IST 707

Final Project Proposal

Mark Roberts

1. INTRODUCTION

Nearly 30,000 people run the Boston Marathon, normally held yearly on Patriots Day. Patriots Day is a local holiday celebrated in Massachusetts and Maine (which was formerly a part of Massachusetts), on the third Monday in April – the holiday is essentially Veterans Day but for pre-Independence veterans of the Revolutionary War. In Boston, the day is celebrated with near-religious fervor, an unofficial beginning of Spring. Men, women, younger and older adults come from all over the world, but many come from our own communities. Most Bostonians know a group of people 'training for the big day.' As the oldest (first event held in 1897) and one of the most prestigious (it is one of the six World Marathon Majors[1]), the Boston Marathon attracts a broad spectrum of runners, from Olympic-level athletes to somewhat casual running enthusiasts. The organization which sponsors the race, the Boston Athletic Association ("BAA") pays respect to its non-profit/amateur founding mission by allowing runners to run for charity[2]; if a runner does not "qualify" for the race conventionally, the BAA maintains spots for runners to run for charity. This adds to the diversity of the population of runners. The questions this analysis will seek to explore is whether any insights can be gained from on the success of a runner when applied to the dimensions of age/generation, gender, and nationality.

2. DATA ACQUISITION

About the Data

The data is made up of 26,657 observations (=runners) of 20 variables:

[1] See, https://www.worldmarathonmajors.com/six-star.
[2] See, https://www.boston.com/sports/boston-marathon/2019/03/15/donate-marathon-charity-teams/.

```
> str(BAA2019Results)
'data.frame':   26657 obs. of  20 variables:
 $ BibNumber          : chr  "2" "6" "7" "8" ...
 $ FullName           : chr  "Lawrence Cherono" "Lelisa Desisa" "Kenneth Kipkemoi" "Felix
andie" ...
 $ SortName           : chr  "Cherono, Lawrence" "Desisa, Lelisa" "Kipkemoi, Kenneth" "Kar
ie, Felix" ...
 $ AgeOnRaceDay       : chr  "30" "29" "34" "32" ...
 $ Gender             : chr  "M" "M" "M" "M" ...
 $ City               : chr  "Eldoret" "Ambo" "Eldoret" "Iten" ...
 $ StateAbbrev        : chr  NA NA NA NA ...
 $ StateName          : chr  NA NA NA NA ...
 $ Zip                : chr  NA NA NA NA ...
 $ CountryOfResAbbrev: chr  "KEN" "ETH" "KEN" "KEN" ...
 $ CountryOfResName   : chr  "Kenya" "Ethiopia" "Kenya" "Kenya" ...
 $ CountryOfCtzAbbrev: chr  "KEN" "ETH" "KEN" "KEN" ...
 $ CountryOfCtzName   : chr  "Kenya" "Ethiopia" "Kenya" "Kenya" ...
 $ OfficialTime       : chr  "2:07:57" "2:07:59" "2:08:07" "2:08:54" ...
 $ RankOverall        : int  1 2 3 4 5 6 7 8 9 10 ...
 $ RankOverGender     : int  1 2 3 4 5 6 7 8 9 10 ...
 $ RankOverDivision   : chr  "1" "2" "3" "4" ...
 $ EventGroup         : chr  "Runners" "Runners" "Runners" "Runners" ...
 $ SubGroupLabel      : chr  NA NA NA NA ...
 $ SubGroup           : chr  NA NA NA NA ...
```

- **BibNumber:** Identification number used by BAA as a unique identifier for each runner. Variable not required for any data analytics modeling and will be dropped.
- **FullName:** Loosely defined as complete name combination of "First", "Middle" and "Last" names. As an international event, this system could be complicated given different cultures' nomenclature customs. Variable is not needed for data analytics modeling and will be dropped.
- **SortName:** Similar to FullName, with "Last" name first. Again, not needed for modeling and will be dropped.
- **AgeOnRaceDay:** Age of the runner on day of race. Runners may be 17 when registering but must be at least 18 on race day. There is no upper age limit. As a variable, it is numeric and continuous, but could also be ordinal and used to discretize. BAA offers 11 age group classes: 18-34, 35-39, 40-44, 45-49, 50-54, 55-59, 60-64, 65-69, 70-74, 75-79, 80+. 11 is too many classifiers; thus, this analysis will simply discretize to four bins based on quartiles.
- **City, StateAbbrev, StateName:** City, State designations. As many runners are international, this observation contains a fair amount of NA's. Fortunately, it will not be used in any analysis and will be dropped..
- **CountryOfRes/CtzAbbrev & CountryOfRes/CtzName:** Observations for the full country name and its attendant abbreviation. The dichotomous Res vs. Ctz relationship allows BAA to differentiate Citizenship designation from Residence. In most, cases, particularly for casual runners these values will likely be the same. For international "elite" or professionals, a runner may be a Citizen of say Kenya, but may 'train' in the US or UK.

This analysis will explore the records where there is a difference but will decide on "CountryOfCtz" as the focus of analysis of runners' provenance. The variable will be used to discretize runners into regional classifiers.

- **OfficialTime:** The official finish time of each runner entered in "hh:mm:ss" format. This is the one true continuous variable in the data set and is the key target for any linear regression models. The observation will require transformation into seconds.
- **RankOverall, RankOverGender, RankOverDivision:** Rank number for each runner based on overall, within gender class and within one of the 11 age classes. This analysis will only consider RankOverall whereby it will discretize the set into quadrants "Top 25%", "Upper Middle 25%", "Lower Middle 25%", and "Bottom 25%". As this variable sorts the data set in order, steps will be taken to "shuffle" the data set so that results are not skewed by position in the data set, but this marker will allow a somewhat general performance distinction for every runner, without a precise rank designation.
- **EventGroup, SubGroupLabel, SubGroup:** Participation in the Boston Marathon is not limited to runners; there are groupings for wheelchair, handcycles, vision impaired, etc. This data set and analysis only covers bipedal runners. Thus, this variable will be deleted.

```
> Keeps <- c("AgeOnRaceDay","Gender","CountryOfResName","CountryOfCtzName","OfficialTime","RankOverall")
> BAA2019Results <- BAA2019Results[Keeps]
> str(BAA2019Results)
'data.frame':   26657 obs. of  6 variables:
 $ AgeOnRaceDay    : chr  "30" "29" "34" "32" ...
 $ Gender          : chr  "M" "M" "M" "M" ...
 $ CountryOfResName: chr  "Kenya" "Ethiopia" "Kenya" "Kenya" ...
 $ CountryOfCtzName: chr  "Kenya" "Ethiopia" "Kenya" "Kenya" ...
 $ OfficialTime    : chr  "2:07:57" "2:07:59" "2:08:07" "2:08:54" ...
 $ RankOverall     : int  1 2 3 4 5 6 7 8 9 10 ...
```

3. EXPERIMENTAL DESIGN

Cleaning Transformation

Checking for NA's

After removing unnecessary and problematic columns which contain all of the NA's, the data set conveniently has no NA's which need to be dealt with.

```
> sum(is.na(BAA2019Results))
[1] 0
```

However, in checking for unique values for classification, a problem in "Gender" is discovered. It would appear that one record was entered incorrectly either through a typo or some other human/data entry error.

```
> unique(BAA2019Results$Gender)
[1] "M"                "F"                "Leighton Buzzard"
```

"Leighton Buzzard" is a town. The runner in question has Bib Number 28913.

```
> BAA2019Results[BAA2019Results$Gender=="Leighton Buzzard",]
      BibNumber AgeOnRaceDay           Gender CountryOfResName CountryOfCtzName OfficialTime RankOverall
26143     28913            F Leighton Buzzard              GBR          5:46:30        26143       11695
```

From a look up at the BAA results website, the correct data was found to be and was corrected in an atomic fashion, one observation updated at at time:

Your search returned 1 result(s).
Criteria: Bib Number 28913, Race Year 2019, sorted by Last Name.

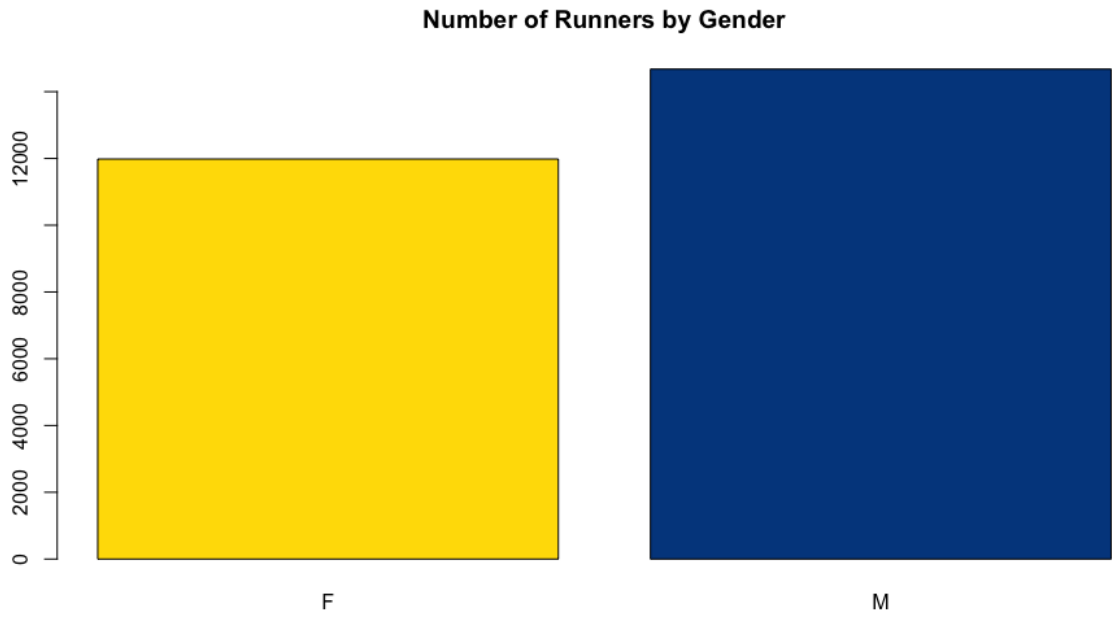
YEAR	BIB	NAME	AGE	M/F	CITY	STATE	COUNTRY
2019	28913	O"Connell, Pamela	62	F	Leighton Buzzard		GBR

Overall	Gender	Division	Official Time	Net Time
26143 / 26657	11695 / 11982	414 / 430	6:04:34	5:46:30

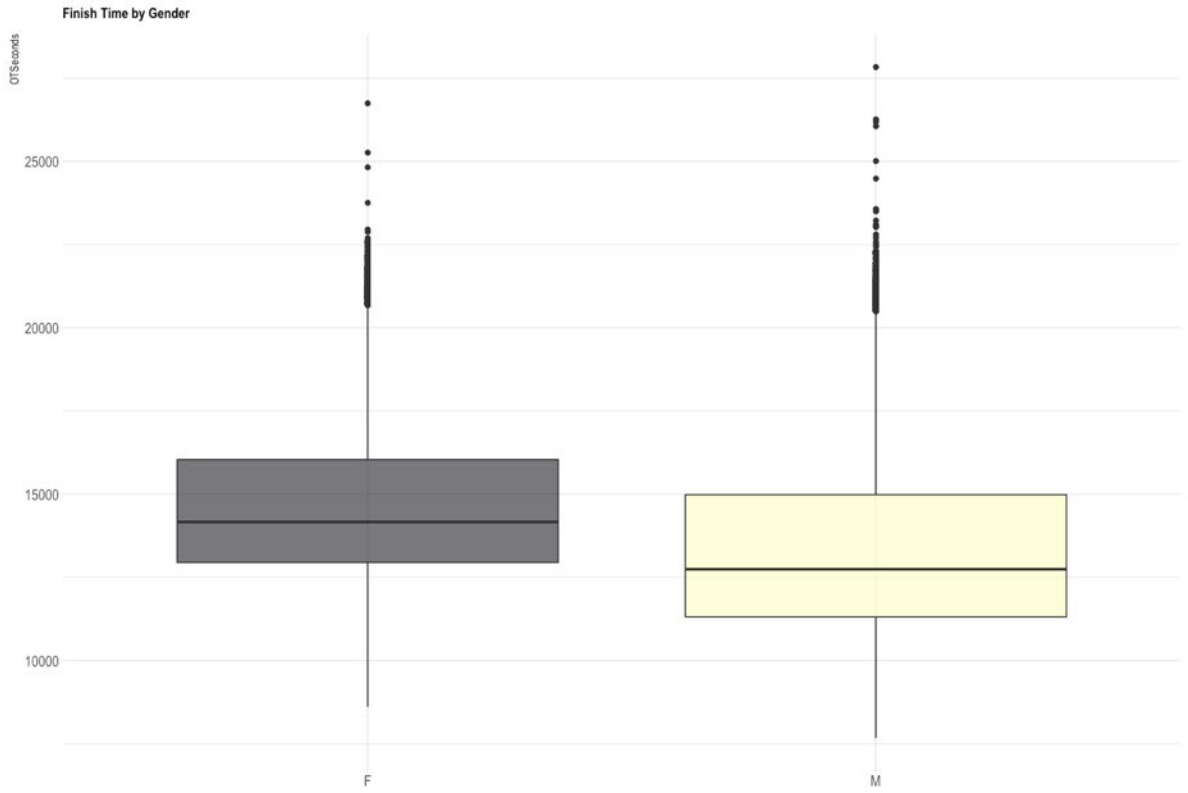
```
> BAA2019Results$Gender[26143]="F"
> BAA2019Results$OfficialTime[26143]="6:04:34"
> BAA2019Results$RankOverall[26143]="26143"
> BAA2019Results$CountryOfCtzName[26143]="United Kingdom"
> BAA2019Results$AgeOnRaceDay[26143]="62"
> BAA2019Results[BAA2019Results$BibNumber=='28913',]
      BibNumber AgeOnRaceDay Gender CountryOfResName CountryOfCtzName OfficialTime RankOverall
26143     28913           62      F              GBR   United Kingdom      6:04:34        26143
```

Once achieving the proper binomial Gender variable, look at the count shows a slightly greater number of Male runners vs Female runners, but not an extreme skew either way:

```
> table(BAA2019Results$Gender)

    F     M
11982 14675
```



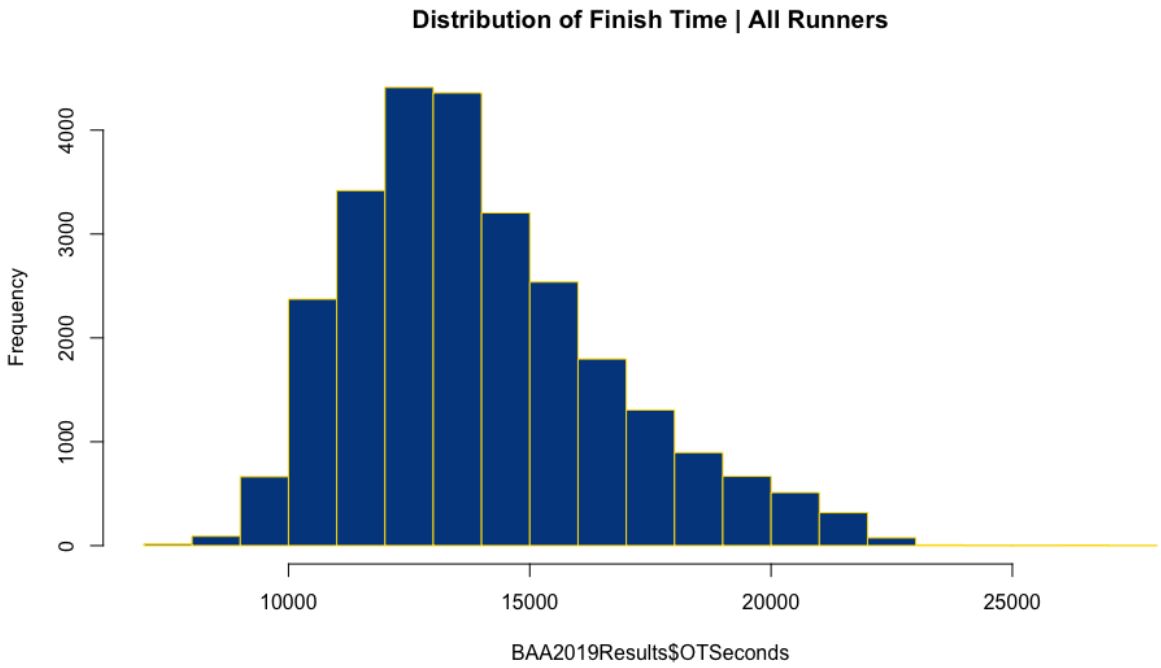
The distribution of completion times by gender shows a density of male population of finishers whereby the median roughly equates to the fastest lower quantile of the female runners, but generally there is overlap between the two distributions. There appears to be a population of outliers beginning ~20,000 seconds for both genders (for finishing time transformation, see below).



Transforming Official Time into Seconds & Creating "Pace" variable

The data set has precious few continuous variables. The key one and probably the most interesting possible dependent variable (were linear regression a model this analysis had planned) is finish time. Unfortunately, result time is currently in "time" format "HH:MM:SS"; using lubridate package, the analysis will transform time into seconds, a numerical value that will allow for regression.

```
> summary(BAA2019Results$OTSeconds)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   7677   12026   13534   13982   15540   27828
```



The distribution of finish times appears to be in a bell-shaped curve, approaching a normal distribution. There's a slight right skew with median value to the left of the mean value. The population size is nearly 30,000 which is somewhat large. If the analysis were to consider more years than just one, the number of observations (=population) would increase and we'd likely see a more normal distribution shape.

Location Mismatches

There are 93 unique CountryOfResName values. There are 109 unique CountryOfCtzName values. This analysis will designate "CountryOfCtzName" for all runner Regional analytics. For point of interest, the accompanying R code provides an analysis of the number of instances in which CountryOrResName and CountryOfCtzName do not match. Since CountryOfResName will be dropped, this analysis will make no further investigation of the mismatches.

```
> length(unique(BAA2019Results$CountryOfResName))
[1] 93
> length(unique(BAA2019Results$CountryOfCtzName))
[1] 109
```

Since it's not a critical distinction for this analysis, a detailed sorted list of countries that "mismatches" and the most frequently occurring instances is only provided in the accompanying RMD file, for reference. However, this discrepancy provides a potential clue that some runners have a "professional" status whereby they may be FROM country X (Ctz) but RESIDE (Res, perhaps for training) in another.

Classification

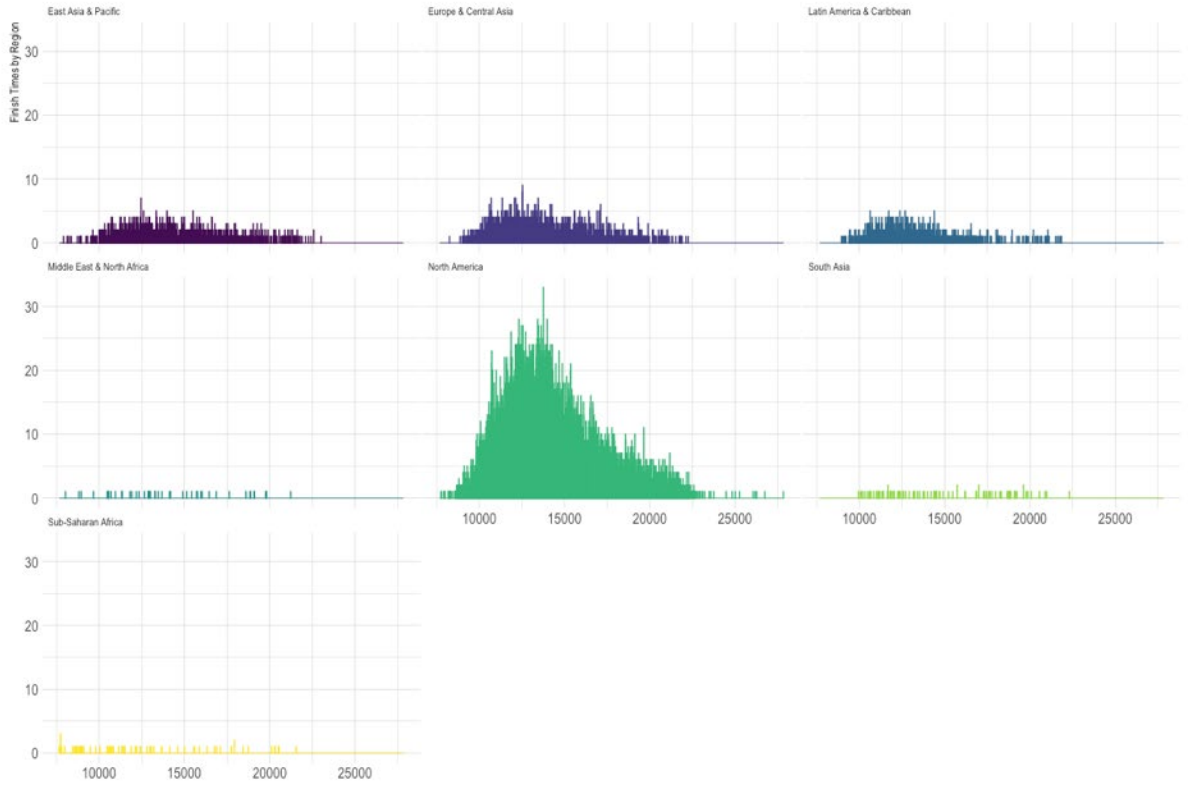
The dataset comes handily equipped with several ready-made classification distinctions: Gender (M/F), Age/Group, Country (Regional). Below describes the process of discretizing the data into classes.

Region Classifier

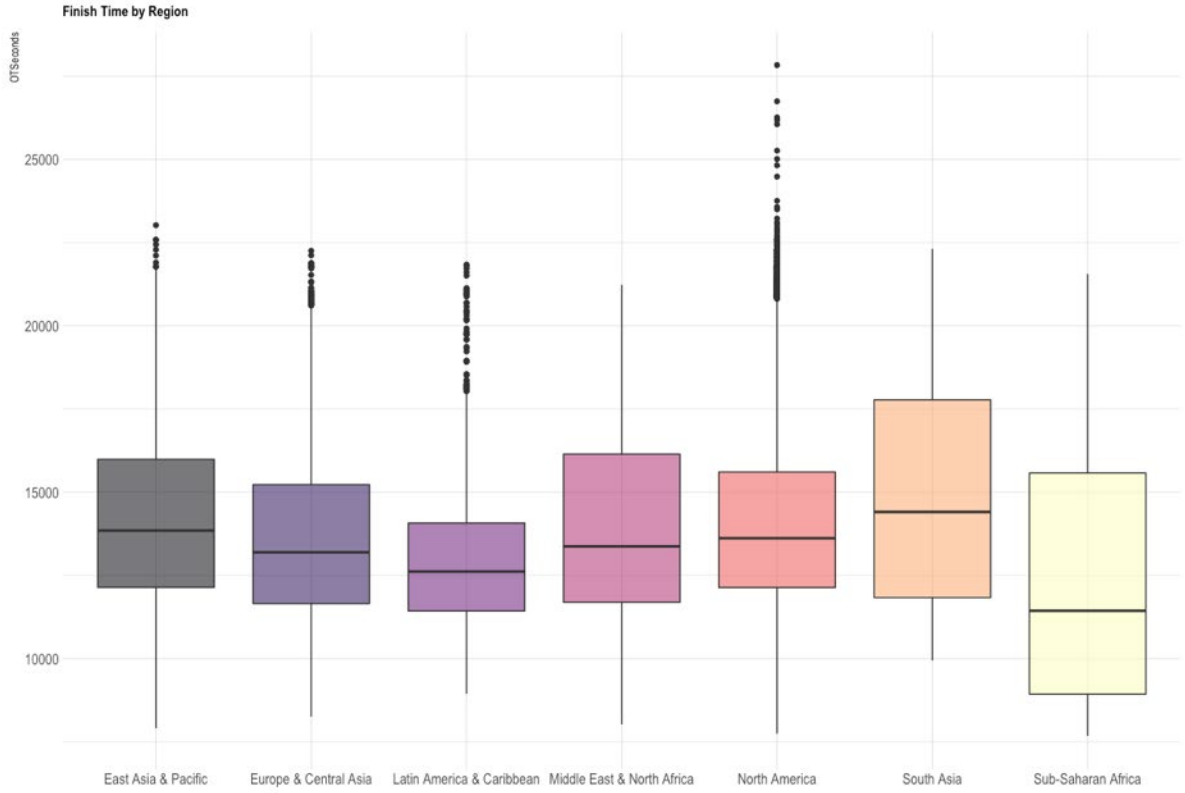
Using the countrycode package, a new variable "Region" is created, taking CountryOfCtzName and applying the appropriate "Region."

```
> (table(BAA2019Results$Region))

      East Asia & Pacific      Europe & Central Asia  Latin America & Caribbean Middle East & North Africa
                     1734                       2823                       1233                         44
             North America                 South Asia         Sub-Saharan Africa
                     20663                      93                         67
```

The overwhelming majority of runners are from North America (20,663). Europe/Central Asia (2,823), East Asia & Pacific (1,734) and Latin America/Caribbean (1,233) each has a considerable number of runners, although if combined amount to just over a 25% of North America's representation. Regions with smaller representation are South Asia (93), Sub-Saharan Africa (67) and Middle East/North Africa (44). As the top runners in Marathons will likely be from Kenya or Ethiopia[3], it will be interesting to see this dynamic play out with a Sub-Saharan Africa Region bucket amounts to just 0.25% of the entire number of runners (26,657).



[3] https://www.runnersworld.com/advanced/a20842528/the-ethiopia-kenya-running-phenomenon/

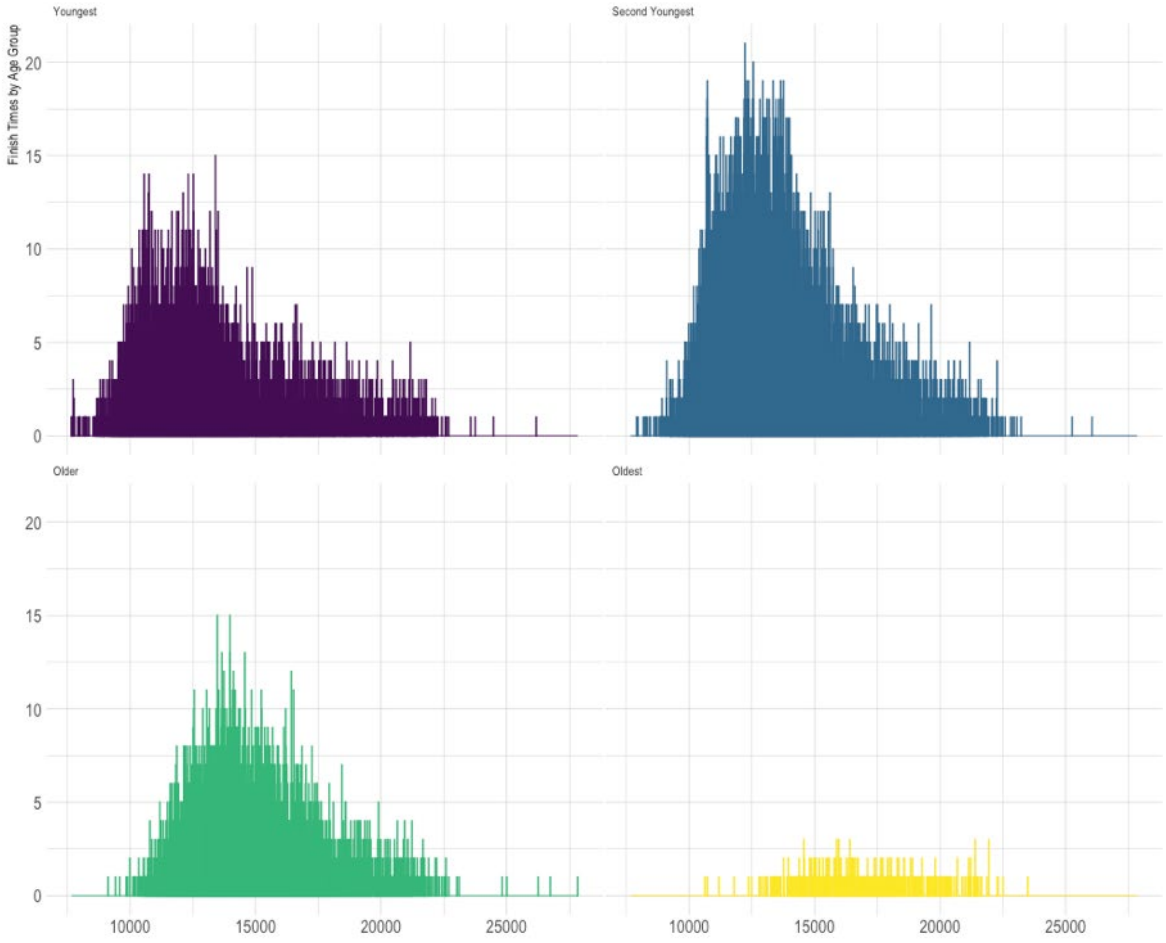


Age Group Classifier

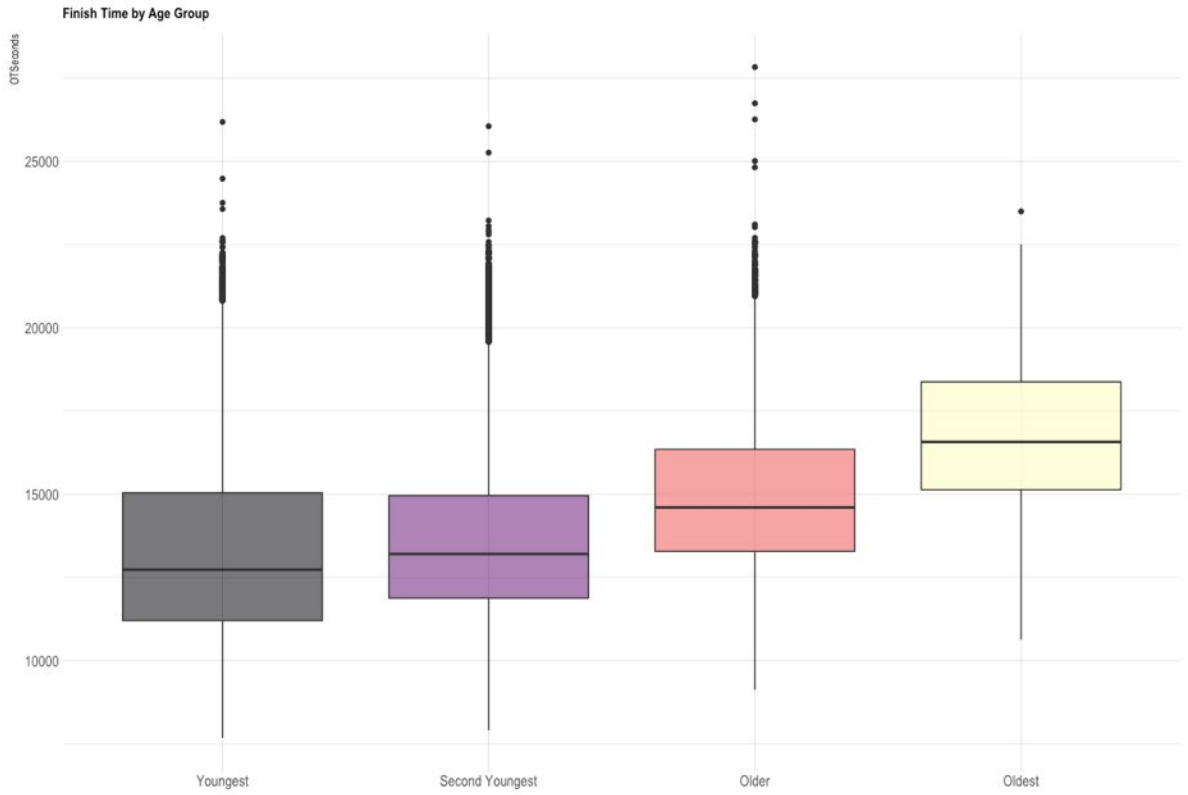
After transforming AgeOnRaceDay to a numeric value, into 4 quartile variables: "Youngest" (7,003 runners, aged 18-34), "Second Youngest" (12,725 runners, aged 35-50), "Older" (6,440 runners, aged 51-66) and "Oldest" (489 runners, aged 67-83).

```
> summary(BAA2019Results$AgeOnRaceDay[(BAA2019Results$AgeQuartiles=="Youngest")])
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   18.0    26.0    29.0    28.4    31.0    34.0
> summary(BAA2019Results$AgeOnRaceDay[(BAA2019Results$AgeQuartiles=="Second Youngest")])
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  35.00   39.00   43.00   42.62   46.00   50.00
> summary(BAA2019Results$AgeOnRaceDay[(BAA2019Results$AgeQuartiles=="Older")])
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  51.00   53.00   56.00   56.75   60.00   66.00
> summary(BAA2019Results$AgeOnRaceDay[(BAA2019Results$AgeQuartiles=="Oldest")])
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  67.00   68.00   69.00   69.98   71.00   83.00
```

```
> table(BAA2019Results$AgeQuartiles)

      Youngest Second Youngest          Older         Oldest
          7003           12725           6440            489
```



Obviously, the "Oldest" category has the smallest group at 489. Oldest's distribution of Finish Times amounts shows a somewhat uniform distribution. Youngest, Second Youngest and Older have more normal distributions, all skewing right, with Second Youngest showing the greatest frequency around the median, naturally as the largest group. One might expect the youngest group to have the largest number of runners, but instead it's the Second Youngest group. Some factors that might explain this might be age limit (must be 18) and/or economic circumstances. A younger runner may not have the resources to pay for the race or accommodate a training regimen. Also, running a marathon is something a runner "works up to". It may be an interesting question to see how many runners are first time runners, what age those runners are and what had been their running resume before the Marathon (5K/10K/Half-Marathon). Unfortunately, that insight can't be inferred from the dataset as it is.



Rank Classifier Classifier

RankOverall was used to discretize four groups by 25% cuts: "Top 25%" (6,665 runners), "Upper Middle 25%" (6,664 runners), "Lower Middle 25%" (6,664 runners), and "Bottom 25%" (6,664 runners).

Histograms are provided in the R file which show with a smaller population size, each distribution is approaching a uniform shape. Using a runner's exact position in the race would not be very productive and could lead to some overfitting prediction issues, this analysis wanted to include a distinction for each runner in terms of general position in the race rank order.

Achieved Qualifying Time Classifier

Every few years, BAA puts out a list of qualifying times ("QT"). It's broken up by 11 age groups and then divided into Male and Female runners, thus creating 22 qualification categories. To outright qualify for the Boston Marathon, a runner must have run a race in the proximate years previously that satisfies the QT according to the runner's Gender and Age Group.

This analysis created a new data frame reflecting the QT for 2019:

	MenQT	WomenQT	MenQTSeconds	WomenQTSeconds
18-34	3:00:00	3:30:00	10800	12600
35-39	3:05:00	3:35:00	11100	12900
40-44	3:10:00	3:40:00	11400	13200
45-49	3:20:00	3:50:00	12000	13800
50-54	3:25:00	3:55:00	12300	14100
55-59	3:35:00	4:05:00	12900	14700
60-64	3:50:00	4:20:00	13800	15600
65-69	4:05:00	4:35:00	14700	16500
70-74	4:20:00	4:50:00	15600	17400
75-79	4:35:00	5:05:00	16500	18300
80 and Over	4:50:00	5:50:00	17400	21000

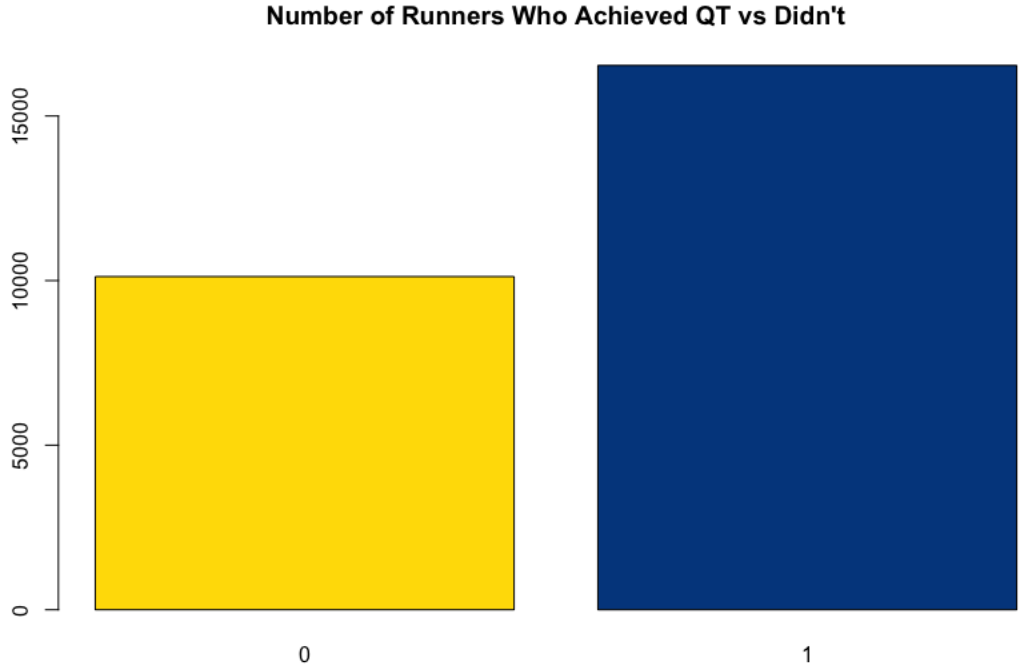
Once transformed into a numerical "seconds" variable, each observation was created as its own variable:

```
#Qualifyingg Times by Gender/Age Group Each As Variable
M1834QT <- BAA2019QT[1,3]
M3539QT <- BAA2019QT[2,3]
M4044QT <- BAA2019QT[3,3]
M4549QT <- BAA2019QT[4,3]
M5054QT <- BAA2019QT[5,3]
M5559QT <- BAA2019QT[6,3]
M6064QT <- BAA2019QT[7,3]
M6569QT <- BAA2019QT[8,3]
M7074QT <- BAA2019QT[9,3]
M7579QT <- BAA2019QT[10,3]
M80PLUSQT <- BAA2019QT[11,3]
F1834QT <- BAA2019QT[1,4]
F3539QT <- BAA2019QT[2,4]
F4044QT <- BAA2019QT[3,4]
F4549QT <- BAA2019QT[4,4]
F5054QT <- BAA2019QT[5,4]
F5559QT <- BAA2019QT[6,4]
F6064QT <- BAA2019QT[7,4]
F6569QT <- BAA2019QT[8,4]
F7074QT <- BAA2019QT[9,4]
F7579QT <- BAA2019QT[10,4]
F80PLUSQT <- BAA2019QT[11,4]
```

Then a two-part process created a "QTClass" variable reflecting the appropriate grouping. Then, each observation that equated a particular QTClass AND was less than/equal to the matching QT seconds variable as assigned to a Boolean 1, or else assigned to 0.

```
#Create Classification Variable of Each QT Group by Gender/Age
BAA2019Results$QTClass[(BAA2019Results$Gender=='M' & BAA2019Results$AgeOnRaceDay>=18 & BAA2019Results$AgeOnRaceDay<=34)] <- "M1834"
BAA2019Results$QTClass[(BAA2019Results$Gender=='M' & BAA2019Results$AgeOnRaceDay>=35 & BAA2019Results$AgeOnRaceDay<=39)] <- "M3539"
BAA2019Results$QTClass[(BAA2019Results$Gender=='M' & BAA2019Results$AgeOnRaceDay>=40 & BAA2019Results$AgeOnRaceDay<=44)] <- "M4044"
BAA2019Results$QTClass[(BAA2019Results$Gender=='M' & BAA2019Results$AgeOnRaceDay>=45 & BAA2019Results$AgeOnRaceDay<=49)] <- "M4549"
BAA2019Results$QTClass[(BAA2019Results$Gender=='M' & BAA2019Results$AgeOnRaceDay>=50 & BAA2019Results$AgeOnRaceDay<=54)] <- "M5054"
BAA2019Results$QTClass[(BAA2019Results$Gender=='M' & BAA2019Results$AgeOnRaceDay>=55 & BAA2019Results$AgeOnRaceDay<=59)] <- "M5559"
> table(BAA2019Results$QTClass)

 F1834  F3539  F4044  F4549  F5054  F5559  F6064  F6569  F7074  F7579 F80PLUS  M1834  M3539  M4044  M4549  M5054
  3867   1793   1840   1835   1200    829    430    141     37      9      1   3136   1956   1989   2447   1858
 M5559  M6064  M6569  M7074  M7579 M80PLUS
  1552   1086    469    132     39     11

BAA2019Results$AchievedQT[(BAA2019Results$QTClass=="M1834"&BAA2019Results$OTSeconds>=M1834QT)] <- 1
BAA2019Results$AchievedQT[(BAA2019Results$QTClass=="M3539"&BAA2019Results$OTSeconds>=M3539QT)] <- 1
BAA2019Results$AchievedQT[(BAA2019Results$QTClass=="M4044"&BAA2019Results$OTSeconds>=M4044QT)] <- 1
BAA2019Results$AchievedQT[(BAA2019Results$QTClass=="M4549"&BAA2019Results$OTSeconds>=M4549QT)] <- 1
BAA2019Results$AchievedQT[(BAA2019Results$QTClass=="M5054"&BAA2019Results$OTSeconds>=M5054QT)] <- 1
```

```
> table(BAA2019Results$AchievedQT)

    0     1
10122 16535
```




```
> table(BAA2019ModelsDF$AchievedQT)

   No   Yes
10122 16535
> (YesProb <- 16535/sum(table(BAA2019ModelsDF$AchievedQT)))
[1] 0.6202874
> (NoProb <- 10122/sum(table(BAA2019ModelsDF$AchievedQT)))
[1] 0.3797126
```

This variable will serve as the object classifier in the models I will be running. There is a slight skew in the data: 62% Yes/AchievedQT and 38% No/AchievedQT. The difference is not so stark that the analysis may not proceed. Exploration of balancing the data set may be an option to be revisited at a later stage.

4. MODEL BUILDING

Planned Models

Next, the analysis will analyze the data set to determine whether a runner achieved qualifying time based on gender, rank, age group and region. The classification models being used are:

- **Decision Tree**
- **Support Vector Machines (SVM)**
- **Random Forest (RF)**

Model Preparation

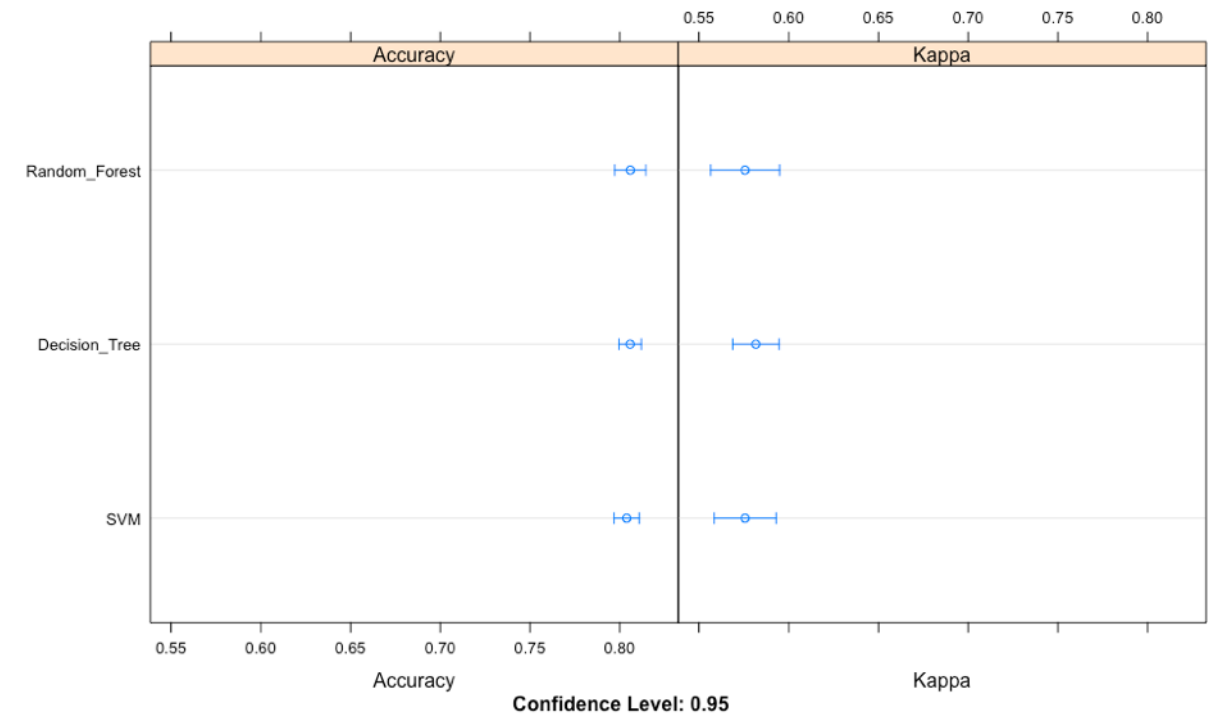
After creating a data frame of only the variables needed for the analysis and removing any that may not be independent (RankOrder/OTSeconds, AgeQuartile/AgeOnRaceDay), the analysis randomizes the rows (which hitherto have been in rank order and therefore need to be 'shuffled'). Then using a 70/30 split, a "train" and "test" set are created.

```
set.seed(123)
ind <- sample(nrow(BAA2019ModelsDF), replace = T)
BAA2019_1 <- BAA2019ModelsDF[ind,]

dt = sort(sample(nrow(BAA2019_1), nrow(BAA2019_1)*.7))
train<-BAA2019_1[dt,]
test<-BAA2019_1[-dt,]
```

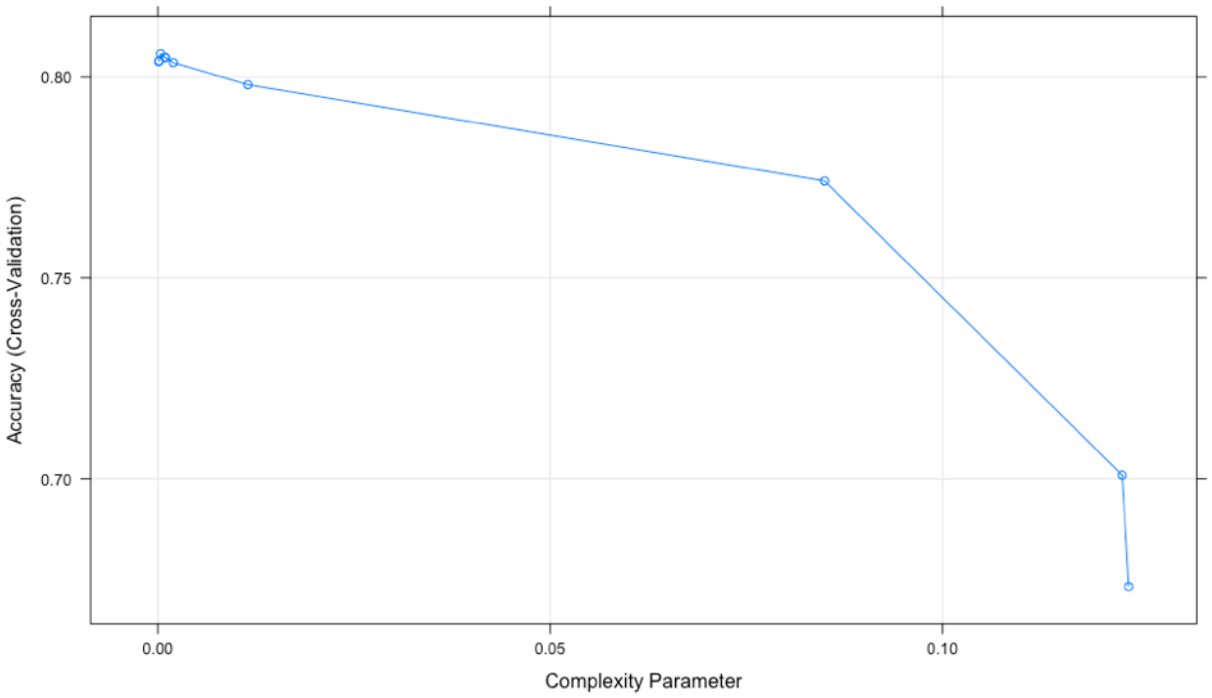
An initial set of models are created using CV with K-Fold of 10 as a conventionally recognized starting point. Each model is then run with a tuneLength definision of 10.

```
# Creating a control with cross validation starting at 10
control <- trainControl(method ='cv',number = 10)
metric <-  "Accuracy" # Metric for comparison will be accuracy for this project

## Train models to predict whether Runner Achieved QT

tree.model1 <- train(AchievedQT ~ ., data = train, method="rpart", metric=metric, trControl=
                     tuneLength = 10) # Decision Tree

svm.model1 <- train(AchievedQT ~ ., data = train, method="svmRadial",metric=metric,trControl
                    tuneLength = 10) # Support Vector Machine (SVM)

rf.model1 <- train(AchievedQT ~ ., data = train, method="rf", metric=metric, trControl=control
                   tuneLength = 10) # Random Forest
```

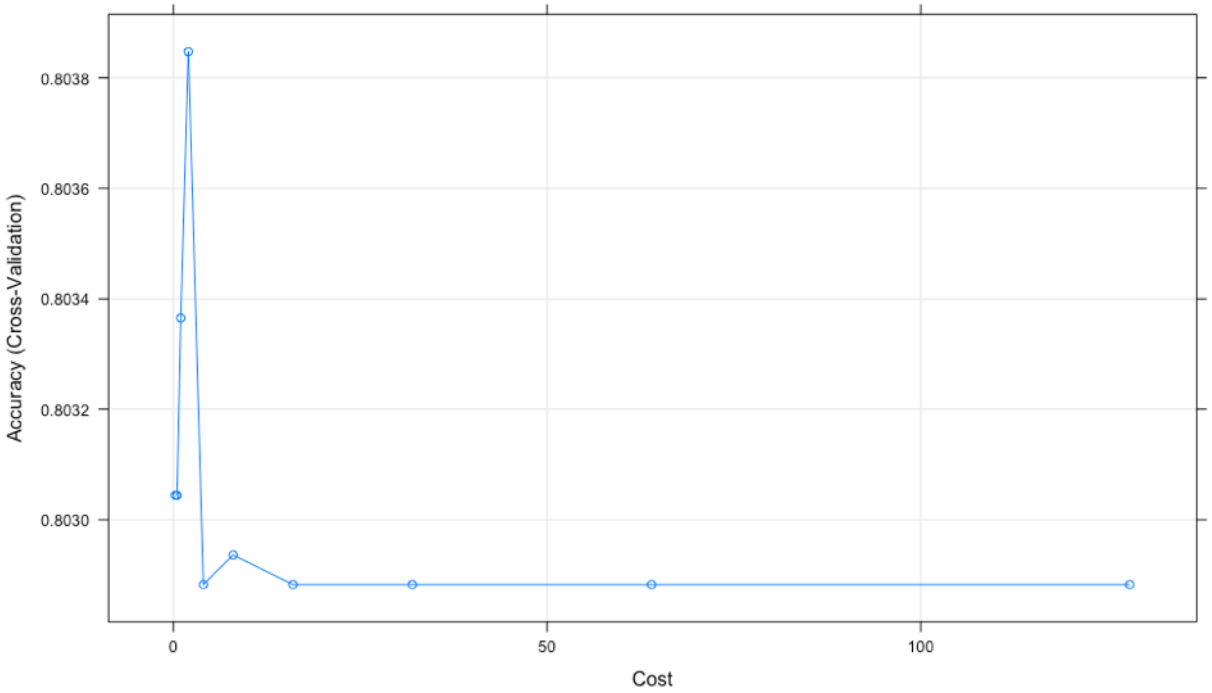
Model Evaluation



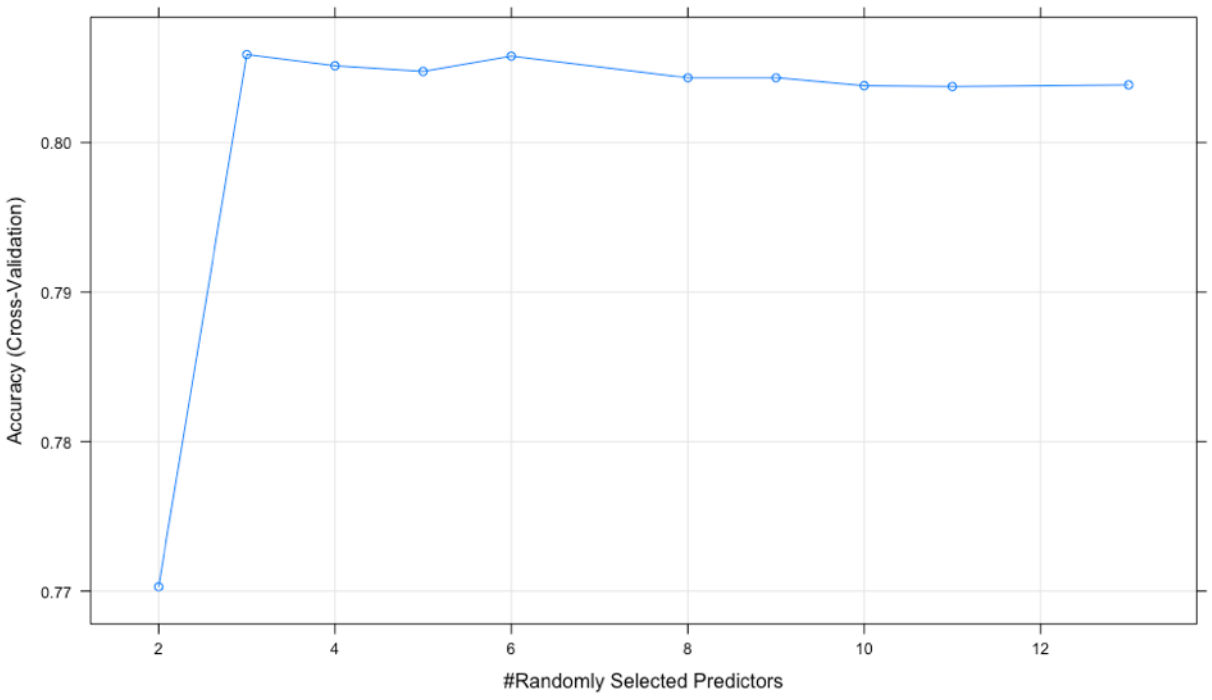
A look at all three models' accuracy shows them all at about 80%, with a tighter confidence interval around Decision Tree. All three have a Cohen's Kappa score between 55-60%, which has a "moderate" significance. It's so-so. Better than a coinflip but not by much. Decision Tree has the highest Kappa at 58%.



Decision Tree's accuracy quality degrades as the model complexity increases, with a gradual decline beginning around 2-3, falling starkly at around 7-8. It may be worth trying to limit the k-fold number and adjusting the tuneLength parameter.

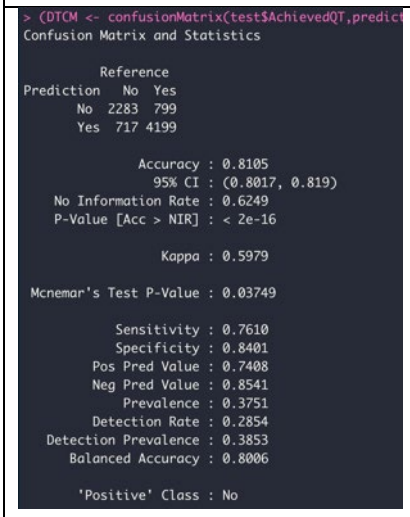
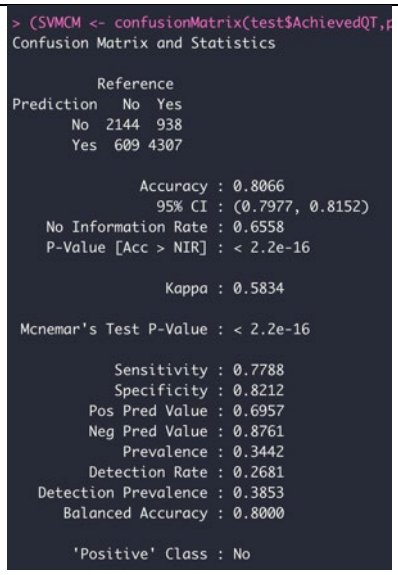
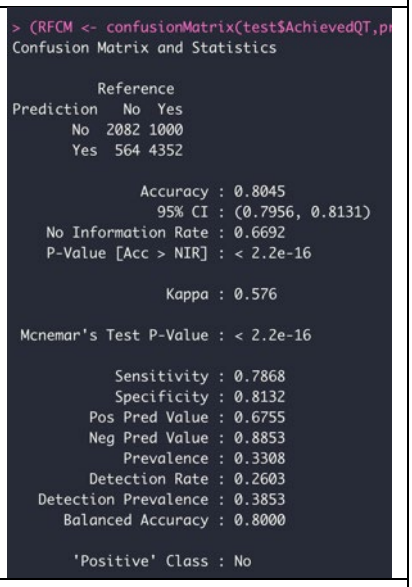


SVM's accuracy peaks early and drops precipitously just after 10. There's a second bump up around 15 and then a flatline. Again, an attempt will be made to adjust parameters to improve accuracy.



As with the other two models, the accuracy performance of Random Forest jumps at about 3 but then promptly levels off. As with the other, models parameter tuning will be attempted.

Model Prediction

Decision Tree Conf.Matrix	SVM Conf.Matrix	RF Conf.Matrix
		

All three models have an accuracy level of approximately 80% with Decision Tree slightly in the lead with 81.05% on prediction. Decision Tree also has the best Kappa value at .5979, just edging out SVM at .5834. All three have a low P-Value in McNemar's Test, all under 0.05 alpha level but Decision Tree is just barely under at 0.03749.

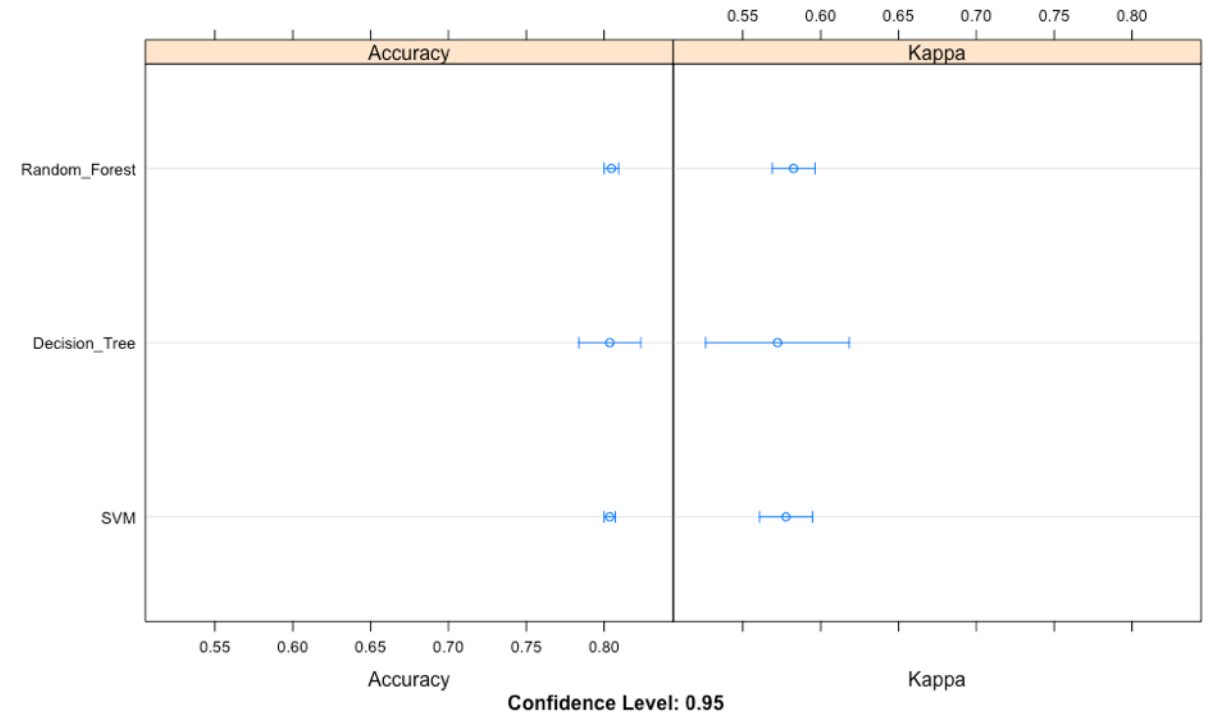
Random Forest has a slightly higher Sensitivity and No-Information rate. However, it's difficult to pick a clear winner. All three "get it right" and "get it wrong" but not all at the same time, as evidenced below:

```
> head(predictionTest,10)
    dt svm random_f test.AchievedQT
1   No Yes      Yes             Yes
2   No  No       No              No
3   No  No       No              No
4  Yes Yes      Yes             Yes
5   No Yes      Yes             Yes
6  Yes Yes      Yes              No
7  Yes Yes      Yes             Yes
8   No  No       No              No
9   No  No       No              No
10 Yes Yes      Yes             Yes
```

Model Tuning

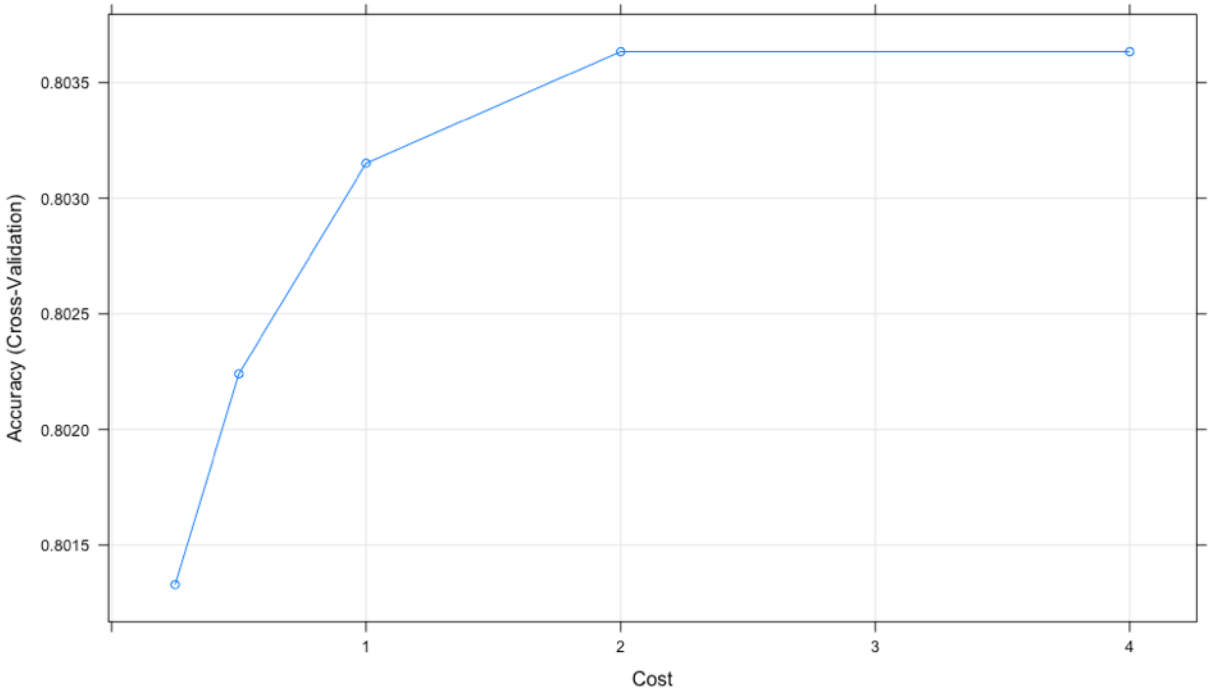
After finding a cost in higher CV K-Fold values, a new set of models are created using CV with K-Fold of 3, since each model showed a drop-off in accuracy beyond this number. Each model is then run with a tuneLength definition of 5.

Tuning Model Evaluation

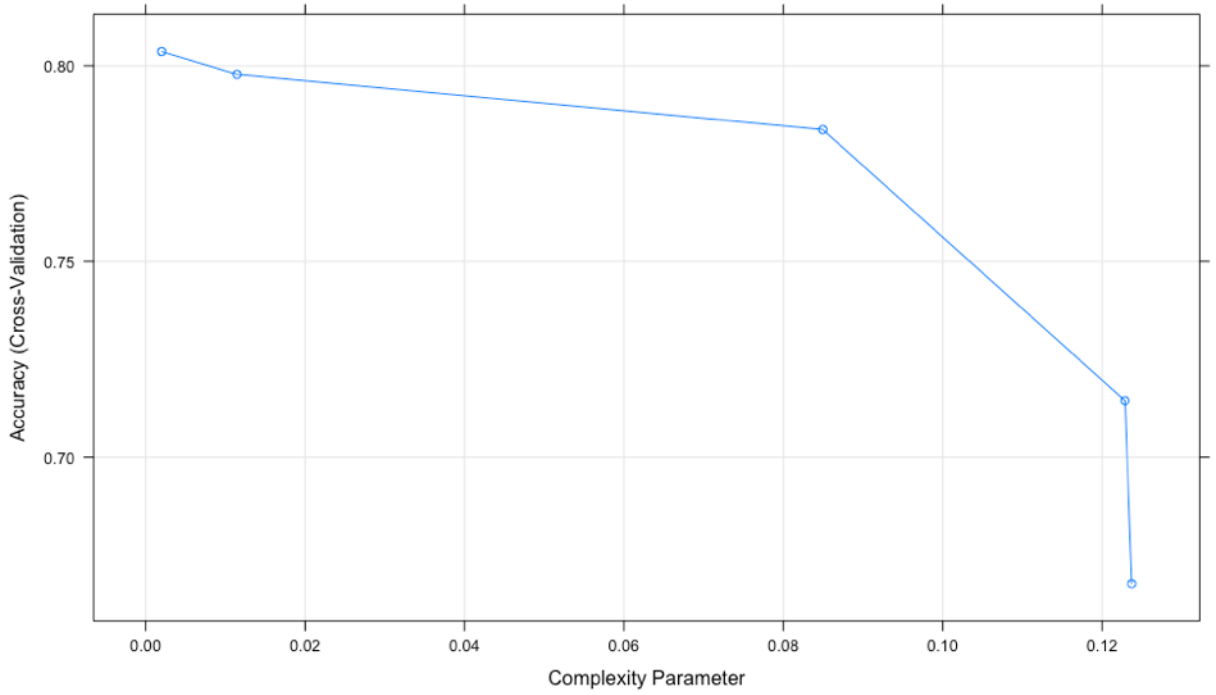


A look at all three models' accuracy shows them all three models still around 80, but Decision Tree has improved somewhat. Of note, is the better Kappa values with Decision Tree's confidence interval running to 60+, with brings the model out of "moderate" significance to "good." Albeit, modest, there has been an improvement.

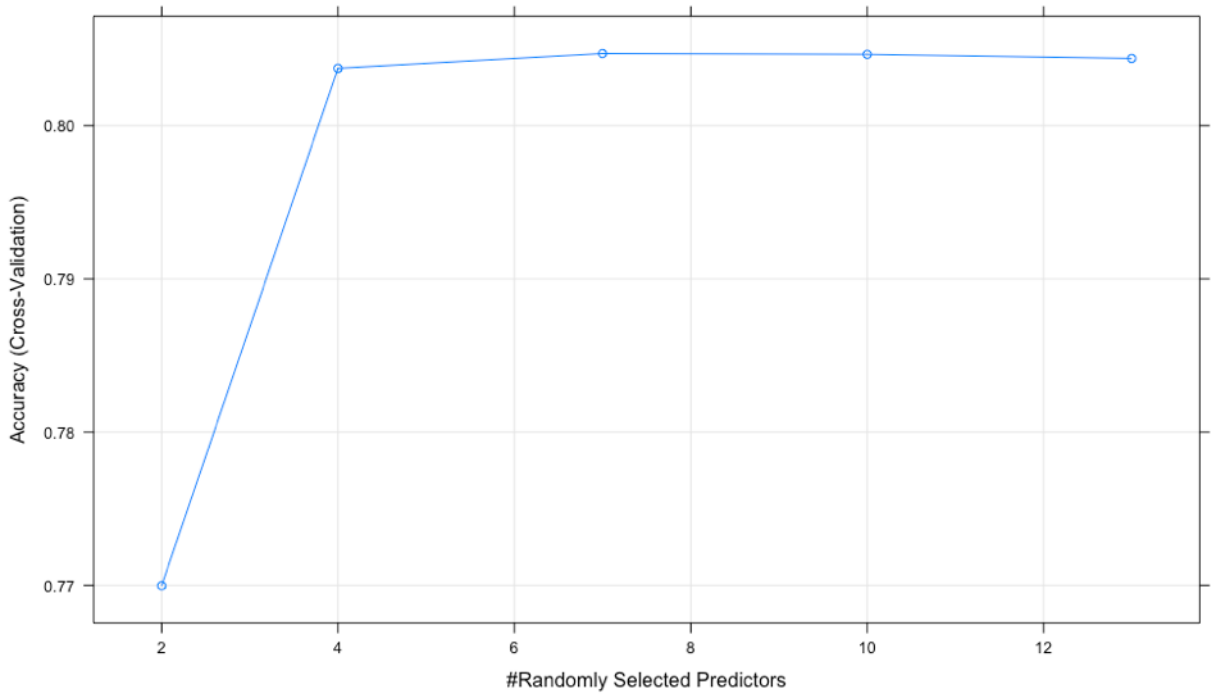
Decision Tree Accuracy Plot



SVM Accuracy Plot

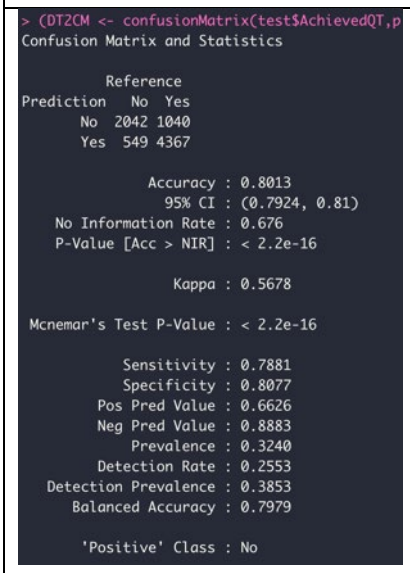
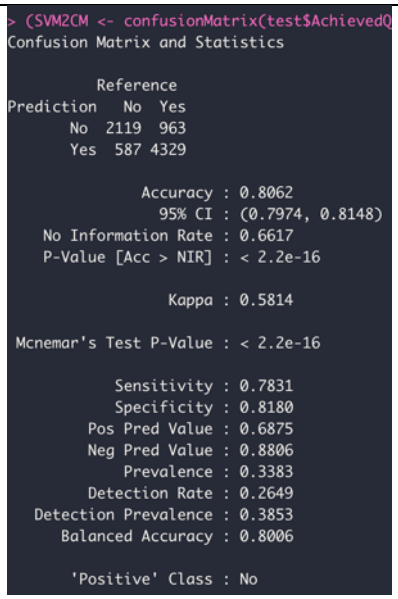
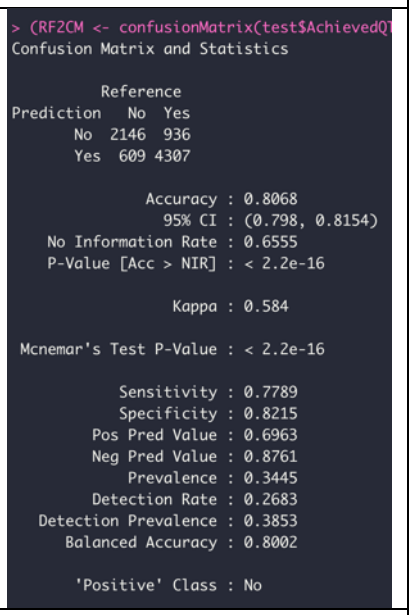


RF Accuracy Plot



Decision Tree shows a flattening at 2 and Random Forest at 4. It's unlikely that additional tuning will be able to improve the accuracy levels of the models in any significant way.

Model Prediction

Decision Tree Conf.Matrix	SVM Conf.Matrix	RF Conf.Matrix
		

```
> (DT2CM <- confusionMatrix(test$AchievedQT,p
Confusion Matrix and Statistics

          Reference
Prediction   No  Yes
       No  2042 1040
       Yes  549 4367

               Accuracy : 0.8013
                 95% CI : (0.7924, 0.81)
    No Information Rate : 0.676
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.5678

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.7881
            Specificity : 0.8077
         Pos Pred Value : 0.6626
         Neg Pred Value : 0.8883
             Prevalence : 0.3240
         Detection Rate : 0.2553
   Detection Prevalence : 0.3853
      Balanced Accuracy : 0.7979

       'Positive' Class : No
```

```
> (SVM2CM <- confusionMatrix(test$AchievedQ
Confusion Matrix and Statistics

          Reference
Prediction   No  Yes
       No  2119  963
       Yes  587 4329

               Accuracy : 0.8062
                 95% CI : (0.7974, 0.8148)
    No Information Rate : 0.6617
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.5814

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.7831
            Specificity : 0.8180
         Pos Pred Value : 0.6875
         Neg Pred Value : 0.8806
             Prevalence : 0.3383
         Detection Rate : 0.2649
   Detection Prevalence : 0.3853
      Balanced Accuracy : 0.8006

       'Positive' Class : No
```

```
> (RF2CM <- confusionMatrix(test$AchievedQT
Confusion Matrix and Statistics

          Reference
Prediction   No  Yes
       No  2146  936
       Yes  609 4307

               Accuracy : 0.8068
                 95% CI : (0.798, 0.8154)
    No Information Rate : 0.6555
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.584

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.7789
            Specificity : 0.8215
         Pos Pred Value : 0.6963
         Neg Pred Value : 0.8761
             Prevalence : 0.3445
         Detection Rate : 0.2683
   Detection Prevalence : 0.3853
      Balanced Accuracy : 0.8002

       'Positive' Class : No
```

All three models have an accuracy level of approximately 80% this time Random Forest slightly in the lead with 80.68% on prediction, although overall not an improvement over the initial models. Kappa values went down overall, the highest being Random Forest at .584. Of note, all retain a low P-Value in McNemar's Test, all under 0.05 alpha level this time including Decision Tree where the P-Value improved somewhat.

Decision Tree this time has the slightly higher Sensitivity and No-Information rate. However, it's difficult to pick a clear winner. All three still "get it right" and "get it wrong" but not all at the same time, as evidenced below:

```
> head(predictionTest2, 10)
   dt2 svm2 random_f2 test.AchievedQT
1  Yes  Yes       Yes              Yes
2   No   No        No               No
3   No   No        No               No
4  Yes  Yes       Yes              Yes
5  Yes  Yes       Yes              Yes
6  Yes  Yes       Yes               No
7  Yes  Yes       Yes              Yes
8   No   No        No               No
9   No   No        No               No
10 Yes  Yes       Yes              Yes
```

5. CONCLUSION

The Boston Marathon brings together runners from all over the globe. As such, the population is diverse. The accuracy and performance of the models are "so-so". 80% accuracy with ~60 Kappa is a decent result but leaves room for improvement. Things that cannot be known from the data set are very likely to impact the outcome. These include a runner's physique (height/weight), how first marathon or number of marathons previous, first time to the Boston Marathon; what is the training regimen; are you professional/sponsored. It is this analysis' contention that knowing these factors could improve the predictive quality of the models presented. Also, this analysis arrived classification based on convenient categories already provided by the dataset and by the author's domain knowledge of the event type. Early experiments with clustering did not prove fruitful in terms of designating previously undiscovered classes. Further, exploration may find useful insights into the complexity of the population.