable being the time in seconds.

```
##
## Call:
## lm(formula = TimeSeconds ~ Gender + DistanceMeters, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -569.26  -87.11  -49.45  124.91  625.05
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -1.749e+02  2.523e+00  -69.33   <2e-16 ***
## GenderWomen     2.127e+02  3.228e+00   65.90   <2e-16 ***
## DistanceMeters  1.925e-01  1.208e-04 1593.08   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 181.4 on 12628 degrees of freedom
## Multiple R-squared:  0.9951, Adjusted R-squared:  0.9951
## F-statistic: 1.271e+06 on 2 and 12628 DF,  p-value: < 2.2e-16
```

Based on the summary of our model, we can see that the equation for the model is

$$y = -174.9 + 212.7x_1 + 0.19x_2,$$

where $x_1$ is the gender dummy variable and $x_2$ is the distance variable. We had an $R^2$ value of 0.9951 and an adjusted $R^2$ value of 0.9951. The p-value for the model was less than $2.2 \times 10^{-16}$. Let us now visualize the diagnostic plots for the model:

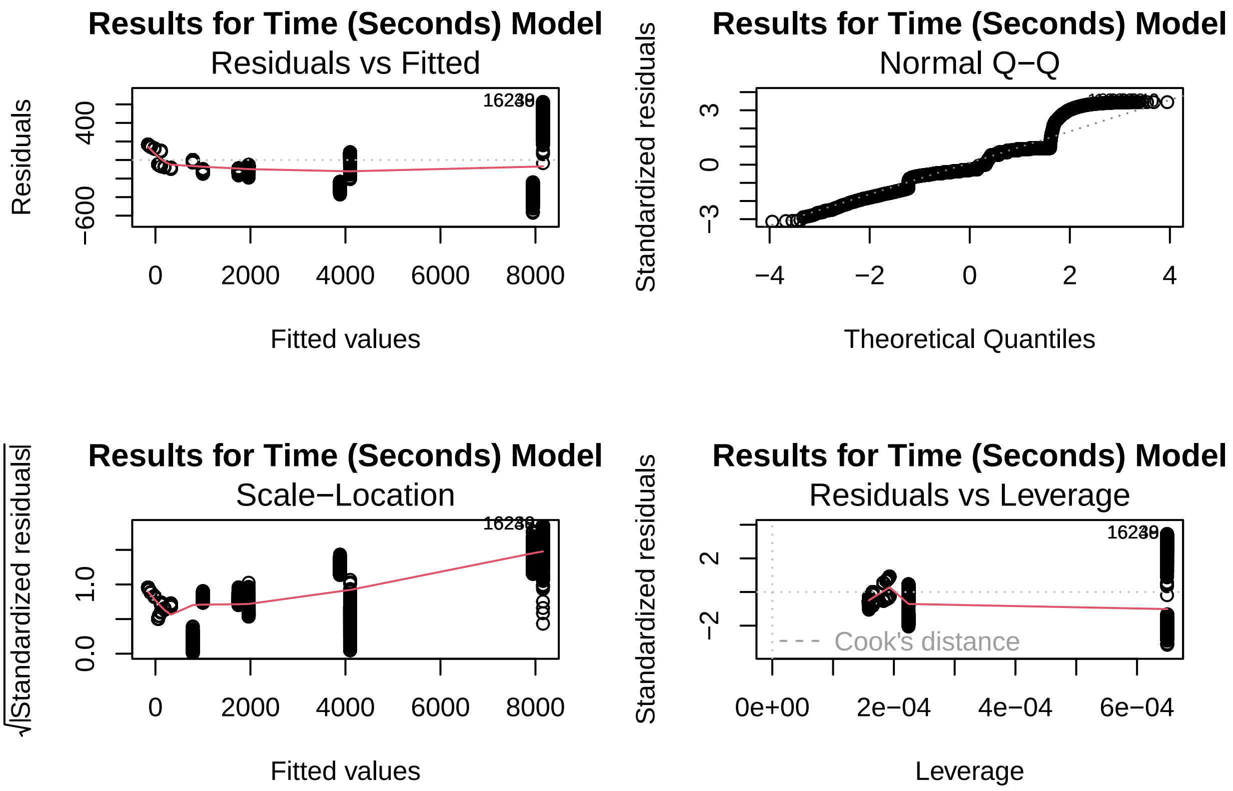

Figure 1: Diagnostic plots for model with predictors Gender and Distance.

The Residuals vs Fitted plot allows us to see the relative deviation between the observed values and the predicted values. We can see some heteroscedasticity, standard errors are non-constant, which leads to a lower precision in the model. The quantile-quantile plot shows the fitted quantiles versus the theoretical quantiles. There is an obvious difference in distributions between the fitted and theoretical residuals. The standardized residuals versus leverage plot allows us to determine any highly influential observations. We can see that there are a few influential observations in the model. The following is the plot of observed time given by the testing data set versus the predicted time given by predicting the time with the testing data set.

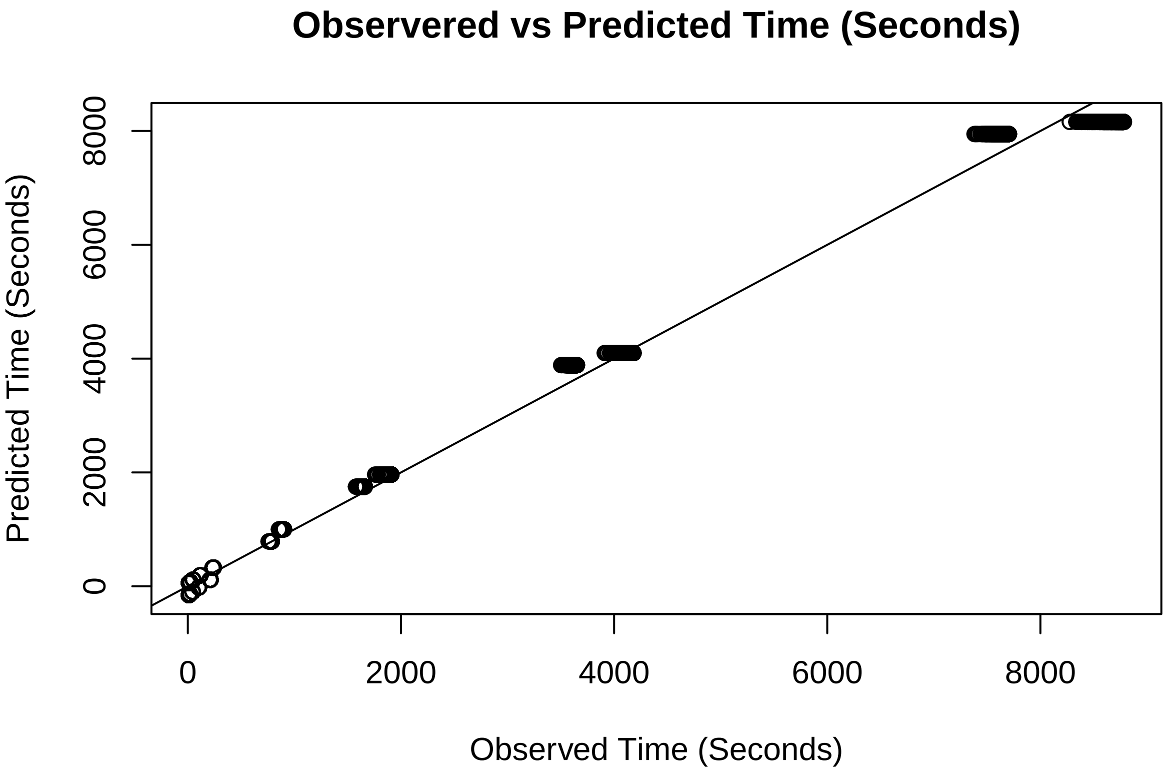

Figure 2: Observed versus Predicted time in seconds using the model with predictors Gender and Distance.

When we observe the plot, we notice that some of the predicted observations have a negative time value. We can see, using the code `pred.plot$fit[which.min(pred.plot$fit)]` that our minimum fitted value of our predicted time is -155.6857. Based on this information, we know that our model may not be able to predict the time properly when the distance is a smaller value.

### 6.2. Model Two

Let us now analyze the linear model with our response variable being the time in seconds given the predictors Gender, Distance, and Placement.

```
summary(fit.time2)
```

```
##
## Call:
## lm(formula = TimeSeconds ~ Gender + DistanceMeters + Place, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -585.05  -85.17  -52.60  122.04  678.74
##
## Coefficients:
##                 Estimate Std. Error  t value Pr(>|t|)
## (Intercept)    -1.573e+02  3.114e+00  -50.511   <2e-16 ***
## GenderWomen     2.121e+02  3.218e+00   65.910   <2e-16 ***
## DistanceMeters  1.926e-01  1.212e-04 1589.413   <2e-16 ***
## Place          -7.265e+00  7.563e-01   -9.606   <2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 180.8 on 12620 degrees of freedom
##   (7 observations deleted due to missingness)
## Multiple R-squared:  0.9951, Adjusted R-squared:  0.9951
## F-statistic: 8.531e+05 on 3 and 12620 DF,  p-value: < 2.2e-16
```

Based on the summary of our model, we can see that the equation for the model is

$$y = -157.3 + 212.1x_1 + 0.19x_2 - 7.27x_3,$$

where $x_1$ is the gender dummy variable, $x_2$ is the distance variable, and $x_3$ is the place variable. We had an $R^2$ value of 0.9951 and an adjusted $R^2$ value of 0.9951. The p-value for the model was less than $2.2 \times 10^{-16}$. Let us now visualize the diagnostic plots for the model:

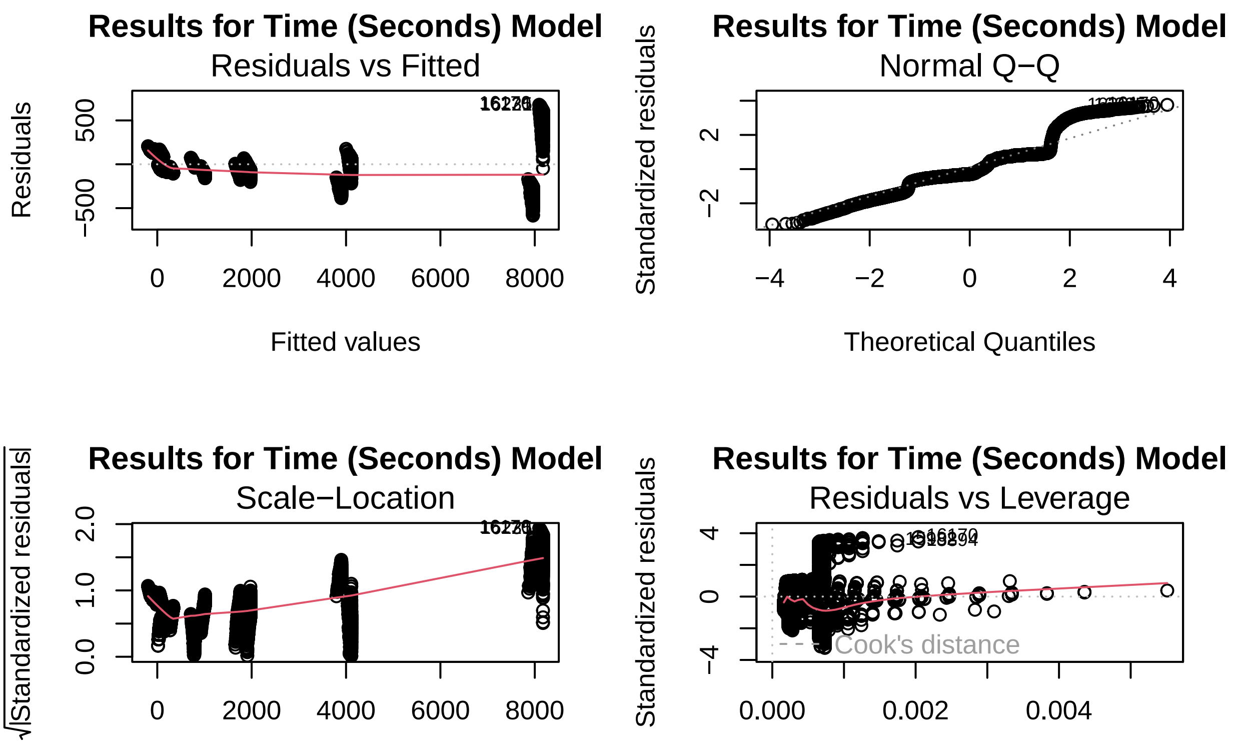

Figure 3: Diagnostic plots for model with predictors Gender, Distance, and Placement.

The second model has similar diagnostic plots. The Residuals vs Fitted plot allows us to see the relative deviation between the observed values and the predicted values. We can see some heteroscedasticity, standard errors are non-constant, which leads to a lower precision in the model. The quantile-quantile plot shows the fitted quantiles versus the theoretical quantiles. There is an obvious difference in distributions between the fitted and theoretical residuals. The standardized residuals versus leverage plot allows us to determine any highly influential observations. We can see that there are a few influential observations in the model. The following is the plot of observed time given by the testing data set versus the predicted time given by predicting the time with the testing data set.

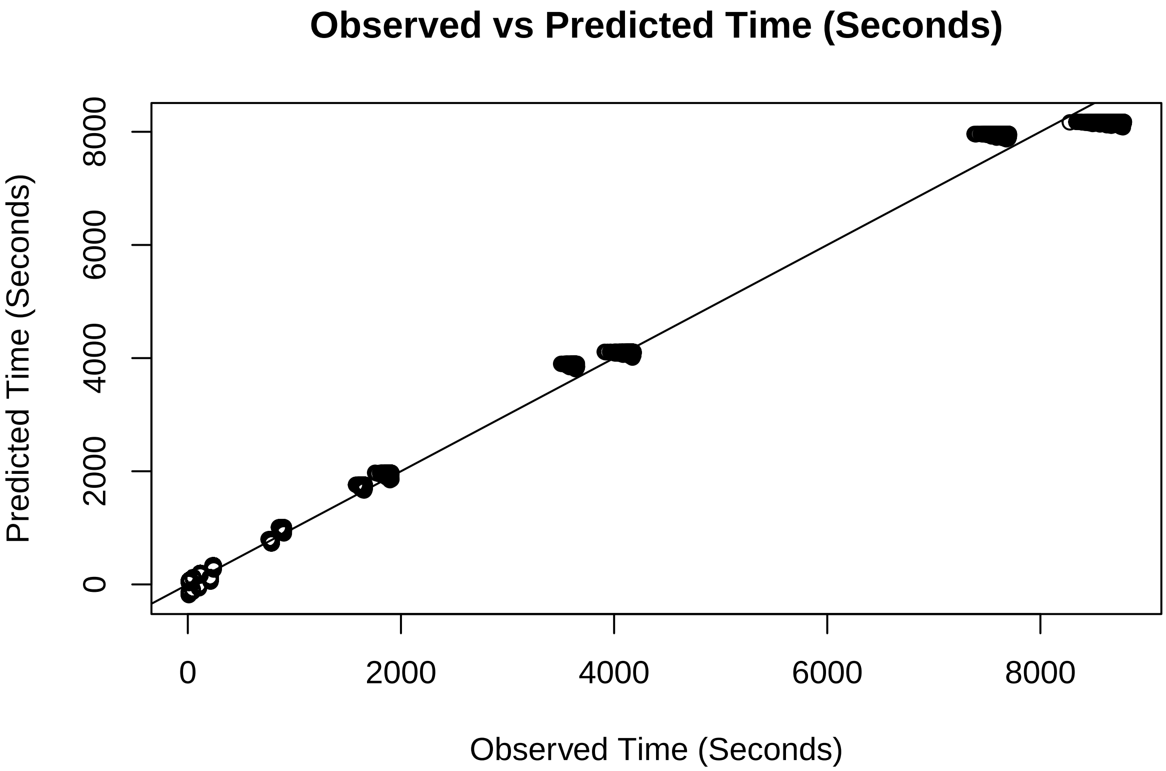

Figure 4: Observed versus Predicted time in seconds using the model with predictors Gender, Distance, and Placement.

When we observe the plot, we notice that some of the predicted observations have a negative time value, much like the first model. We can see, using the code `pred.plot2$fit[which.min(pred.plot2$fit)]` that our minimum fitted value of our predicted time is -188.8757. Based on this information, we know that our model may not be able to predict the time properly when the distance is a smaller value.

## 7. Conclusion

After evaluating the models and running predictions, there seems to be evidence of underlying issues with the current methodology. While the models had an acceptable $R^2$s and p-values, the diagnostic plots and predicted values had less then satisfactory results. James states, "…many of the techniques used throughout this paper may not provide the same level of insight when small annual samples of results lead to extremely volatile trends." (2). Further study using different statistical learning methods to get a better understanding of limitations and achieving more accurate results for time prediction is suggested.

## 8. Code and Implementation

The following sections are the code segments used to manipulate the data and run linear regression.

### 8.1. Time To Seconds

The following code converts the time from a string format to a float (real) number type. It then creates a new column in the dataframe to store those values.

```
# Creates a function that converts time (in a string format) to a float type
to.seconds = function(x){
  unlist(lapply(x, function(i){
    i = as.numeric(strsplit(i,':',fixed=TRUE)[[1]])
      if (length(i) == 3)
        i[1]*3600 + i[2]*60 + i[3]
      else if (length(i) == 2)
        i[1]*60 + i[2]
      else if (length(i) == 1)
        i[1]
  }))
}

# Creates a new data column for the time in seconds,
# and then converts all time to seconds for each observation
for (i in 1:length(data$Time)) {
  data$TimeSeconds[i] = to.seconds(data$Time[i])
}
```

### 8.2. Distance To Meters

The following code converts the distance from a string format to a integer number type. It then creates a new column in the dataframe to store those values.

```
# Converts each distance from string to meters (half-marathon and marathon distances har
to.meters = function(x) {
  hmarathon = 21098
  marathon = 42195

  if(x == "Half marathon") {
    i = hmarathon
    return(i)
  }
  if (x == "Marathon") {
    i = marathon
    return(i)
  }
  else {
  unlist(lapply(x,function(i){
    i = gsub(",", "", i)
    i = as.numeric(strsplit(i, ' ', fixed = TRUE)[[1]])
  }))}
}

# Creates a new data column for the distance in meters
n = 1
for (i in data$Event) {
  data$DistanceMeters[n] = to.meters(i)
  n = n + 1
}
```

### 8.3. Gender To Boolean

The following code converts the gender from a string format to a integer (boolean) number type. It then creates a new column in the dataframe to store those values.

```
# Converts each Gender to a 1 if woman and 0 otherwise
to.boolean = function(x) {

  if(x == "Women") {
    return(1)
  }
  else {
    return(0)
  }
}

# Creates a new data column for the gender as a boolean integer
n = 1
for (i in data$Gender) {
  data$GenderBool[n] = to.boolean(i)
  n = n + 1
}
```

### 8.4. Linear Regression

The following code splits the data set into training and testing sets. It then creates linear regression models according to the criteria specified in section 5 on the training data, and runs predictions using the testing data.

```
# Splitting the training and testing data
library(caTools)
set.seed(1)
split = sample.split(data, SplitRatio= 0.7)
train = subset(data, split == T)
test = subset(data, split == F)

# Linear Regression Model to Estimate Time
fit.time = lm(TimeSeconds ~ Gender+DistanceMeters,data = train)

# Linear Regression Model to Estimate Distance
fit.time2 = lm(TimeSeconds~Gender+DistanceMeters+Place, data = train)

# Prediction of time using time model
pred.time = predict(fit.time, test, interval = "prediction")

# Prediction of distance using distance model
pred.time2 = predict(fit.time2, test, interval = "prediction")
```

## References

(1) Liu, Yuanlong & Schutz, Robert. (1998). Prediction Models for Track and Field Performances. Measurement in Physical Education and Exercise Science. 2. 205-223. https://doi.org/10.1207/s15327841mpee0204_2

(2) James, Nick & Menzies, Max & Bondell, Howard. (2021). In search of peak human athletic potential: A mathematical investigation. https://arxiv.org/pdf/2109.13517.pdf

(3) Blythe, D. A., & Király, F. J. (2016). Prediction and Quantification of Individual Athletic Performance of Runners. PloS one, 11(6), e0157257. https://doi.org/10.1371/journal.pone.0157257

```
sessionInfo()
```

```
## R version 4.2.1 (2022-06-23 ucrt)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 22621)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_United States.utf8
## [2] LC_CTYPE=English_United States.utf8
## [3] LC_MONETARY=English_United States.utf8
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.utf8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] caTools_1.18.2 knitr_1.41     ggplot2_3.4.0  car_3.1-1      carData_3.0-5
##
## loaded via a namespace (and not attached):
##  [1] compiler_4.2.1   pillar_1.8.1     bitops_1.0-7     tools_4.2.1
##  [5] digest_0.6.30    evaluate_0.18    lifecycle_1.0.3  tibble_3.1.8
##  [9] gtable_0.3.1     pkgconfig_2.0.3  rlang_1.0.6      cli_3.3.0
## [13] DBI_1.1.3        rstudioapi_0.14  yaml_2.3.6       xfun_0.35
## [17] fastmap_1.1.0    withr_2.5.0      stringr_1.5.0    dplyr_1.0.10
## [21] generics_0.1.3   vctrs_0.5.1      grid_4.2.1       tidyselect_1.2.0
## [25] glue_1.6.2       R6_2.5.1         fansi_1.0.3      rmarkdown_2.18
## [29] magrittr_2.0.3   scales_1.2.1     htmltools_0.5.3  assertthat_0.2.1
## [33] abind_1.4-5      colorspace_2.0-3 utf8_1.2.2       stringi_1.7.8
## [37] munsell_0.5.0
```