SMDE Second Assignment

Wilmer Uruchi Ticona

January 9, 2019

Abstract

The following is a simulation project of the performance of runners of the Boston Marathon of 2017. This project is based on material from the course "Statistical Modelling and Design of Experiments", and the book "Simulation: The Practice of Model Development and Use" by Stewart Robinson. The project starts with a brief description of the system to model, including the specification of its components, and processes involved. Then we follow up with a definition of the experimental framework that uses a factorial design, and the validation of the model. We have also included the source code of the program we wrote in python to facilitate the experimentation process.

Contents

1 Executive Summary **3**

2 System Description, Introduction **3**

3 Problem Description **4**

3.1 Structural and Simplifying Hypothesis 4

4 Model Specification **5**

4.1 Modeling Objectives 5

4.2 Model Inputs and Outputs 5

4.2.1 Experimental Factors 5

4.2.2 Responses 5

4.3 Model Content 6

4.3.1 Model Scope 6

4.3.2 Model Level of Detail 7

4.3.3 Assumptions 7

4.3.4 Simplifications 7

4.4 Model Data 8

5 Codification **8**

5.1 Simulation Program 8

5.2 Data 12

6 Definition of the Experimental Framework **14**

6.1 Arrival Rate 14

6.1.1 Sample Size 14

6.2 Obtaining accurate results 15

6.2.1 The nature of the simulation model 15

6.2.2 The nature of the simulation model output 15

6.2.3 Dealing with initialization bias 15

6.2.4 Amount of output data required 16

6.3 Searching the solution space 16

6.3.1 Specific Objective 16

6.3.2 Analyzing the results 18

7 Model Validation **18**

8 Results/Conclusions **19**

1 Executive Summary

Every year, several people from different countries and with different backgrounds participate in the Boston Marathon. Although some factors are generally accepted as being determinant enough to rule the performance of a runner, it is hard to try to identify the influence that other factors have because the competition takes place only once a year.

Finishers Boston Marathon 2017 has the names, times, and general demographics of the finishers. Using this information we will develop a simulation model reliable enough to allow for experimentation that can let us measure the impact that the factors, based on the variables included in the dataset, have in the result of the simulation, i.e. the total time for the marathon. We will use *batch experiments* running different combinations of factors to obtain statistically significant results.

It is worth nothing that the data presents some limitations regarding the competitors physical condition, e.g. weight, height, blood pressure, fat level, etc.

2 System Description, Introduction

The Boston Marathon can be seen as a queue system at which runners arrive and after a variable amount of time leave the system having spend some *transient* amount of time in it. This is a terminating system, that starts with a certain amount of participants and finishes when all these participants have finished going through the system. It is possible that some participants drop out of the system, but that case is not covered in this simulation.

Every year, more than $30,000$ participants sign up for the Boston Marathon. They come from different backgrounds, and have prepared for the event in different ways. There are professionals who come with the objective of winning the event, but also people who are running their first marathon and come with the goal of at least finishing it. This set up presents a variety of situations that can be analyzed.

Since the data does not offer much information regarding the physical condition of the participants, it has been decided to start the system after the first 5 km. That means that the time done by a participant in the first 5 km is going to be included as an initial condition in the simulation for the rest of the marathon. Thus, avoiding *initialization bias* to some extent.

When a participant arrives to the goal, the total time is recorded, that is, the total time it took for the participant to complete the marathon (including the time for the first 5 km).

There are some other factors that represent unpredictable variability, one of them is the weather, other could be the effect of the surrounding competitors in the performance of a runner.

3 Problem Description

The Boston Marathon is an annual marathon hosted by several cities in Massachusetts, United States. It is one of the six World Marathon Majors. The event attracts $500,000$ spectators, and an average of $30,000$ participants each year, through the hilly terrain of Massachusetts and varying weather conditions.

The race uses a staggered “wave start”, where top groups start earlier to help reduce congestion. For the 2017 marathon the start was divided into 8 waves. Certainly, the group of elites will have a better performance than the rest of the participants, so it might be convenient to exclude them in favor of finding a homogeneous group to experiment with.

The course runs through 42.195 km of winding roads through 8 Massachusetts cities and tows, to finish a Copley Square. The weather and terrain conditions are varied through the course, and could influence in the performance of a competitor. However, there is not enough data to predict the variability of these factors.

The main problem is the lack of a simulation model that reflects the results of the system accurately enough to allow for experimentation of the factor that influence in the performance of the variety of participants. Where the result of the system is the total time it takes a participant to complete the course; and the factors, variables such as age, ranking, gender, initial time.

Through experimentation with the simulation model we want to identify how the factors presented in the marathon data of 2017, influence in the performance of the competitors.

3.1 Structural and Simplifying Hypothesis

- **SH_01** Unpredictable variables that affect the model will be represented by random noise in the simulation process.
  Identifier: Random Noise

- **SH_02** The model will start after the first 5 km to avoid initialization bias, so the time for the first 5 km will serve as the arrival time.
  Identifier: Arrival time

- **SH_03** The participants classified as Elite will not be considered for the formulation of the model.
  Identifier: Data Sample

- **EH_01** The arrival time is represented by an empirical distribution from which random samples will be taken for the simulation process.
  Identifier: Arrival Time Empirical Distribution

- **EH_02** The result, which is the time a participant spends in the system, will be represented by a linear model.
  Identifier: Linear Model

- **EH_03** This a terminating simulation with a transient output.
  Identifier: Nature of the Simulation Model

4 Model Specification

4.1 Modeling Objectives

The **general modeling objective** is to:

- Design a simulation model accurate enough to perform experimentation on it. [1]

The **specific objective** is to:

- Through experimentation with the model, find the influence of the factors in the performance of the participants.

4.2 Model Inputs and Outputs

4.2.1 Experimental Factors

- Bib: Assigned race number based on qualifying time (range $3 - 31437$)
- Gender: Runner's gender (range $1 - 2$, where 1 is Male, and 2 is Female)
- Age: Age in race day (range $20 - 60$)
- Time 5k: Time in seconds for the first 5 km (range $924 - 3254$)

4.2.2 Responses

- Result: Total time for the marathon in seconds for each participant

[1] The development of this objective is explained in the Validation section.

4.3 Model Content

4.3.1 Model Scope

Table 1: Model Scope

Component	Include/exclude	Justification
Participant	Include	The main actor of the model. **Experimental Factor.**
Running Marathon	Include	Key process of the simulation model.
Physical Condition	Exclude	Not enough data.
Weather	Exclude	Not enough data.
Terrain variation	Exclude	Not enough data.

The main opportunity for scope reduction comes from the exclusion of weather conditions, terrain variation, and the physical condition of the participants; since there is not enough data to take these factors into consideration.

4.3.2 Model Level of Detail

Table 2: Model Level of Detail

Component	Detail	Include/exclude	Comment
Participant	Bib	Include	Assigned race number based on qualifying time. **Experimental Factor.**
	Gender	Include	Runner's gender. **Experimental Factor.**
	Age	Include	Age in race day. **Experimental Factor.**
	Time 5k	Include	Time in seconds for the first 5 km. **Experimental Factor.**
	Elite	Exclude	Elite participants are excluded from the model.
Running Marathon	Result	Include	Time it takes a runner to complete the marathon in seconds. **Response**
	Abandons	Excluded	Participants that abandon the race before finishing.
	Start	Included	The simulation model of the marathon starts after the first 5 km.

4.3.3 Assumptions

- The weather conditions are not extreme enough, and the terrain variation is negligible.
- Most of the participants finish the marathon.
- The marathon is a terminating process.
- Interruptions of the marathon are infrequent, therefore they are not modeled.

4.3.4 Simplifications

- Negligible variations can be represented by random noise.
- The simulation model can start after the first 5 km to avoid initialization bias.
- The results from Elite participant are not relevant enough and can be excluded.
- Only participants in the age between 20 and 60, inclusive, are considered.

4.4 Model Data

Participant

Table 3: Participant Data

Variable	Detail
Quantity	24597 participants
Bib	Max: 31437 Min: 3 Median: 14802
Gender	range$(1-2)$ 1: Male 2: Female
Age	Max: 60 Min: 20 Median: 42 Mean: 41
Time 5k (seconds)	Max: 3254 Min: 924 Median: 1491 Mean: 1523 Standard Deviation: 237.718 Inter Quartile Range: 297

5 Codification

5.1 Simulation Program

This is the python (3.6) code that runs the simulation. The program imports its data from a file called *EmpiricalData.txt*, the process to generate this file will be explained in the following section. The program writes the result of each experiment in a corresponding text file that can be easily imported to *RStudio* for further analysis.

```
import random
import csv
import copy
import numpy as np

class Runner(object):
    # Constructor of the runner class that represents a participant
    def __init__(self, gender, bib, age, t5k, officialTime):
        self.Gender = gender
        self.Bib = bib
        self.Age = age
        self.Time5k = t5k
        self.OfficialTime = officialTime
        self.Result = 0

    # Printing override
    def __str__(self):
        output = 'Runner Gender = ' + str(self.Gender) + ', Bib : ' + \
        str(self.Bib) + ', Age : ' + str(self.Age) + ', Time5k : ' + \
        str(self.Time5k) + ', Official Time: ' + str(self.OfficialTime) + \
        ", Result: " + str(self.Result) + "\n"
        return output

    # Calculation using the linear model
    # Includes random noise from a normal distribution
    def calcResultL1(self, cAge, c5k, cBibN, cGender):
        self.Result = float(self.Age) * cAge + \
        float(self.Gender) * cGender + \
        float(self.Bib) * cBibN + float(self.Time5k) * c5k + \
        np.random.normal(0,0.5) * 100
        self.Result = int(self.Result)


class EmpiricalDistribution(object):
    # Constructor of the empirical distribution
    def __init__(self):
        self.Name = 'Empirical Distribution'
        self.Runners = []
        self.Selection = []

    # Method that reads the data from a txt file previously
    # build using RStudio that contains the empirical distribution
    # of the total of participants
    def readData(self):
        with open('EmpiricalData.txt') as csv_file:
            csv_reader = csv.reader(csv_file, delimiter=',')
            line_count = 0
            for row in csv_reader:
                if line_count == 0:
                    print(f'Column names are {", ".join(row)}')
                    line_count += 1
                else:
                    self.Runners.append(Runner(row[2], row[1],\
                                               row[3], row[4], row[5]))
            print(f'Processed {line_count} lines.')

    # Selects a sample of size n with replacement from
    # the empirical distribution
    def selectRandom(self, n):
        self.Selection = []
        print(str(n) + ' runners picked at random')
        for i in range(n):
          runner = random.choice(self.Runners)
          self.Selection.append(runner)
          print(str(runner))

        return self.Selection


class Simulation(object):
    # Constructor, contains the coefficient of the linear model
    # Also contains the min and max values for the 2^k Factorial Exp.
    def __init__(self, distribution):
      self.Empirical = distribution

      self.L1Age = 5.404e+00
      self.L15k = 9.006e+00
      self.L1BibN = 8.650e-03
      self.L1Gender = -1.865e+02

      self.AgePos = 60
      self.AgeNeg = 20
      self.GenderPos = 1
      self.GenderNeg = 2
      self.BibPos = 31437
      self.BibNeg = 3

    # Method that executes the simulation for all the runners
# Calls the calcResultL1 methods and sends the coefficientes
# The Result property of each runner is filled with the result
def runSimulation(self,currentDistro,file):

  for runner in currentDistro:
      runner.calcResultL1(self.L1Age, self.L15k, \
                          self.L1BibN, self.L1Gender)
      print(str(runner))

  output = '"Num","Bib","Gender","Age","Time5k","OfficialTime","Result"\n'
  i = 0
  for run in currentDistro:
      i += 1
      output += str(i) + ',' + str(run.Bib) + ',' + \
      str(run.Gender) + ',' + \
      str(run.Age) + ',' + str(run.Time5k) + ',' + \
      str(run.OfficialTime) + \
      ',' + str(run.Result) + '\n'

  print(output,  file=open(file, 'w'))

# Runs a regular simulation with a random sample
# Does not modify factors
def runRegularSim(self):
    currentDistro = copy.deepcopy(self.Empirical)
    self.runSimulation(currentDistro,'regularSim.txt')

# Runs a simulation according to some factor setup
# Receives a + or - sign for each factor
# Uses the values from the constructor accordingly
# Stores the results in a file that can be import to RStudio
def runFactorialSim(self, age, gender, bib, identity):
    currentDistro = copy.deepcopy(self.Empirical)
    for runner in currentDistro:
        runner.Bib = self.BibPos if bib == '+' else self.BibNeg
        runner.Gender = self.GenderPos if \
        gender == '+' else self.GenderNeg
        runner.Age = self.AgePos if age == '+' else self.AgeNeg
    self.runSimulation(currentDistro, str(identity) + 'Sim_Bib' + \
                       str(bib) + 'Gender' + str(gender) + 'Age' + \
                       str(age) + 'factors.txt')
    def printRunners(self):
        for runner in self.Empirical:
            print(runner)

empirical = EmpiricalDistribution()
empirical.readData()
# Taking a random sample of 2000 participants
mySim1 = Simulation(empirical.selectRandom(2000))

mySim1.runRegularSim()

# Running the total of Factorial Experiments
mySim1.runFactorialSim('-','-','-','R1')
mySim1.runFactorialSim('+','-','-','R2')
mySim1.runFactorialSim('-','+','-','R3')
mySim1.runFactorialSim('+','+','-','R4')
mySim1.runFactorialSim('-','-','+','R5')
mySim1.runFactorialSim('+','-','+','R6')
mySim1.runFactorialSim('-','+','+','R7')
mySim1.runFactorialSim('+','+','+','R8')
```

5.2 Data

The data for the Boston Marathon of 2017 has been selected as the basis of this experiment, available in the https://www.kaggle.com/rojour/boston-results website. Data transformation and analysis has been performed in the data to make it usable for the current experiment. The details for this process can be found in the attached file **DataPreparation.html**, also, the data is included in the file **Boston2017.RData**. The whole process has been performed using RStudio.

Linear regression assumptions, such as: Linear relationship, multivariate normality, no or little multicollinearity, no auto-correlation, and homoscedasticity; have been tested in a previous exercise, consequently, are not considered in the present modeling process.

Here is a full description of all the variables found in the data.

Table 4: Data description

Variable	Detail	Description
Bib	Experimental Factor	Assigned race number based on qualifying time. “F” could appear for female elites.
Name		Name of runner
Age	Experimental Factor	Age on race day
M/F	Experimental Factor	Runner’s gender
City		Runner’s city of residence
State		Runner’s state of residence
Country		Runner’s country of residence
Citizen		Runner’s nationality (optional)
5k	Experimental Factor	Runner’s time at 5k
10k		Runner’s time at 10k
15k		Runner’s time at 15k
20k		Runner’s time at 20k
Half		Runner’s time at halfway point
25k		Runner’s time at 25k
30k		Runner’s time at 30k
35k		Runner’s time at 35k
40k		Runner’s time at 40k
Pace		Runner’s overall minute per mile pace
OfficialTime	Result	Runner’s official finishing time
Overall		Runner’s overall ranking
Gender		Runner’s ranking in their gender (Not to be confused with the experimental factor used in the experimentation)
Division		Runner’s ranking in their age division

This is the description of the linear model of the relationship of the experimental factors as explanatory variables for the result OfficalTimeSeconds (OfficialTime in seconds).

```
Call:
lm(formula = OfficialTimeSeconds ~ Age + X5kseconds + BibN +
    GenderN, data = Boston2017NoPro)

Residuals:
     Min        1Q    Median        3Q       Max
-12334.5    -770.0    -212.6     551.1   12764.8

Coefficients:
```

```
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  3.580e+02  8.397e+01   4.264 2.02e-05 ***
Age          5.404e+00  7.820e-01   6.911 4.92e-12 ***
X5kseconds   9.006e+00  6.157e-02 146.268  < 2e-16 ***
BibN         8.650e-03  1.652e-03   5.236 1.66e-07 ***
GenderN     -1.865e+02  1.721e+01 -10.834  < 2e-16 ***
---


Residual standard error: 1183 on 24592 degrees of freedom
Multiple R-squared:  0.7736,    Adjusted R-squared:  0.7735
F-statistic: 2.1e+04 on 4 and 24592 DF,  p-value: < 2.2e-16
```

In this linear model, X5kseconds represents the variable “5k” expressed in seconds, BibN represents “Bib”, and GenderN represents “M/F” as a numerical value.

The coefficients found in this linear model will be used to calculate the behavior of the model in the experimentation.

The variable “X5kseconds”, which is the time for the first 5 km in seconds, will be used as the distribution of the arrival time, in the form of an **empirical distribution**.

6 Definition of the Experimental Framework

6.1 Arrival Rate

In this special case we have the data for all the participants in the marathon, and this does not usually happen. Starting from that point, we tried to model the **arrival rate** of runners to the model using a statistical distribution, defining our arrival rate as the time for the first **5 km**; however, those distributions didn’t really fit into the distribution of the historical data. So, our approach is to use the historical data as an **empirical distribution** from which we can obtain random samples with replacement to test the model, the technique applied in this case is **bootstrapping**.

Moreover, by using bootstrapping we are streamlining the analysis of the simulation and the model validation, which are main objectives in this project.

6.1.1 Sample Size

Considering a **confidence level** of 95%, a **margin of error** of 5%, and a **population** of 24,597 runners, after cleaning the data. We calculated a **sample size** of 379 observations with replacement.

6.2 Obtaining accurate results

6.2.1 The nature of the simulation model

The Boston Marathon simulation is a terminating simulation. The termination point is after the last runner has reached the finish line. This might take no more than 9 hours.

6.2.2 The nature of the simulation model output

There is one key output value:

- Result: Total time in seconds for a runner to finish the marathon.

The output is expected to be transient since different runners have different performances. A histogram of the distribution of total time per runner (Figure 1) shows that this output varies highly, and that there is a min and a max values, indicating that the output is transient.

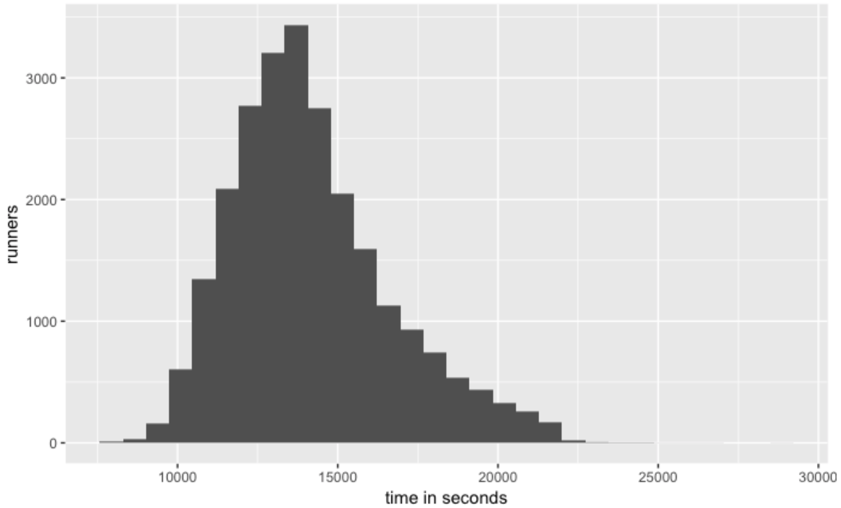

Figure 1: Plot of results

6.2.3 Dealing with initialization bias

To avoid initialization bias, the model will start after the first 5 km have been completed, using the time for that track as the arrival time.

6.2.4 Amount of output data required

Given that this is a terminating simulation, and that every runner represents a simulation by itself, we can define each runner as a simulation, so we are going to have the same number of replications as the size of our random sample, that is 379.

6.3 Searching the solution space

6.3.1 Specific Objective

To search the solution space according to our specific objective we are going to adopt a 2^k factorial design, specifically, a 2^{3-1} factorial design where our three factors will be **Bib**, **Gender**, and **Age**; while the factor not included will be **Time 5k**, as it will represent the arrival time.

For this purpose we define the following table of scenarios:

Table 5: Factorial Design

Scenerio	Bib	Gender	Age	Time 5k	Response
1	-	-	-	Arrival Time	R_1
2	+	-	-	Arrival Time	R_2
3	-	+	-	Arrival Time	R_3
4	+	+	-	Arrival Time	R_4
5	-	-	+	Arrival Time	R_5
6	+	-	+	Arrival Time	R_6
7	-	+	+	Arrival Time	R_7
8	+	+	+	Arrival Time	R_8

Table 6: Factor Values

Factor	+ Value	- Value
Bib	3	31437
Gender	1	2
Age	20	60

Then, the 379 simulations are performed for each scenario, and the Response is recorded. In this case the Response for each scenario is the mean Response for all the simulations of the scenario. We obtain the results:

Table 7: Scenarios Results

Scenario	Result	Value
1	R_1	13760.87
2	R_2	13543.17
3	R_3	13949.38
4	R_4	13729.60
5	R_5	13486.13
6	R_6	13270.29
7	R_7	13670.58
8	R_8	13455.01

For these experiments the value **Time 5k** is the arrival time sampled at random with replacement from the empirical distribution represented by the data provided from the Boston Marathon of 2017.

Based on these scenarios we can now compute the effect of changing each factor.

The main effect of changing **Bib (Factor 1)** can be computed using the following formula:

$$e_1 = \frac{(R_2 - R_1) + (R_4 - R_3) + (R_6 - R_5) + (R_8 - R_7)}{4}$$

$$e_1 = \frac{(13543.17 - 13760.87) + (13729.60 - 13949.38) + (13270.29 - 13486.13) + (13455.01 - 13670.58)}{4}$$

$$e_1 = -217.2243$$

Similarly, we calculate the main effect of **Gender (Factor 2)** e_2:

$$e_2 = \frac{(-R_1 - R_2 + R_3 + R_4 - R_5 - R_6 + R_7 + R_8)}{4}$$

$$e_2 = 186.029$$

And **Age (Factor 3)** e_3:

$$e_3 = \frac{(-R_1 - R_2 - R_3 - R_4 + R_5 + R_6 + R_7 + R_8)}{4}$$

$$e_3 = -275.252$$

6.3.2 Analyzing the results

The results from our factorial experiments tell us that:

- **Bib (Factor 1)** has a negative main effect factor that indicates that, on average, changing the factor from its - to its + level decreases the response value of the main effect. So, we can interpret that as closer the values are to its + level (3), we can expect lower average times for the marathon.
- **Gender (Factor 2)** has a positive main effect factor that indicates that, on average, changing the factor to its - to its + level increases the response value of the main effect. In our context that means that in average women tend to perform better than men in this competition.
- **Age (Factor 3)** has a negative main effect factor that indicates that, on average, changing the factor from its - to its + level decreases the response value of the main effect. We can interpret that as closer the values are to its + level (20), we can expect lower average times for the marathon.

From these factors, moving Bib and Age to their positive values has the effect of decreasing the average result by -217.2243 and -275.252, correspondingly. So it is possible to conclude that **Age (Factor 3)** is the factor that has a stronger influence on the result of the simulation.

7 Model Validation

In this section we are going to perform a black-box validation test in the form of a comparison with the real system, given that we have complete data from the real system. We start by defining our hypothesis:

H_0: The output of the simulation and the real model are different.

H_1: Under the same conditions, the output of the simulation and the real system should be sufficiently similar.

We can define H_1 as:

H_1 : If $I_S = I_R$ then $O_S \sim O_R$

Where:

I_R = Inputs to real system.
O_R = Outputs from real system.
I_S = Inputs to simulation model.

O_S = Outputs from simulation model.

We want not only to check the average levels of the results, but also to compare their spread. We are going to do this by calculating a confidence interval for the difference in the means. We do this by using the following formula:

$$\bar{X}_S - \bar{X}_R \pm t_{2n-2,\alpha/2}\sqrt{\frac{S_S^2 + S_R^2}{n}}$$

where:

$\bar{X}_S$ = mean of simulated output data
$\bar{X}_R$ = mean of real system output data
S_S = standard deviation of simulated output data
S_R = standard deviation of real system output data
n = number of observations
$t_{2n-2,\alpha/2}$ = value from Student's t-distribution with $2n-2$ degrees of freedom and significance level of $\alpha/2$.

and replacing we have:

$$13609.11 - 14032.98 \pm 1.96\sqrt{\frac{2092.04^2 + 2459.63^2}{379}}$$

$$-423.87 \pm 1.28 \cdot 165.86 = -423.87 \pm 212.30$$

With 95% confidence the difference in mean output data between the simulation and the real system is between -211.57 and -635.44 seconds. Our best estimate of the difference, the point estimate, is -423.87 seconds. The standard error of the difference is 165.86, and the margin of error is 212.30 units. These results are not bad, considering that our model only applied linear regression.

Given this information we reject the null hypothesis.

8 Results/Conclusions

Defining the objectives of the simulation is necessary before starting to work in the design of experiments. Although we did not have a clear objective at the beginning, we could come up with something relevant after analyzing the data, and how the real system works. Otherwise, we would not have been able to define experiments.

The model predicts the behavior of the system accurately enough to allow experimentation with the explanatory factors. From these factors, we have detected that the one that influences the most in the final result of the simulation process is **Age (Factor 3)**, as the model results project that a lower age tends to indicate a lower result. Remember that a lower results indicates a better performance. On the other hand, we have found that the factor **Gender (Factor 2)** indicates that, in average, females perform better at the marathon than males. Finally, as expected, **Bib (Factor 1)** has a great effect in the result of the simulation, better ranked runners perform better in average.

We decided to simplify the model by making the time for the first 5 km serve as the arrival time. By doing it we also simplified the sampling process for our experiments, and it seems that it also allowed for better accuracy and ease of validation of our model.

In the case of this experiment is hard to objectively state that the model results match accurately the results of the real system, for this reason we opted for a validation that gave us an estimation of how close we are to the real system. Given the restrictions of the data we used in the model, we can optimistically conclude that our model results match the result with certain confidence that make our factorial experimentation significative.

Finally, we can conclude that our model, and the experiments performed, allowed better understanding of the factors that influence the results of runners in the Boston Marathon.