# Western States DNF Model - Appendix

## I. Code for Setup and Data Wrangling

Below is the commented code used to perform the data import and wrangling.

```
# Relevant Libraries
library(dplyr)
library(data.table)
library(tidyr)
library(stringr)
library(ggplot2)
library(ggthemes)
library(corrplot)
library(caTools)
library(ROCR)
library(caret)
library(kernlab)
library(rpart)
library(rattle)
library(randomForest)
library(nnet)
library(Boruta)

# Setting up the workspace
setwd("E:/Users/Mas/OneDrive/School/Springboard/FoundationsDS/CapstoneProject/Final Files - WSER/")
wser <- read.csv("data/wserFull.csv", stringsAsFactors = FALSE, na.strings = c(""))
```

```
# All the wrangling
## Specify columns for aidstations
keycol <- 'AS_time_name'
valuecol <- 'AS_time'

## Use the underscore (_) to easily select headers
gathercols <- colnames(wser)[grep('_', colnames(wser))]

### A regex expression is used to group the titles in the way we want

wserLong <- wser %>%
  gather_(keycol, valuecol, gathercols) %>%
  extract_(keycol, c("AS_indicator", "AS_distance", "AS_InOut"),
           "(^.)([0-9.]*)\\_([:alpha:]*)") %>%
  spread(AS_indicator, AS_time)

# Rename Column Names for p --> AS_Place, t --> AS_Time
colnames(wserLong)[which(names(wserLong) == "p")] <- "AS_Place"
colnames(wserLong)[which(names(wserLong) == "t")] <- "AS_Time"

# Reformat a few of the columns, remove empty aid station values
wserLong$AS_distance <- as.double(wserLong$AS_distance)
wserLong$AS_Place <- as.integer(wserLong$AS_Place)
wserLong$Year <- as.factor(wserLong$Year)
wserLong$Place <- as.integer(wserLong$Place)
```

```r
wserLong$First.Name <- str_trim(wserLong$First.Name)
wserLong <- wserLong %>% filter(!is.na(AS_Time)) %>% arrange(Year, Place, Last.Name)

# Convert Aid Station Splits from Time to a Number
wserLong$AS_Time_NUM <- (as.numeric(as.POSIXct(paste("2000-01-01",
                                                     wserLong$AS_Time),
                                               format = "%Y-%m-%d %H:%M")) -
  as.numeric(as.POSIXct("2000-01-01 0:0:0")))/60

## Change pre-2010 split times to "Time Elapsed"
wserLong$YearInt <- as.numeric(levels(wserLong$Year))[wserLong$Year]

wserLong <- wserLong %>% mutate(AS_Time_NUM =
                                  ifelse(AS_distance >= 55 &
                                           AS_Time_NUM <= 690 &
                                           YearInt < 2010,
                                         AS_Time_NUM + 1440,
                                         AS_Time_NUM))

# Times over 24 hours (recorded as time elapsed) return NA - fix
wserLong <- wserLong %>%
  mutate(AS_Time_NUM = ifelse(!is.na(AS_Time_NUM),
                              AS_Time_NUM,
                              sapply(strsplit(AS_Time, ":"),
                                     function(x) {
                                       x <- as.numeric(x)
                                       x[1]*60 + x[2] + x[3]/60
                                     })))

# Convert Time Elapsed field into number
wserLong <- wserLong %>%
  mutate(timeElapsed_MIN =
           ifelse(is.na(timeElapsed),
                  NA,
                  sapply(strsplit(timeElapsed, ":"),
                         function(x) {
                           x <- as.numeric(x)
                           x[1]*60 + x[2] + x[3]/60
                         })))

## Change pre-2010 split times to "Time Elapsed"
wserLong$YearInt <- as.numeric(levels(wserLong$Year))[wserLong$Year]
wserLong <- wserLong %>%
  mutate(AS_Time_NUM = ifelse(YearInt < 2010,
                              AS_Time_NUM - 300,
                              AS_Time_NUM))

# Setup for Age Group bins
ageBin <- c(0,20,30,40,50,60,70,200)

# Transform for Age Groups

# Initialize Grade Rank Column
```

```r
wserLong$ageGroup <- 0

# Step through list of grades and categorize
for (i in 1:(length(ageBin) - 1)) {
  wserLong <- wserLong %>%
    mutate(ageGroup =
             ifelse((ageGroup == 0 & Age >= ageBin[i] & Age < ageBin[i + 1]),
                    i,
                    ageGroup))
}

# Transform for DNF
wserLong <- wserLong %>%
  mutate(DNF = ifelse(is.na(timeElapsed), 1, 0))

# DNF as factor and a label
wserLong$DNF <- factor(wserLong$DNF, levels = c("1", "0"), labels = c("DNF", "Finished"))

## Only interested in "Time In"
wserLong <- wserLong %>% filter(AS_InOut != "out") %>% select(-AS_InOut)

## Re-Rank All AS Splits
wserLong <- wserLong %>% group_by(Year, AS_distance) %>%
  mutate(AS_Place = rank(AS_Time_NUM,
                         na.last = "keep",
                         ties.method = "random")) %>%
  ungroup()



# Create Long DF containing decile rank of for pace grouped by:
#   1.  Years
#   2.  Aid Stations (distance)
#   3.  Overall

wser.long <- wserLong %>%
  mutate(qRank.overall = ntile(timeElapsed_MIN, 10),
         pace.overall = timeElapsed_MIN/100) %>%
  group_by(distFact = factor(AS_distance)) %>%
  mutate(qRank.as = ntile(AS_Time_NUM, 10),
         pace.as = AS_Time_NUM/AS_distance) %>%
  ungroup() %>%
  group_by(Year) %>%
  mutate(qRank.year = ntile(timeElapsed_MIN, 10)) %>%
  ungroup()

## Add Index for which Aid Station number
iAS <- unique(wser.long$AS_distance)
iAS <- sort(iAS)
re.ind <- function(x) {
  which(iAS == x, arr.ind = TRUE)
}
```

```r
## Quickly apply index number to different aid stations for all years
wser.long$AS.Num <- sapply(wser.long$AS_distance, re.ind)

## Add in key for an individual runner per year
wser.long <- wser.long %>%
  mutate(key = paste(Year, Place, sep = "-"))

## Determine distance, time, and pace between current and previous aid station
### First we'll distinguish between the aid station number overall (AS.Num), and aid station they hit f

wser.long$AS.count <- as.integer(ave(wser.long$key, wser.long$key, FUN = seq_along))

### Set up the differential times and distances between each aid station
### Use of IFELSE to set the 1st aid station reached as the differential as well since the differential

### Use of data.table::shift function to return previous value in a row's value to get differential
wser.long <- wser.long %>% group_by(key) %>%
  mutate(diff.dist = ifelse(AS.count == 1,
                            AS_distance,
                            AS_distance - shift(AS_distance, 1L, type = "lag")),
         diff.time = ifelse(AS.count == 1,
                            AS_Time_NUM,
                            AS_Time_NUM - shift(AS_Time_NUM, 1L, type = "lag")),
         diff.pace = diff.time/diff.dist) %>%
  ungroup()

## Format DNF as a factor
wser.long$DNF <- factor(wser.long$DNF,
                        levels = rev(levels(wser.long$DNF)))

rm(wserLong)
rm(wser)
```

```r
############################################################
# Transform Long Data set (wser.long) into a wide format

## Set up key column and identifiers
wserWide <- wser.long %>%
  mutate(key = paste(Year, Place, sep = "-"),
         AS_T = paste(AS_distance, "T", sep = "_")) %>%
  select(-AS_distance, -AS_Time)

## Reorganize DF
wserTemp <- wserWide[c(25,14,1:7,16,15,17:19,23)] %>% distinct(key, .keep_all = TRUE)

## Create 2 temp DF for converting to wide
wideTime <- wserWide %>% select(key, AS_Time_NUM, AS_T)

## Spread varying data
wideTime <- wideTime %>% spread(AS_T, AS_Time_NUM)

## Merge back into single dataframe
wserT <- merge(wserTemp, wideTime, by = "key")
```

```
wserTemp <- wser.long %>%
  group_by(key) %>%
  mutate(pace.avg = mean(diff.pace),
         pace.sd = sd(diff.pace),
         as.max = as.numeric(ifelse(DNF == "Finished",
                           max(AS.count) + 1,
                           max(AS.count)))) %>%
  ungroup() %>%
  select(key, pace.avg, pace.sd, as.max) %>%
  distinct(key, .keep_all = TRUE)

## Get all the info into one DataFrame
wserT <- merge(wserT, wserTemp, by = "key")

## Standardize the name
wser.wide <- wserT

wser.wide <- wser.wide %>% mutate(bibNo = as.integer(gsub("[a-zA-Z]","", Bib)))
wser.wide <- wser.wide %>% mutate(bibNo = ifelse(is.na(bibNo), 150, bibNo))

## Tidy up our Workspace
rm(wserT)
rm(wideTime)
rm(wserTemp)
rm(wserWide)

wser.long <- wser.long %>%
  mutate(sex.male = ifelse(Sex == "M",1,0),
         DNF = ifelse(DNF == "Finished", 0, 1))

wser.wide <- wser.wide %>%
  mutate(sex.male = ifelse(Sex == "M",1,0),
         DNF = ifelse(DNF == "Finished", 0, 1))
```

---

## II. Table for Long Data Frame

The long data frame is split so that each row not only represents a specific runner for each year, but also represents a given aid station.

| Variable Name | type | Explanation |
|---|---|---|
| `key` | chr | Unique ID of year and overall place (e.g., 1995-3) |
| `Year` | factor | Year of race (1986 - 2016) |
| `YearInt` | num | Conversion of `Year` to a number |
| `Place` | int | Finishing place for that year |
| `Last.Name` | chr | Last name of racer |
| `First.Name` | chr | First name of racer |
| `Bib` | chr | Bib number of racer (contains alphanumeric) |
| `Sex` | chr | Male or Female ("M" or "F") |
| `sex.male` | num | Binary version of `Sex` ("0" = Female, "1" = Male) |
| `Age` | int | Age of runner for given year |
| `ageGroup` | num | Index number for 7 bins - (0,20,30,40,50,60,70,200) |
| `City` | chr | Listed city of origin |
| `State` | chr | Listed state of origin |
| `timeFinish` | chr | Time of day runner finished (NA for DNF) |
| `timeElapsed` | chr | Overall time elapsed for runner (NA for DNF) |
| `timeElapsed_MIN` | num | Conversion of `timeElapsed` to number |
| `qRank.overall` | int | Decile ranking of runner's finish time to overall times |
| `pace.overall` | num | Average pace held for entire race |
| `DNF` | num | Finish status - "0" = Finished, "1" = DNF |
| `AS_distance` | num | Distance of aid station from start |
| `distFact` | factor | Conversion of `AS_distance` as factor |
| `AS_Place` | int | Place ranking of runner into given aid station |
| `AS_Time` | chr | Time of runner's arrival into given aid station |
| `AS_Time_NUM` | num | Conversion of `AS_Time` to elapsed minutes from start |
| `qRank.as` | int | Decile ranking of runner's place into an aid station |
| `pace.as` | num | Pace runner averaged to the distance of given aid station |
| `qRank.year` | int | Decile ranking of runner's finish time to current year |
| `AS.Num` | int | Index number of aid station for all aid stations (1-22) |
| `AS.count` | int | Index number of given aid station within given year |
| `diff.dist` | num | Distance between current aid station and previous |
| `diff.time` | num | Time to reach current aid station from previous |
| `diff.pace` | num | Average pace to reach current aid station from previous |

## III. Table for Wide Data Frame

The wide data frame is split so that each row represents a specific runner for each year

| Variable Name | type | Explanation |
|---|---|---|
| `key` | chr | Unique ID of year and overall place (e.g. 1995-3) |
| `YearInt` | num | Year of race (1986-2016) |
| `Year` | factor | Year of race as a factor |
| `Place` | int | Overall place |
| `Last.Name` | chr | Last name of runner |
| `First.Name` | chr | First name of runner |
| `Bib` | chr | Bib number of runner (mostly numbers, some alpha numeric) |
| `Sex` | chr | Male or Female (M or F) |
| `Age` | int | Age of runner for given year |
| `ageGroup` | num | Index number for 7 bins - (0,20,30,40,50,60,70,200) |
| `timeElapsed_MIN` | num | Overall time for race (NA if DNF) |
| `DNF` | num | Finish status - "Finished" = 0, "DNF" = 1 |
| `qRank.overall` | int | Decile rank based on overall time vs all years |
| `pace.overall` | num | Overall pace for duration of race (assumes distance of 100 miles) |
| `qRank.year` | int | Decile rank based on overall time vs given year |
| ** AID STATIONS ** | num | Time elapsed (in minutes) to reach given aid station |
| `pace.avg` | num | Average pace between each aid station |
| `pace.sd` | num | Standard deviation of the paces between each aid station |
| `as.max` | num | Maximum number of aid stations given runner checked in to |
| `bibNo` | num | Reformatted `Bib` removing the ALPHA characters |
| `sex.male` | num | Reformatted `Sex` - "Female" = 0, "Male" = 1 |

## IV. Table of all Aid Stations

| Checkpoint | Distance |
|---|---|
| Talbot Creek (snow) | 15.2 |
| Poppy (snow) | 20.3 |
| Lyon Ridge | 10 |
| Red Star | 16 |
| Duncan Canyon | 23.8 |
| Robinson Flat | 29.7 |
| Millers Defeat | 34.4 |
| Dusty Corners | 38 |
| Last Chance | 43.3 |
| Devils Thumb | 47.8 |
| El Dorado Creek | 52.9 |
| Michigan Bluff | 55.7 |
| Foresthill | 62 |
| Cal-1 (Dardanelle) | 65.7 |
| Cal-2 (Peaches) | 70.7 |
| Rucky Chucky/River | 78 |
| Green Gate | 79.8 |
| Auburn Lake Trails | 85.2 |
| Browns Bar | 89.9 |
| HWY 49 | 93.5 |
| No Hands Bridge | 96.8 |
| Robie Point | 98.9 |
| Placer High School | 100.2 |