Reducing Participants' Interference at the World's Best 10K Race

A Thesis submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

in:

INDUSTRIAL ENGINEERING

(Management Systems)

Presented by:

Rosemarie Santa González

Sonia M. Bartolomei Suárez, Ph.D.
Committee Member

Date

Mercedes Ferrer Alameda, M.S.
Committee Member

Date

Héctor J. Carlo Colón, Ph.D.
President of Graduate Committee

Date

Viviana I. Cesaní Vázquez, Ph.D.
Department Director

Date

José M. Atiles Osoria, Ph.D.
Graduate School Representative

Date

ABSTRACT

The World's Best 10K Race (WB10K) is a yearly international event that takes place in Puerto Rico with approximately 10,000 participants. This study seeks to reduce interference between participants during the race by improving the corrals design and assignment of participants to corrals. Interference is quantified as the number of times a participant passed another participant at predetermined checkpoints. A theoretical lower bound for the number of interferences is presented. Regression analyses are used to estimate the participants' finish time. The finish time estimates are used to assign bib numbers, and consequently corrals, to participants. In addition, Monte Carlo simulations are combined with the regression-based assignments to find the corral design that minimizes the expected number of interferences. Six corral policies are proposed and evaluated. It was found that the most effective policy to reduce interferences is an implementation of waves (*i.e.*, sequential corral release).

RESUMEN

El World’s Best 10K Race (WB10K) es una carrera internacional en Puerto Rico con aproximadamente 10,000 participantes. Este estudio busca reducir la interferencia entre los participantes durante la carrera mejorando el diseño de corrales y la asignación de participantes a corrales. Interferencia se cuantifica como el número de veces que un participante pasó otro en los puntos de cotejo predeterminados. Una cota inferior (“lower bound”) teórica para el número de interferencias es presentada. Regresiones estadísticas son utilizadas para estimar el tiempo de llegada de los participantes. Las estimaciones de tiempo de carrera son utilizados para asignar los “bib numbers,” y consecuentemente corrales, a los participantes. Además, simulaciones Monte Carlo son combinadas con las regresiones para identificar las políticas de diseño de corrales que minimizan el número de interferencias. Seis políticas de corrales son propuestas y evaluadas. Se encontró que la política más efectiva para reducir interferencias es la implementación de olas (*i.e.*, salida secuencial de corrales).

To my mother and my fiancé.

Thank you for your unconditional support.

Acknowledgements

First of all, I would like to thank Professor Hector J. Carlo. For more than five years Professor Carlo has been my academic mentor. He guided me during my B.S.I.E. and inspired me to pursuit an M.S.I.E. Thanks to him I have been exposed to the process of creating knowledge. During my years working with Professor Carlo I discovered my true passion, applying I.E. techniques in non-traditional environments. He also inspired me to pursue a Ph.D. that would allow me to further contribute to my field.

Also, I would like to thank Professor Mercedes Ferrer Alameda. When I started this study as an undergraduate student Professor Ferrer always had her door open to clarify any doubts that emerged along the way. Even though I never got the opportunity to be in Professor Ferrer’s classroom, she never denied me the chance to learn from her, despite her busy schedule. In addition, as a member of my thesis committee she advised me through the statistical analyses and helped me grow as a graduate student.

None the less, I would like to thank Professor Sonia M. Bartolomei Suárez. Professor Bartolomei Suárez has influenced my academic career even before I had a chance to join her classroom. She supported my interest in simulation by being my mentor for the Arena Simulation competition. Once I got to be part of her classroom in ININ 4040 and later ININ 6026 she gave me the chance to gain practical experiences in her classroom. Her passion for Industrial Engineering and Simulation is contagious and made me realized the real value of IEs.

Last but not least, I would like to thank the World’s Best 10K organization for sharing their time, knowledge and data. In particular, I wish to thank Mr. Rafael Acosta for the opportunity to collaborate in this outstanding event, and for the confidence in this work. Also, to Ms. María Archilla for facilitating the data.

Table of Contents

List of Figures

List of Tables

Nomenclature

The following are the key terms used in this thesis:

Bib number: A bib number is a unique id by which participants are identified, also referred to as *race bib*. The bib number corresponds to a particular corral identified by a color. The larger the bib number the further from the start line the participant will start the race.

Corral: A corral is a designated starting group of a race. Typically a corral can accommodate a large number of participants who are assigned based on the estimated time to finish a race.

Checkpoint: A checkpoint is a specific distance at which the time is registered for each participants. The checkpoint location depends on the distance of the race event.

Elite Athlete: Athletes who can finish the WB10K in 30 minutes or less. This status is earned by registering a finish time in an official international 10 km races that can prove they can comply with the time requirement. In the WB10K race these athletes are assigned to the first corral with a capacity for 100 runners.

Gun Time: The gun time is the time it takes a participant to complete the race, measured from the start of the race. It is called gun time due to the tradition of firing a gun to indicate the beginning of the race. Note that several minutes may pass from the gun start to the time a participant passes the start line.

IAAF: The International Association of Athletics Federation was founded on 1912 to fulfil the need for a world governing authority, competition programme, standardized technical equipment, and a list of official world records [2].

Interference: If participant A passes a checkpoint after participant B, but reaches the next checkpoint before participant B, it is said that A passed B during that interval, hence there was an interference between the participants. Interferences are calculated using the time registered by participants at the four checkpoints

Net Time: The net time is the time it takes the participant to travel from the start line to the finish line, also referred to as *chip time*.

Non-elite: Athletes who did not registered a qualifying elite time WB10K. These participants are assigned a higher bib number than the elite athletes.

Start line: The start line depicts the point where the race begins.

Waves: During a race, participants are assigned different starting groups to begin the same track and compete in the same events. The different starting groups are referred to as waves or *start waves*. Each wave will start the race at a different moments.

WB10K: The World's Best 10K is a world renown 10 kilometer race that takes place every year on the Puente Teodoro Moscoso in San Juan, Puerto Rico since 1998 [1].

1. Introduction

1.1 Motivation and General Purpose

The World’s Best 10K Race (WB10K) is a yearly international event that takes place on the Teodoro Moscoso Bridge at San Juan, Puerto Rico. The first race was celebrated in 1998 with 1,215 participants, a number that has increased to nearly 10,000 participants in the most recent editions. Originally called the Teodoro Moscoso Bridge Race, the WB10K changed its name in 2000 to cater to an international audience. Since then, the race has become an international sensation. In fact, in the 2003 edition of the WB10K, British athlete Paula Radcliffe broke the female world record for the 10 kilometer distance.

Since the inception of the International Association of Athletics Federation (IAAF) Road Race Label Events classification in 2008, the WB10K has been recognized with the most prestigious classification - a Gold Label [1]. In 2008 only ten races received this distinction worldwide; four of them from the Occidental hemisphere: Boston Marathon, Chicago Marathon, New York City Marathon, and the WB10K. Currently, the WB10K is one of only three 10K races world-wide with a Gold Label. Through time the WB10K has gone from a local event, to an internationally recognized race, to one of the best races in the world.

Elite athletes and local sports enthusiast from all ages participate in this event by either walking or running the 10 kilometers. In order to ensure that the race starts in an orderly manner, participants are divided into groups, also known as *corrals*. Ideally, participants that will maintain a similar pace will be assigned to the same corral, placing the fastest participants to the front corral. Corral assignments directly influence how participants experience the event. A good corral assignment would reduce interference (*i.e.*, passing) between participants during the race. Interference between participants may cause sudden change of pace, unnecessary zigzagging, frustration, and can make participants prone to accidents.

1.2 Objectives

The main goal of this thesis is to develop a methodology to reduce interference between participants during the WB10K race. In order to achieve this goal, the following objectives are defined:

Primary Objectives:

- Quantify the number of interferences between participants during previous editions of the WB10K race;
- Develop regression-based models to estimate the finish time of participants;
- Propose a mathematical formulation to assign participants to corrals such that the interference between participants is minimized, assuming deterministic finish times;
- Design and evaluate corral designs via simulations.

1.3 Problem Description

During the race, WB10K participants are identified by a unique id, referred to as *bib number*. Bib numbers are assigned to participants considering their expected finish sequence for the race (*i.e.*, the lowest bib numbers are expected to arrive at the finish line first). In fact, the lowest bib numbers are reserved for elite athletes, a status earned by their finish time in official international 10K races. The bib numbers for non-elite athletes in the WB10K are currently assigned solely based on an estimated completion time provided by the participants upon registration. Typically, large races assign bib numbers for non-elite participants based on historic data from official events. Unfortunately, in Puerto Rico there is a dearth of historic data from other local official events, which hinders this possibility.

The WB10K distributes participants into five corals, each identified by a color, as shown in **Error! Reference source not found.**. The first corral is the yellow corral, reserved for lite athletes. The remaining, non-elite, participants are assigned to the orange, green, purple, and blue corrals, respectively, depending on their bib number. **Error! Reference ource not found.**depicts the last bib number for each corral. It is important to highlight that participants are not organized in any specific order within corrals.

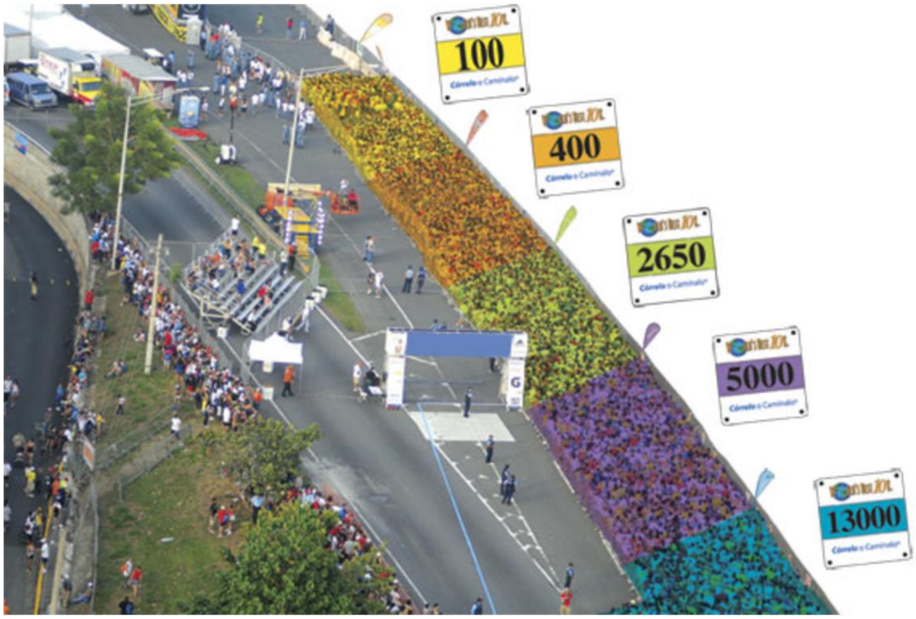


Figure 1.1 WB10K Corral Assignment [3}

The current corral assignment policy divides participants in two groups, elite and non-elite. The elite athletes are assigned a bib number from 1 to 100. The rest non-elite athletes are assigned numbers from 101 and over. Note that participants who register with a small estimate time are assigned a lower bib number and therefore are placed on a corral closer to the start line. The participants receive their bibs during the fitness festival that is celebrated two to three days before the race. The day of the race participants arrive at the Teodoro Moscoso Bridge and are directed to the corral area. The corrals are divided by fences and each participant should enter the corral assigned to them. Even though the WB10K has an establish corral policy there is no system in place to ensure that all participants that enter a corral were actually assign to it.

There are two main challenges with the honor-based bib assignment strategy currently used at WB10K:

1. Participants may provide inaccurate finish time estimates for the race because of inexperience, and
2. Participants may intentionally underestimate their finish time to obtain a smaller bib number and gain access to the front corrals.

There are different motives for a participant to want to be assign to front corral. Being assigned to a front corral has two main advantages:

1. Participants seek to avoid interference - because there are fewer racers in front, and,
2. Participants finish the race earlier, which is important for some participants as the race ends close to sundown on a Sunday.

Furthermore, some participants get anxious because it takes several minutes for all participants to cross the start line after the race begins. On the other hand, if slower participants are assigned to the front corrals (because they underestimated their race completion time) then the interference between participants increases. Historically participants have been underestimating their finish times in order to be assigned a front coral, Marginal Plot of WB10K Finish Times versus Estimated Times for 2014presents a Marginal Plot of the estimates provided by the participants of the 2014 edition and their actual finish times.

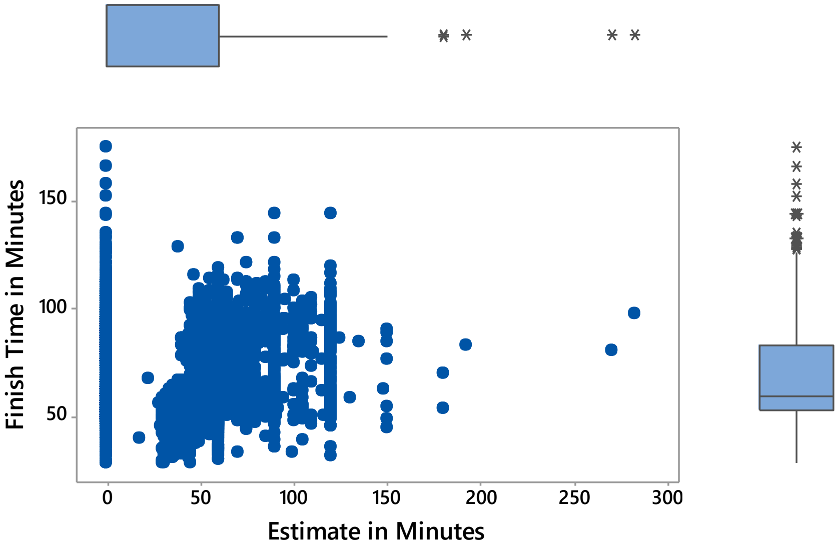


Figure 1.2 Marginal Plot of WB10K Finish Times versus Estimated Times for 2014

Extraordinarily, from Figure 1.2 one can observe that some participants estimate that they will take 0 minutes to complete a 10K race. On the other hand, others overestimate their finish time. Both phenomena, under-estimating and over estimating, will have an impact in the interferences during the event. By examining the marginal plot it is evident that the estimate provided by the participants is a poor predictor for the actual finish time. The use of the current method to assign starting positions may be responsible for a significant interference between participants.

1.4 Thesis Structure

This thesis is organized as follows. Chapter 0 contains the literature review. Chapter 0 summarizes the historical data and quantifies the amount of interferences in past editions of the WB10K. Chapter 0 presents a regression methods proposed to predict the completion time for the race. Chapter 0 presents a mathematical model to minimize interferences during the WB10K that serves as a lower bound for the problem. Chapter 6 defines the Monte Carlo simulation used to determine a corral policy and its validation. Chapter 0 describes how participants should be scheduled to corrals for the 2014 race using the resulting regression analysis. Lastly, Chapter 0 presents the conclusions of this study.

2. Literature Review

2.1 Starting positions at marathons

There is a dearth of literature in marathon and race starting position arrangement. The common practice in large marathons is to require official records from other races in order to assign participants to corrals. Based on the best official historical times registered, a bib number, and hence a corral, is assigned to the runner. Races like the Disney Marathon [4] and Chicago Marathon [5] have adopted this practice. The WB10K welcomes all types of participants and this practice would require the race to be an elite athlete only event. In addition, most of the local participants only run sports events that take place in Puerto Rico. Hence, in most athletic events in Puerto Rico there is a lack of historical record system.

On the other hand, the Philadelphia Marathon [6] allows participants to switch to their desired corral. If the participant wants to move to a slower corral he or she can do it without requiring any official action. But if he or she wishes to transfer to a faster corral it is only required that they present their bib number at the solution center at the Expo. The main difference between this race and the WB10K is that corrals are released in waves with a predetermined time between them. The same practice is adopted by the New York City Marathon [7].

A totally different strategy is used by the Boston Marathon [8]. In order to participate in this particular marathon a runner has to submit a proof of time in a full marathon and comply with the qualifying time. Only the fastest runners are allowed to participate in the Boston Marathon. This particular strategy cannot be employed in the WB10K given that it welcomes both the professional runner and the casual participant. Note that a main difference between marathons and 10 kilometer races is that participants finish faster and by consequence the event has a shorter duration.

2.2 Participants of Athletic Events

Some authors have studied the behavior and profile of those who participate in marathons. For starters, Frederick et. al [9] studied the motivation of runners this led them to classifying runners in two groups; “fun runners” and “serious runners”. On the other hand, Griffin [10] states that women, on the contrary of men, tend to run in groups during these events and tend to be less competitive. Hallman et. al [11] conducted a study using data collected from three marathons in Germany in which they found that the overall satisfaction with the event influences the return of participants in marathon. On the other hand, Knechtle et. al [12] performed a statistical study to identify the age effect on the performance during 100-km ultra-marathon finding that the best age, for both female and male runners, ranges between 39 and 40.

Recent studies have been conducted to better understand the race mass behavior. Particularly Rodriguez et al [13] studied the race mass behavior of the Chicago Marathon, after the terrorist attacks on the Boston Marathon, to understand if there is a diffusion pattern in marathons. In this study the authors used differential equations to model the diffusion effect during the marathon. They found that as the mass of runner’s sticks together until it passes the 10 kilometer checkpoint, this is where it start dispersing. On another study, Alvarez-Ramirez and Rodriguez [14], study the pattern and dynamic of the runners during a race. In the study they classify elite runners as outliers and seek to understand an 80-85% of the mass of runners. They found that there are many external factors that affect the individual pattern of runners. They establish that some runners participate in races for different reasons; overcoming his or her past records, socialize with friends, winning, among others. Even though authors have conducted studies about marathons there are no publications regarding corral assignments or participants’ starting positions during such events.

2.3 Pedestrian Behavior

Additional studies have been conducted to explain pedestrian behavior that could be used to model participants of the WB10K. Some studies focus on pedestrian behavior during emergency situation. Sime [15], presents different historical events and analyzes the pattern of evacuation of the citizens. Another study by Desmet and Gelenbe [16] uses discrete simulation and queuing theory to replicate the evacuation of building and determine the best location for emergency sensors. Alternatively Al-Kodmany [17] evaluates different methods of pedestrian simulation to replicate the behavior of masses during the Hajj, a religious event that takes place in Saudi Arabia each year. This study implements a combination of fluid simulation to replicate the individual movement within the crowd and cellular simulation, where the masses are divided in cells to study the mass displacement. Usher and Strawderman [18] and Usher et. al [19] simulate pedestrian movement taking into consideration the individual. The former, develop probabilistic models to take into account the pedestrian speed and direction and scales the model to represent the mass, whereas the latter implements various equations to simulate decisions made by pedestrian like for example avoiding other pedestrians and obstacles in front of them.

All the studies mentioned used very complex methods to model the behavior of individual. None of these studies can aid directly to the improvement of bib numbers and corral assignments.

2.4 Studies to Improve Sport Events

Other authors have used engineering and scientific techniques to study and improve sport events. For example Bekker and Lotz [20] designed a mathematical model to aid the strategic planning of Formula One race cars. In this study they use a database available to the general public containing details of each racer and previous results on different races. Another example would be Atuahene et. al [21] were they model the ticketing and seating process of spectator during large events.

From the literature review no study was found focusing on the design of corrals or the assignment of participants to starting positions.

3. Data Analysis

In this chapter we discuss details of the WB10K and analyze historical data from 2011, 2012, 2013, and 2014 editions.

3.1 WB10K Checkpoints

There are four checkpoints in the WB10K, located at the 0K, 3K, 8K, and 10K marks, as shown in Figure 3.1. The first and last checkpoints correspond to the start and finish times. Hence, the time registered at the 0K checkpoint quantifies how long it took a participant to reach the start line from the assign corral. The clock time registered at the 10K checkpoint is known as the gun time. The actual finish time for a racer, known as the net time, is the time from the 10K checkpoint minus the time from the 0K checkpoint.

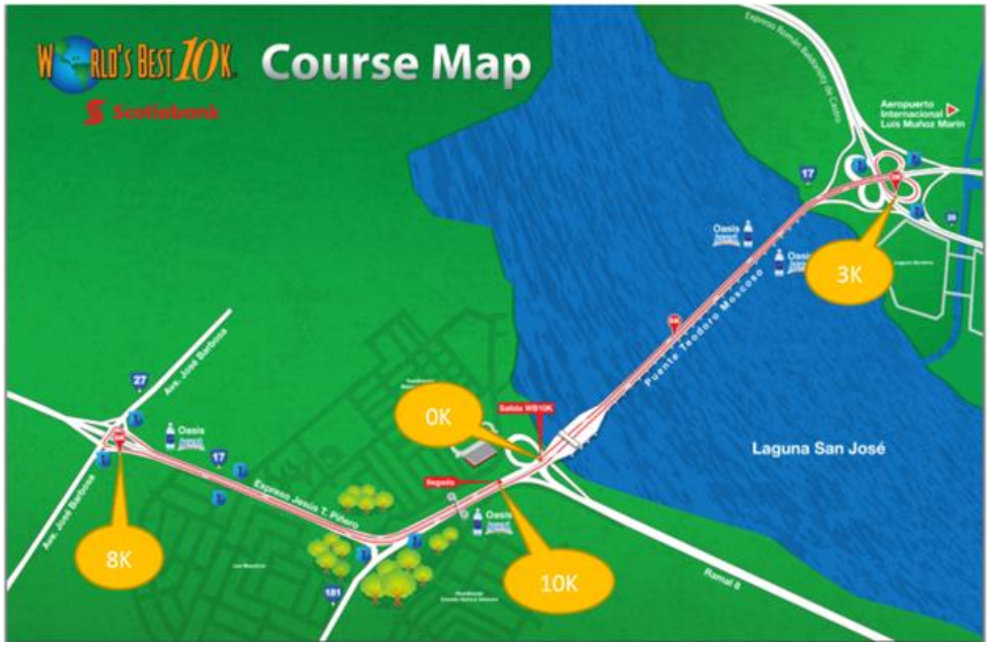


Figure 3.1 WB10K Route Map with Checkpoints [3]

3.2 Historical Finish Times

During the WB10K each participant carries a radio-frequency identification (RFID) tracking chip inside their bib. This chip allows automated and real-time tracking of athletes at each checkpoint. The time registered at the 0K checkpoint indicates how long it took a participant to reach the start line after the race starts. The time registered at the 10K checkpoint is known as the gun time (finish time). The actual finish time for a participant, known as the net time, is the time from the 10K checkpoint minus the time from the 0K checkpoint. Table 3.1 presents the net times at each checkpoint for the 2011-2013 WB10K races in terms of top percentiles.

Table 3.1 Net Times at Checkpoints as Top Percentiles

Percentile	3K	8K	10K
10th	00:12:16	00:43:10	00:56:43
20th	00:15:06	00:47:35	01:03:21
30th	00:17:35	00:51:31	01:09:13
40th	00:20:03	00:56:09	01:15:37
50th	00:22:34	01:01:08	01:22:21
60th	00:25:41	01:07:01	01:30:02
70th	00:32:12	01:13:24	01:38:24
80th	00:36:30	01:20:09	01:47:17
90th	00:41:09	01:28:18	01:58:14

From Table 3.1 it may be observed that the top ten percent of non-elite participants reach the 3k checkpoint in less than 12 minutes and finish the race in less than 56 minutes. Also, it can be appreciated that fifty percent of non-elite participants take longer than one hour and twenty minutes to complete the 10 kilometers.

3.3 Assumptions

The modelling assumptions made throughout this study are:

1. The data provided by participants and the WB10K organizing committee is reliable and complete;
2. Interference between participants does not affect the finish time; and
3. Interference among participants is only measured at predetermined checkpoints.

The first assumption suggests that participants provided accurate information upon registration (*e.g.*, age, gender, name). Furthermore, it is assumed that the automated time keeping mechanism (*i.e.*, Time Tracker) records times correctly (although we were warned that in 2012 there was a malfunction of some timekeeping devices, which forced us to disregard some corrupted data from that year). The second assumption is necessary to estimate the participants' finish time irrespectively of their corral assignment. Notice that although bib numbers correspond to corrals, it is impossible to predict the specific starting location of a participant within their corral. The third assumption will be further explained in Section 3.4.

3.4 Calculating Interferences

For this study interferences were calculated using the time registered by participant at the four checkpoints. The logic behind this calculations is: if participant A passes a checkpoint after participant B, but reaches the next checkpoint before participant B, it is said that A passed B during that interval. In order to quantify the interferences between participants a Java code was designed to count the number of times a participant passes (i.e., overtakes) another participant in previous editions of the race. The pseudo code is presented in Figure 3.2.

```
Given:
$i, j = indices\ for\ participants$
$k = index\ for\ checkpoints$ {k = 1 ($0K$), k = 2 ($3K$), k = 3 ($8K$), k = 4 ($10K$)}
$t_i^k = time\ at\ checkpoint\ k\ for\ participant\ i$
$C_j = counter\ for\ number\ of\ passes\ by\ j$

For $i$ = 1 to number of participants
        For $j$ = 1 to number of participants
                If $j \neq i$
                    For $k$ = 1 to (number of checkpoints – 1)
                        If $t_i^k < t_j^k$ and $t_i^{k+1} > t_j^{k+1}$
                                $C_j = C_j + 1$
                        End If
                    Next $k$
                End If
        Next $j$
Next $i$
End
```

Figure 3.2 Pseudo Code to Determine Number of Passes During the Race

Notice that in reality two participants running next to each other might be constantly passing each other. In this study we only measure passes at the predetermined checkpoints. Hence, at most one pass may occur between any two participants on each segment between checkpoints as mentioned in the last assumption on Section 3.3. Table 3.2 summarizes the total number of passes between checkpoints for the previous WB10K editions.

Table 3.2 Number of Interferences between Participants

Interval	2011	2012	2013	2014
0K to 3K	8,525,421	9,596,791	7,574,731	12,443,247
3K to 8K	5,059,948	6,314,257	4,740,781	3,221,238
8K to 10K	2,045,301	2,307,470	1,351,365	2,414,538
Total Interferences	15,630,670	18,218,518	13,666,877	18,079,023
Total Participants	9, 405	9,258	7,911	9,050

From Table 3.2 it may be observed that the majority of the interferences occur from the 0K to the 3K checkpoint. Table 3.3 presents more detailed statistics obtained with the Java code. Rows *Min*, *Max,* and *Ave* refer to the participants with the least, most, and average number of passes. The columns in Table 3.3 present the total number of passes and the percentage of all participants to which this total corresponds.

Table 3.3 Interference Statistics in Terms of Number of Passes

		2011		2012		2013		2014	
		Total	**Percentage**	**Total**	**Percentage**	**Total**	**Percentage**	**Total**	**Percentage**
0K to 3K	***Max***	8,770	93.25	8,713	94.11	4,358	55.06	8,976	99.18
	Min	0	0	0	0	0	0	0	0
	Ave	649	19.05	775	8.16	759	9.6	1,375	15.19
3K to 8K	***Max***	7,222	76.79	8,975	96.94	7,388	93.4	8,934	98.72
	Min	0	0	0	0	0	0	0	0
	Ave	455	13.36	412	4.45	314	3.97	356	3.93
8K to 10K	***Max***	1,336	14.2	9,157	98.9	7,807	98.69	6,407	70.8
	Min	0	0	0	0	0	0	0	0
	Ave	216	6.34	165	1.78	110	1.39	267	2.95
0K to 10K	***Max***	8,921	94.85	9,157	98.9	7,808	98.7	8,976	99.18
	Min	1	0.01	1	0.01	11	0.14	0	0
	Ave	1,294	13.76	1,554	16.78	1,448	18.3	1,998	22.07

From the results presented on Figure 3.3 it can be observed that the average participant has number of interferences that ranges between 1,294 and 1,998 interferences. Furthermore, the majority of interferences during the 2014 edition occurred in the first interval (*i.e.*, between the 0K and 3K checkpoints). These results suggest that the current corral assignment method is not effective. Figure 3.3 presents the percent of interferences per checkpoint for previous editions of the WB10K.

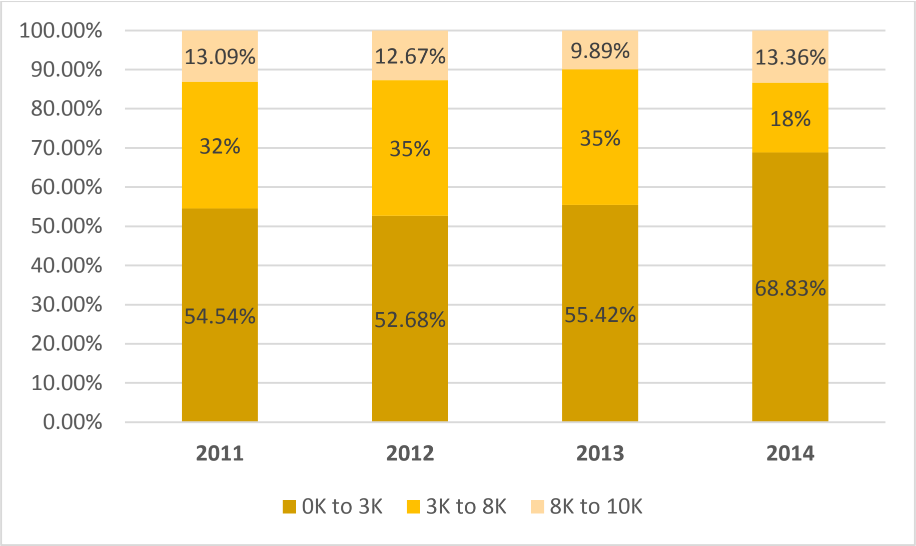


Figure 3.3 Percent of Interferences by Checkpoint for Past Editions of the WB10K

From Figure 3.3 it can be observed that the last four WB10K editions show similar distribution of total interferences per checkpoint, where most interferences occur between the 0K-3K checkpoints. The results presented in Table 3.2 and Table 3.3 confirm that particular emphasis needs to be given to the bib number (and hence corral) assignments to reduce the interference in the first three kilometers. The honor-based strategy has been used in all 4 editions, hence suggesting that the current assignment method may be contributing to the interference among participants. An original though for assigning participants to bib numbers was their expected time for the first three kilometers. However, since passes continue to occur throughout the race it was decided that a more appropriate metric to assign participants to bib numbers was their race finish time. The next Chapter presents regression analyses to estimate participants finish time.

4. Corral Assignment

This chapter presents and compares four different regression methods to estimate non-elite participants finish times.

4.1 *Variables Available for Regressions*

When a participant registers for the WB10K they fill out a form providing their name, estimated time to complete the 10 kilometers, gender, and age. This information is stored in a database and paired to their finish times after the event. The registration names of the participants were matched within two different editions of the WB10K to determine if a participant had ran before (*i.e.*, a returning participant) in order to incorporate the historical time to finish the race into a regression model to predict non-elite participants' finish time in 2013 (*i.e.*, Finish Time).

In other words, data from the 2011 and 2012 races will be used to predict the 2013 finish time, which is our dependent variable. The first analysis was to visualize the data via a Matrix Plot, presented in Figure 4.1. In the Matrix Plot finish time for 2013 is labeled *Finish Time*, estimated time for 2013 is *Estimated Time,* gender is *Gender*, age is *Age*, estimated time for 2012 is *Estimate 2012*, finish time for 2012 is *Time 2012*, estimated time for 2011 is *Time 2011*, and finish time for 2011 is *Time 2011*. As a modeling decision we opted to assign an *Estimated Time* of zero to participants who did not estimate their finish time. Historical data predating 2011 is not considered in the analyses as the time tracking technology had not matured enough to be considered reliable. Note that in the matrix plot the variable that is parallel to the x-axis is the one presented on the x-axis of the graph in the particular space within the matrix. The same applies to thy-axis, the variable parallel to the y-axis are the ones represented by the y-axis.

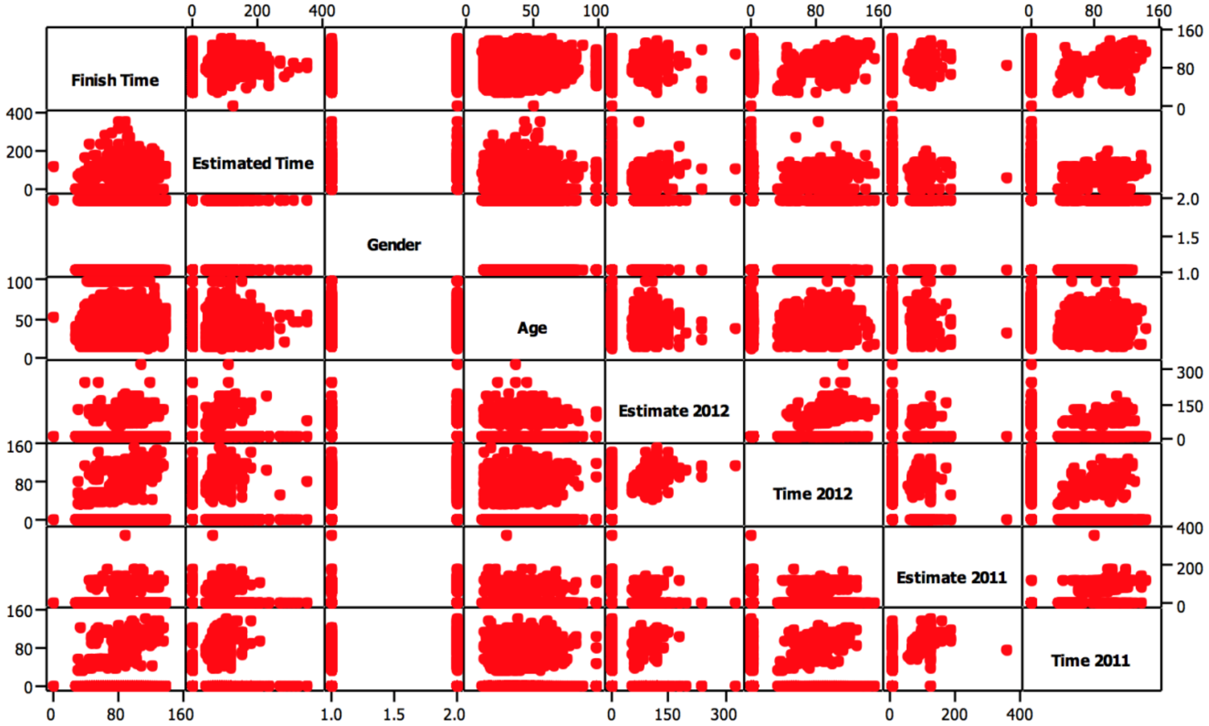


Figure 4.1 Matrix Plot

(Times are showed in minutes, Gender is 1 for male and 2 for female, and Age is in years)

The first row of the matrix plot in Fig 4.1 presents the relationship between the information available and the 2013 finish time (Finish Time). Estimated Time, the time estimations provided by participants in 2013, appears to have a weak linear relation to the Finish Time. On the other hand, the previous years' estimates (Estimate 2011 and Estimate 2012) show no clearly defined relation with the dependent variable (Finish Time). Based on this observation, the time estimates for the previous years will not be taken under further consideration for the regression analysis. Time 2011 and Time 2012 both show a similar linear tendency with respect to Finish Time. The resemblance between Time 2011 and Time 2012 suggests that they could be merged into a single coefficient (e.g., considering their average). The average of Time 2011 and Time 2012 is henceforth termed *Averaged Time*. The aggregation of finish times for previous races is also desirable as some participants may have only participated in one of the races. Lastly Age and Gender do not show a strong relationship to Finish Time. Instead of eliminating these coefficients from future consideration, we opted to maintain their interaction (i.e., Age*Gender). Figure 4.2

shows the resulting scatterplots for the coefficients included in the regression based on the analyses from the matrix plot.

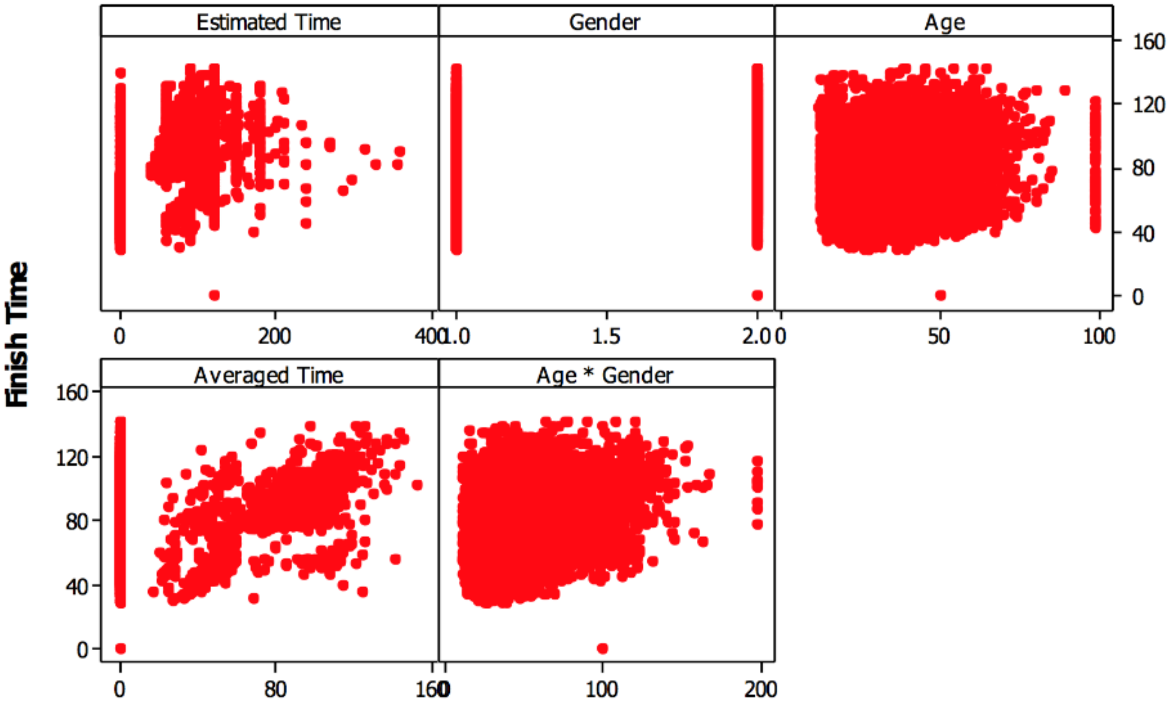


Figure 4.2 Scatterplot of Finish Time versus Estimated Time, Gender, Age, and Averaged Time

(all times are showed in minutes, Gender is 1 for male and 2 for female, Age is in years)

In Figure 4.2 the first plot presented is equivalent to the position (2,1) in the matrix plot in Fig. 4.1. Just like in the matrix plot it can be appreciated that the linear relationship between Finish Time and Estimate Time is not clear, it even suggest that there is no relationship. Note that, for this figure, the y-axis is represented by the finish time and the x-axis by the label on top of each plot. It may also be observed that the interaction between *Age* and *Gender* results in a more defined tendency than each variable individually. Also, the average of past finish times (*Averaged Time*) shows a noteworthy difference between the data with value of zero and the remaining data. The zero values for *Averaged Time* correspond to those who did not participate on previous races.

Before creating the regression models, each variable will be codified in order to understand the contribution attributed to each variable in the model. Codified variables are obtained by converting each factor to the same units. The codification scheme used in this study limits each variable to the range [-1, 1], where the median value is zero. To codify the variables we used a linear interpolation within the ranges. In the regression models *F* is *Finish Time*, A_c is the codified *Age*, G_c is the codified *Gender*, AT_c is the codified *Averaged Time*, E_c is the codified *Estimated Time* (for 2013), and the interactions between variables will be represented with an asterisk between the two variables (e.g. the interaction between *Age* and *Gender* would be A_c*G_c).

4.2 Age-Grade

The age-grade divides participants by age groups and assigns the corresponding average and the median finish times as coefficients in the regression model. The term age-grade is taken from runners forums where they used historical times, age and gender to calculate a runners age-grade. The WB10K age-grade was calculated with data from 2011 to 2012, dividing participants by age, gender, and classifying them as new or returning. Table 4.1 presents the corresponding age-grade using the WB10K data from 2011 to 2012.

Table 4.1 WB10K 2011-2012 Age-grade (in minutes)

	New Participants				Returning Participants			
	Males		Females		Males		Females	
	Average (min.)	Median (min.)	Average (min.)	Median (min.)	Average (min.)	Median (min.)	Average (min.)	Median (min.)
0-19	71.00	67.30	96.48	98.18	71.40	67.27	95.88	97.52
20-29	65.83	61.99	81.12	78.23	66.33	62.45	83.35	82.82
30-39	68.10	63.38	83.59	81.90	66.91	61.96	83.67	81.95
40-49	70.49	66.08	86.00	85.23	69.39	64.88	85.80	85.45
50-59	74.49	70.12	94.79	95.07	71.73	67.37	92.47	94.38
60-69	83.24	82.43	16.63	97.73	79.61	77.23	102.65	104.32
70-79	85.15	92.71	105.02	103.43	85.25	84.20	107.47	110.10
80-89	82.50	84.41	92.79	101.25	86.98	73.23	96.02	99.48
90+	76.07	68.79	96.59	96.90	73.93	70.12	95.63	100.04

From Table 4.1 it may be observed that the different age and gender groups have diverse averages and medians for their finish times. Note that older participants (*i.e.*, older than 60 years) have a greater average and median, this may indicate that they take longer to finish the race. On the other hand, it may be appreciated that the female group is slower in comparison to the male group. Also, in most cases returning participants have lower average and median than the new participants.

4.3 Estimating Finish Times

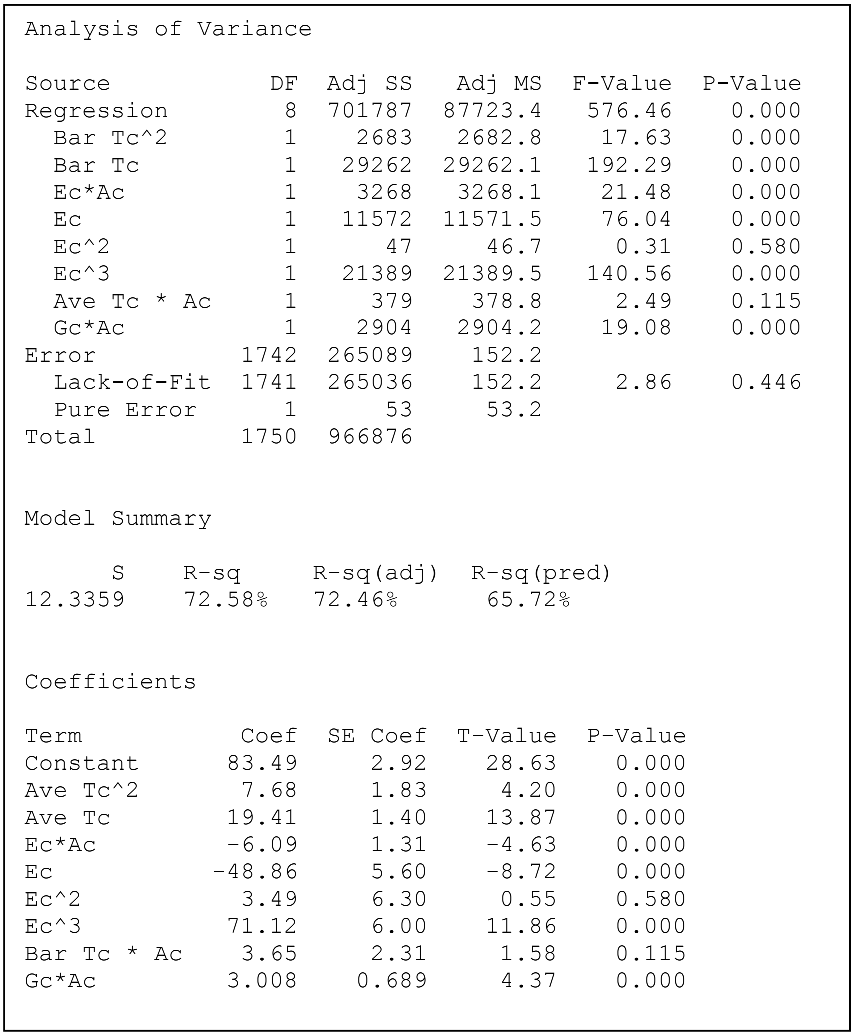
In this section four regression-based methods to estimate finish times for non-elite participants are compared. The first method, Method 1, is composed of two regressions to predict finish times: one for the returning participants (*i.e.*, participants that already have recorded times at the WB10K), and another for new participants. The regressions used include participant estimates of zero minutes as predictors in both regression models. Although this method may yield good statistical results, the model considers those unrealistic estimates provided by participants as part of the model. Intuitively, participants' estimates of zero minutes to complete the race should be treated as categorical variables, instead of as numerical variables. When used as a categorical variable, this information would be used as an indicator that the value provided is unrealistic. Hence, three additional models are proposed to overcome this limitation.

The second method, Method 2, augments Method 1 by substituting unrealistic finish time estimates (such as zero minutes) with the fastest finish time registered by a non-elite participant in 2011 and 2012. The third method, Method 3 further discriminates small unrealistic finish time estimates by incorporating age-grade to the regressions. The fourth method, Method 4, extends the third method by subdividing participants into two sub-groups: the ones with realistic estimates and non-realistic estimates. These four methods are then compared with Status Quo, the current method used by the WB10K organization, which uses finish time estimates that are voluntarily provided by participants upon registration.

Methods 1, 2, 3, and 4 use regression models constructed in Minitab 17 to predict the time it will take non-elite participants to complete the 10 kilometers. Data from the 2011 and 2012 editions were used to generate all regression models using 2013 edition finish times as the response variable. The process to generate the regression models was iterative, which meticulously removed non-significant factors from the models until a model where all predictors were statistically significant at the 90% level. The predictors evaluated for the regressions are limited to the information available in the WB10K database. The four regression methods are detailed in the next subsections.

4.4 Method 1

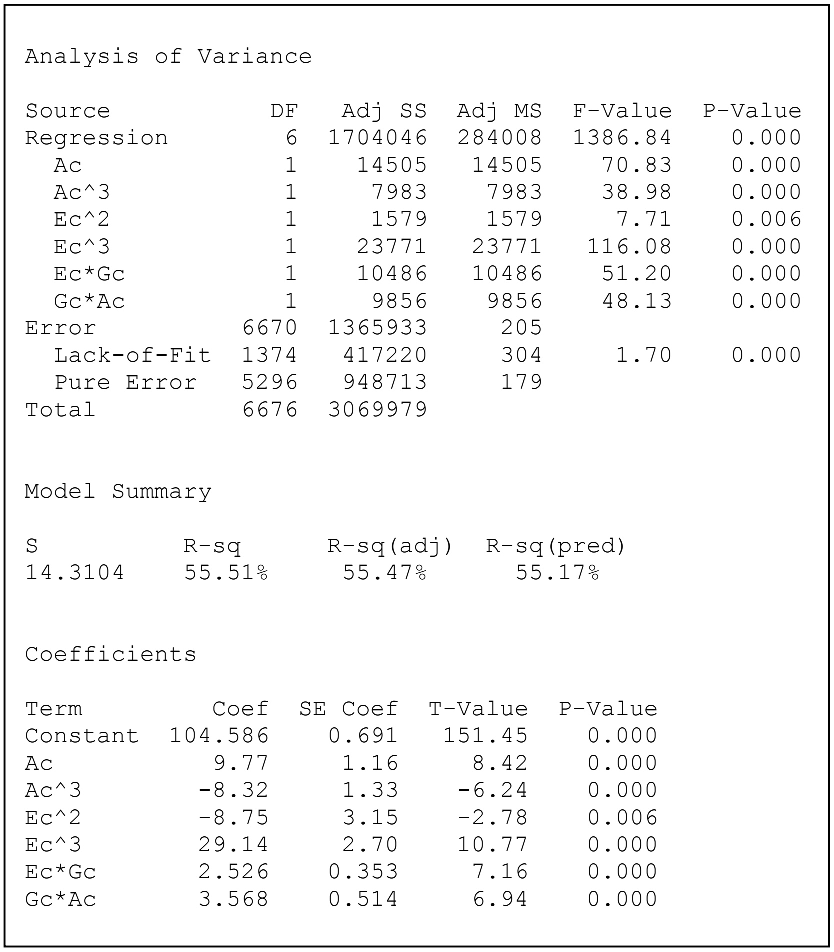
To reduce estimation errors the zero-values add to the model, the participants will be divided into two groups: the ones with non-zero Averaged Time (*i.e.*, previous participation) and those that did not run (or did not finish) in 2011 and 2012. Dividing the participants implies creating two different regression models. These regression models will be label as: for participants with registered times or *returning* participants and for *new* (or no-time) participants. Figure 4.3 presents the Analysis of Variance (ANOVA) and model summary for the final regression with returning participants. Figure 4.4 presents the equivalent ANOVA for new participants. The final regressions were obtained by methodically removing non-statistically significant variables from the original models in a step-wise manner.



```
Analysis of Variance

Source           DF  Adj SS   Adj MS  F-Value  P-Value
Regression        8  701787  87723.4   576.46    0.000
  Bar Tc^2        1    2683   2682.8    17.63    0.000
  Bar Tc          1   29262  29262.1   192.29    0.000
  Ec*Ac           1    3268   3268.1    21.48    0.000
  Ec              1   11572  11571.5    76.04    0.000
  Ec^2            1      47     46.7     0.31    0.580
  Ec^3            1   21389  21389.5   140.56    0.000
  Ave Tc * Ac     1     379    378.8     2.49    0.115
  Gc*Ac           1    2904   2904.2    19.08    0.000
Error          1742  265089    152.2
  Lack-of-Fit  1741  265036    152.2     2.86    0.446
  Pure Error      1      53     53.2
Total          1750  966876


Model Summary

      S    R-sq     R-sq(adj)  R-sq(pred)
12.3359    72.58%   72.46%      65.72%


Coefficients

Term            Coef  SE Coef  T-Value  P-Value
Constant       83.49     2.92    28.63    0.000
Ave Tc^2        7.68     1.83     4.20    0.000
Ave Tc         19.41     1.40    13.87    0.000
Ec*Ac          -6.09     1.31    -4.63    0.000
Ec            -48.86     5.60    -8.72    0.000
Ec^2            3.49     6.30     0.55    0.580
Ec^3           71.12     6.00    11.86    0.000
Bar Tc * Ac     3.65     2.31     1.58    0.115
Gc*Ac          3.008    0.689     4.37    0.000
```

Figure 4.3 ANOVA and Model Summary for Returning Participants

All coefficients of the regression for returning participants presented in Figure 4.3 have low p-values, indicating they play a statistically significant role on the regression model. The summary of model shows an adjusted R –squared (*i.e.*, coefficient of determination) of 72.46%, with a standard deviation of the errors of 12.34. This means that the model can explain 72.46% of the variation in the predicted Finish Time; which is considered very satisfactory. The ANOVA shows that all the predictors are statistically significant for the prediction of the finish times, with p-values lower than 0.10. Figure 4.4 presents the ANOVA and model summary for first time participants.



```
Analysis of Variance

Source           DF   Adj SS  Adj MS  F-Value  P-Value
Regression        6  1704046  284008  1386.84    0.000
  Ac              1    14505   14505    70.83    0.000
  Ac^3            1     7983    7983    38.98    0.000
  Ec^2            1     1579    1579     7.71    0.006
  Ec^3            1    23771   23771   116.08    0.000
  Ec*Gc           1    10486   10486    51.20    0.000
  Gc*Ac           1     9856    9856    48.13    0.000
Error          6670  1365933     205
  Lack-of-Fit  1374   417220     304     1.70    0.000
  Pure Error   5296   948713     179
Total          6676  3069979


Model Summary

S          R-sq      R-sq(adj)  R-sq(pred)
14.3104    55.51%     55.47%      55.17%


Coefficients

Term          Coef  SE Coef  T-Value  P-Value
Constant  104.586    0.691   151.45    0.000
Ac           9.77     1.16     8.42    0.000
Ac^3        -8.32     1.33    -6.24    0.000
Ec^2        -8.75     3.15    -2.78    0.006
Ec^3        29.14     2.70    10.77    0.000
Ec*Gc       2.526    0.353     7.16    0.000
Gc*Ac       3.568    0.514     6.94    0.000
```

Figure 4.4 ANOVA and Model Summary for Final Regression of New Participants

The final regression model that predicts the Finish Time of new participants yields p-values below 0.10, indicating that the coefficients and the predictors are significant to determine the finishing times at the 90% significance level. This model is able to explain 55.47% of the estimation error, with a standard deviation of 14.31. Although this coefficient of determination is lower than the one for returning participants, it is still considered acceptable to predict Finish Time for new participants, particularly given there is less information on new participants. Figure 4.5 and Figure 4.6 present the residual plots for each regression model.

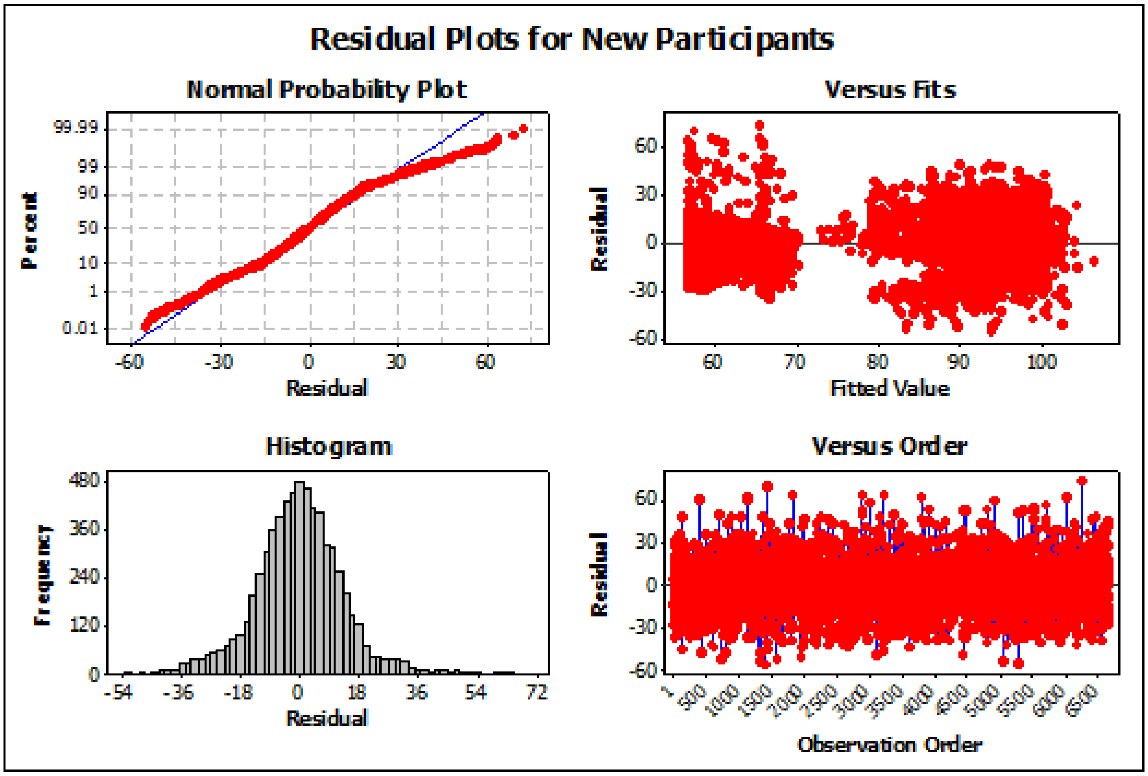


Figure 4.5 Residual Plots for Returning Participants

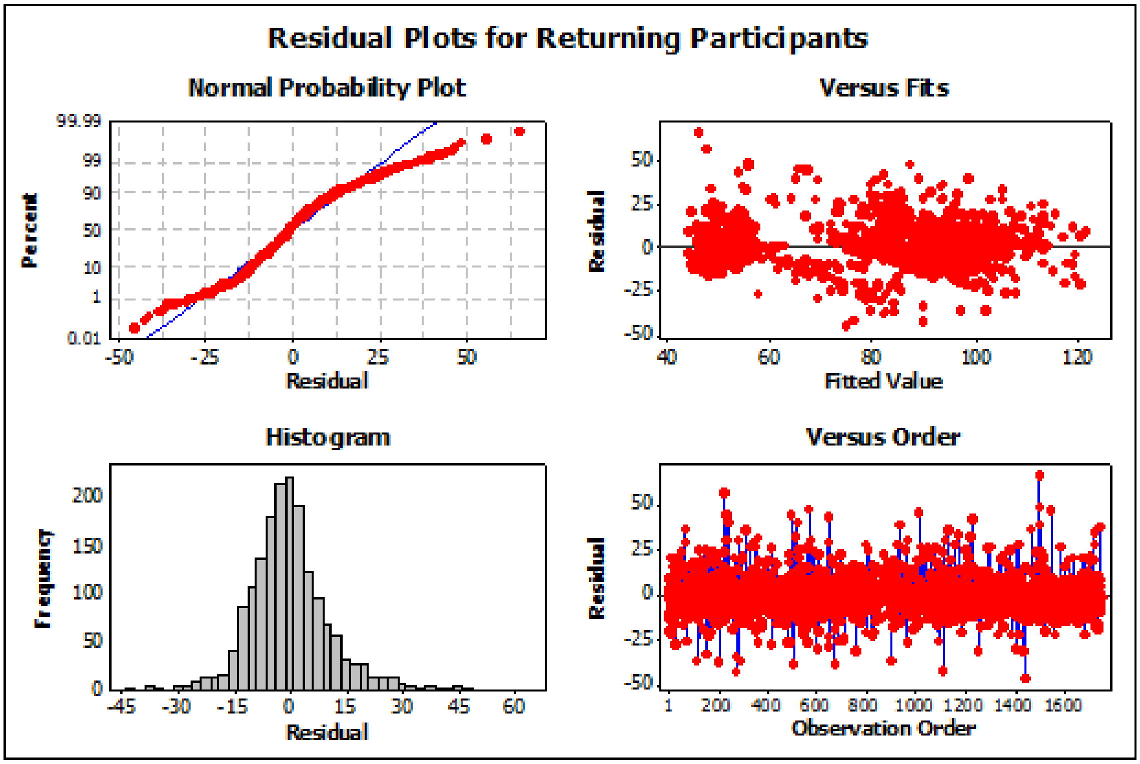


Figure 4.6 Residual Plots for New Participants

The four different residual plots in Figure 4.5 and Figure 4.6 include Normal probability plots and histograms. The Normal probability plot suggests that the Normality assumption of estimation errors is valid for both models. The versus fit graphs in Figure 4.5 and Figure 4.6 show an approximately even distribution of estimation errors below and above zero, suggesting that estimation errors have a mean of zero. These graphs are key to validate the basic assumptions of independence and distribution of errors required for the statistical analyses performed. The following equations result from Method 1.

$$\begin{aligned} New\ Participants \\ &= 104.586 + 9.77\,Ac - 8.32\,Ac^3 - 8.75\,Ec^2 + 29.14\,Ec^3 \\ &+ 2.526\,Ec * Gc + 3.568\,Gc * Ac \end{aligned} \qquad (1)$$

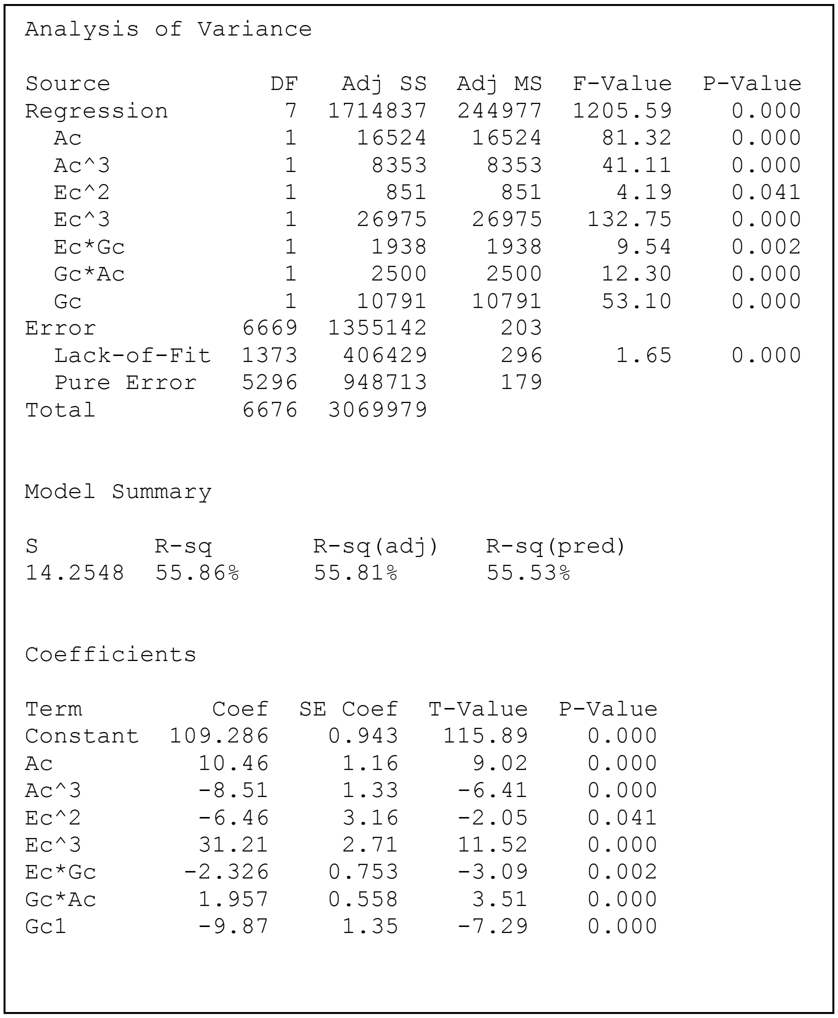
$$
\begin{aligned}
Returning\ Participants \\
&= 83.49 + 7.68\, Ave\, Tc^2 + 19.41\, Ave\, Tc - 6.09\, Ec * Ac \\
&- 48.86\, Ec + 3.49\, Ec^2 + 71.12\, Ec^3 + 3.65\, Bar\, Tc * Ac \\
&+ 3.008\, Gc * Ac
\end{aligned}
\tag{2}
$$

For Method 1 equation one and two would be used to estimate a participant's finish time. First participants must be classified as new or returning. All variables must be coded before substituting in the equations. From Eq. 1 and Eq. 2 it can be seen that the estimated time a participant voluntarily submits (for both returning and new) is incorporated as a multiplying effect. In other words participants usually take longer than what they estimate.

4.5 Method 2

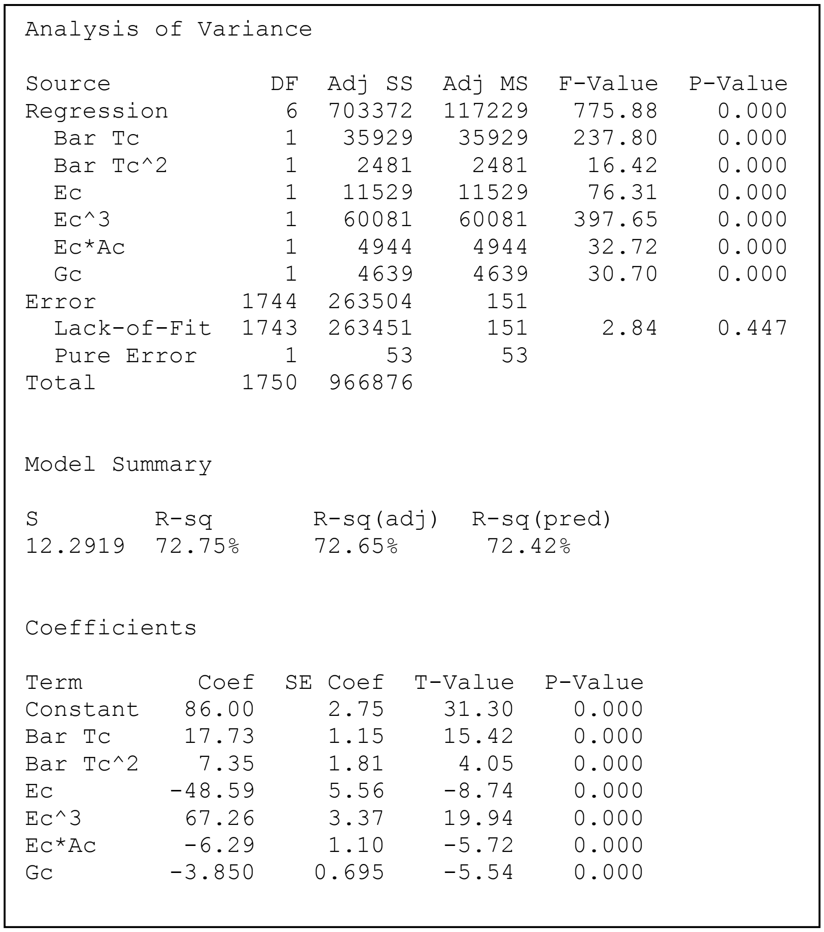
This method augments Method 1 by substituting unrealistic finish time estimates, times below 30 minutes, with the fastest finish time registered by a non-elite participant in 2011 and 2012. Analogously, finish time estimates that are unrealistically long are replaced by the largest registered finish time in 2011 and 2012. A realistic estimate for a non-elite athlete was considered between 30 and 170 minutes, based on the best and worst historical time by a non-elite participant between 2011 and 2012.

This method is composed of a regression for returning participants and new (first time) participants. Another main difference of these regressions when compared to the ones from the first method is that the in order to codify the estimation provided by participants the zeroes are not considered as the minimum, making the values range from below -1. The variables used in these regressions are represented as follows; Ec is the estimate provided by the participants upon registration, Ac is the age, and Gc is the gender. Figure 4.7 and Figure 4.8 present the ANOVA and model summary for returning participants and new participants.



```
Analysis of Variance

Source           DF   Adj SS   Adj MS  F-Value  P-Value
Regression        7  1714837   244977  1205.59    0.000
  Ac              1    16524    16524    81.32    0.000
  Ac^3            1     8353     8353    41.11    0.000
  Ec^2            1      851      851     4.19    0.041
  Ec^3            1    26975    26975   132.75    0.000
  Ec*Gc           1     1938     1938     9.54    0.002
  Gc*Ac           1     2500     2500    12.30    0.000
  Gc              1    10791    10791    53.10    0.000
Error          6669  1355142      203
  Lack-of-Fit  1373   406429      296     1.65    0.000
  Pure Error   5296   948713      179
Total          6676  3069979


Model Summary

S         R-sq       R-sq(adj)   R-sq(pred)
14.2548   55.86%     55.81%      55.53%


Coefficients

Term         Coef  SE Coef  T-Value  P-Value
Constant  109.286    0.943   115.89    0.000
Ac          10.46     1.16     9.02    0.000
Ac^3        -8.51     1.33    -6.41    0.000
Ec^2        -6.46     3.16    -2.05    0.041
Ec^3        31.21     2.71    11.52    0.000
Ec*Gc      -2.326    0.753    -3.09    0.002
Gc*Ac       1.957    0.558     3.51    0.000
Gc1         -9.87     1.35    -7.29    0.000
```

Figure 4.7 ANOVA and Model Summary for New with Gc as Categorical



```
Analysis of Variance

Source           DF   Adj SS   Adj MS  F-Value  P-Value
Regression        6   703372   117229   775.88    0.000
  Bar Tc          1    35929    35929   237.80    0.000
  Bar Tc^2        1     2481     2481    16.42    0.000
  Ec              1    11529    11529    76.31    0.000
  Ec^3            1    60081    60081   397.65    0.000
  Ec*Ac           1     4944     4944    32.72    0.000
  Gc              1     4639     4639    30.70    0.000
Error          1744   263504      151
  Lack-of-Fit  1743   263451      151     2.84    0.447
  Pure Error      1       53       53
Total          1750   966876


Model Summary

S        R-sq       R-sq(adj)  R-sq(pred)
12.2919  72.75%     72.65%      72.42%


Coefficients

Term         Coef  SE Coef  T-Value  P-Value
Constant    86.00     2.75    31.30    0.000
Bar Tc      17.73     1.15    15.42    0.000
Bar Tc^2     7.35     1.81     4.05    0.000
Ec         -48.59     5.56    -8.74    0.000
Ec^3        67.26     3.37    19.94    0.000
Ec*Ac       -6.29     1.10    -5.72    0.000
Gc         -3.850    0.695    -5.54    0.000
```

Figure 4.8 ANOVA and Model Summary for Returning with Gc as Categorical

The models yield adjusted R-Squares of 55.81% for the new participants and 72.65% for returning participants. A standard deviation of the error at 12.29 for the new participants and at 0.160608 for the returning participants. The presence of p-values below 0.10 show a confidence level of 90% for the predictors and their coefficients. In the next figures (Figure 4.9 and Figure 4.10) the residual plots for the models are presented.

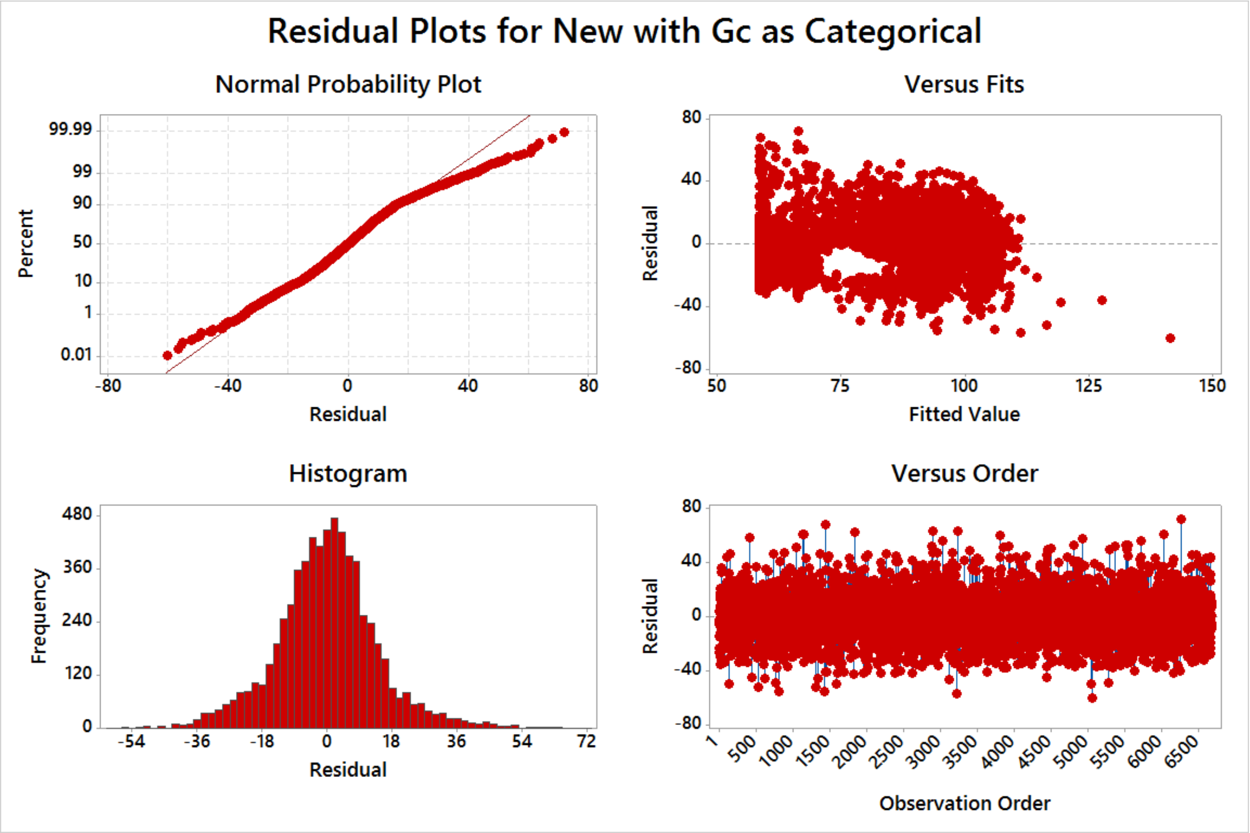


Figure 4.9 Residual Plots for New with Gc as Categorical

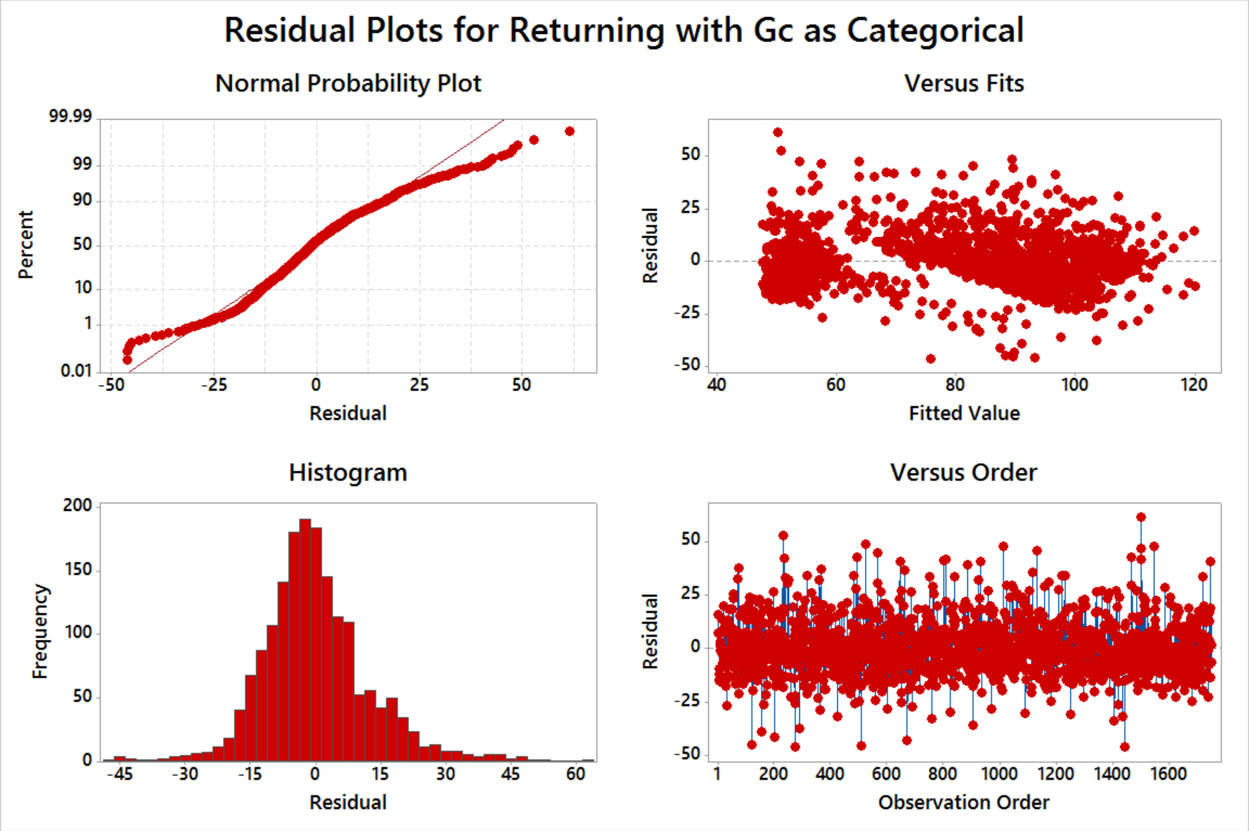


Figure 4.10 Residual Plots for Returning with Gc as Categorical

The plots in Figure 4.9 and Figure 4.10 validate the basic assumptions of independence and distribution of errors required for the statistical analyses performed. The normal probability plots with the histograms support the normality assumption of the errors; whereas the versus fit plot suggest a mean of zero in the residuals. Equations 1 to 6 present the regressions used in in this method.

$$\begin{aligned} New\ Female\ Participants \\ &= 109.286 + 10.46\,Ac - 8.51\,Ac^3 - 6.46\,Ec^2 + 31.21\,Ec^3 \\ &- 2.326\,Ec * Gc + 1.957\,Gc * Ac \end{aligned} \tag{3}$$

$$\begin{aligned} New\ Male\ Participants \\ &= 99.413 + 10.46\,Ac - 8.51\,Ac^3 - 6.46\,Ec^2 + 31.21\,Ec^3 \\ &- 2.326\,Ec * Gc + 1.957\,Gc * Ac \end{aligned} \tag{4}$$

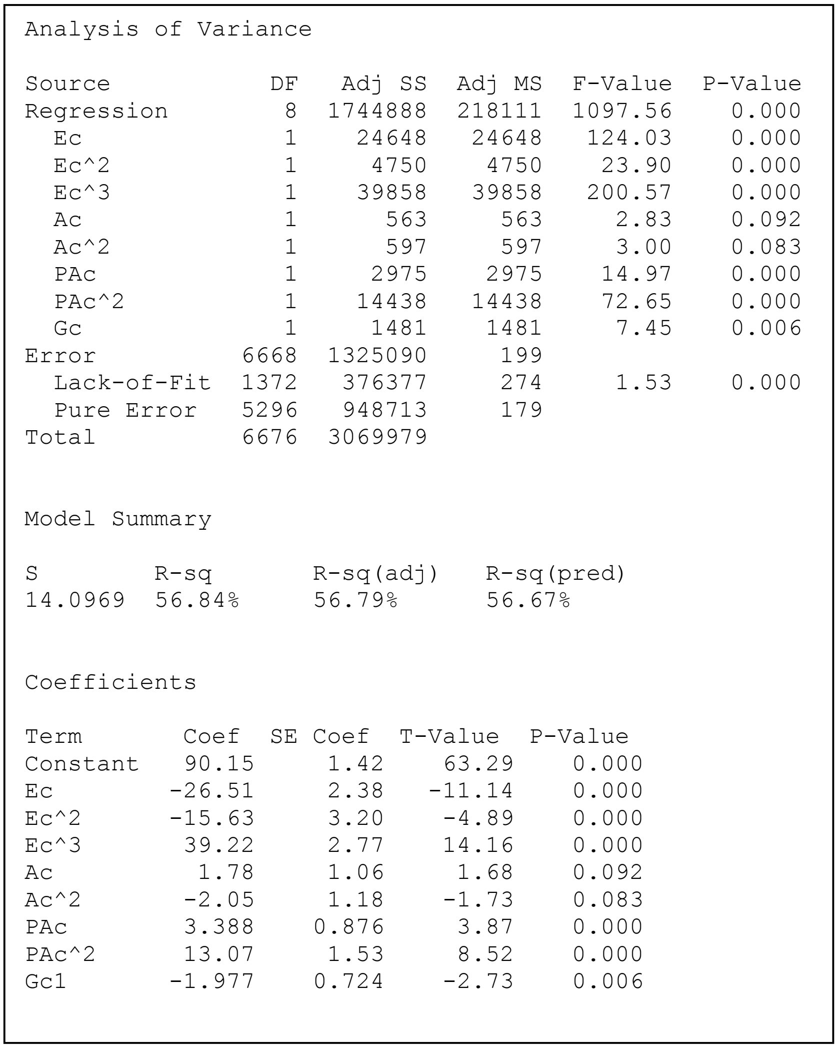
$$Returning\ Female\ Participants = 86.00 + 17.73\, Bar\, Tc + 7.35\, Bar\, Tc^2 - 48.59\, Ec + 67.26\, Ec^3 - 6.29\, Ec * Ac \tag{5}$$

$$Returning\ Male\ Participants = 82.15 + 17.73\, Bar\, Tc + 7.35\, Bar\, Tc^2 - 48.59\, Ec + 67.26\, Ec^3 - 6.29\, Ec * Ac \tag{6}$$

Given that the gender (Gc) is used as a categorical variable in the regressions this results in two regressions for each participants category (returning and new), for a total of 4 equations in this method. Equations 3 and 5 would be used to estimate the time for Female participants, new and returning, respectively. On the other hand, equations 4 and 6 present the formulas to estimate the time it would take new male participants and returning male participants.

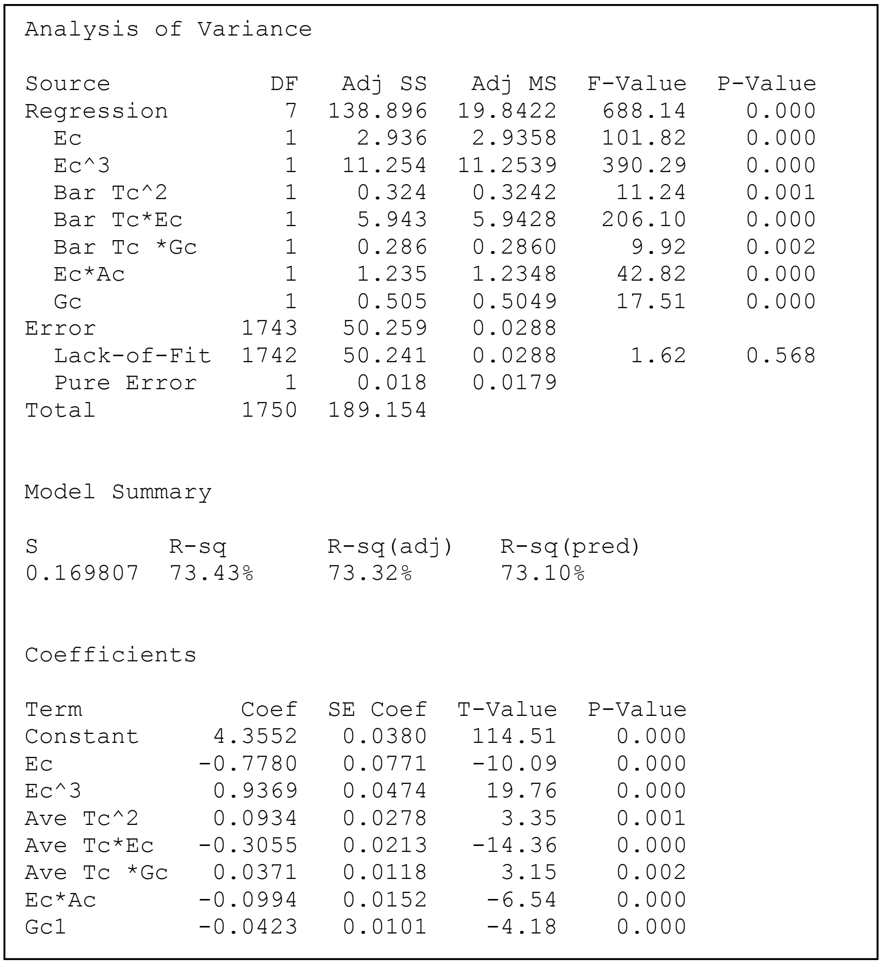
4.6 Method 3

The third method, Method 3 further discriminates small unrealistic finish time estimates by incorporating age-grade to the regressions. Age-grade, as presented on Section 4.2, divides participants by age groups and assigns the corresponding average and the median finish times (calculated with historical data from the WB10K) as coefficients in the regression model. The encoded variables used in these regressions are represented as follows; Ec is the estimate provided by the participants upon registration, Ac is the age, Gc is the gender, and Pac is the average from the age-grade corresponding to the first time participants. The next figures present the ANOVA and model summary for returning participants and new participants.



```
Analysis of Variance

Source           DF   Adj SS  Adj MS  F-Value  P-Value
Regression        8  1744888  218111  1097.56    0.000
  Ec              1    24648   24648   124.03    0.000
  Ec^2            1     4750    4750    23.90    0.000
  Ec^3            1    39858   39858   200.57    0.000
  Ac              1      563     563     2.83    0.092
  Ac^2            1      597     597     3.00    0.083
  PAc             1     2975    2975    14.97    0.000
  PAc^2           1    14438   14438    72.65    0.000
  Gc              1     1481    1481     7.45    0.006
Error          6668  1325090     199
  Lack-of-Fit  1372   376377     274     1.53    0.000
  Pure Error   5296   948713     179
Total          6676  3069979


Model Summary

S        R-sq       R-sq(adj)   R-sq(pred)
14.0969  56.84%     56.79%      56.67%


Coefficients

Term       Coef  SE Coef  T-Value  P-Value
Constant   90.15     1.42    63.29    0.000
Ec        -26.51     2.38   -11.14    0.000
Ec^2      -15.63     3.20    -4.89    0.000
Ec^3       39.22     2.77    14.16    0.000
Ac          1.78     1.06     1.68    0.092
Ac^2       -2.05     1.18    -1.73    0.083
PAc        3.388    0.876     3.87    0.000
PAc^2      13.07     1.53     8.52    0.000
Gc1       -1.977    0.724    -2.73    0.006
```

Figure 4.11 ANOVA and Model Summary for New with Age-Grade



```
Analysis of Variance

Source           DF   Adj SS   Adj MS  F-Value  P-Value
Regression        7  138.896  19.8422   688.14    0.000
  Ec              1    2.936   2.9358   101.82    0.000
  Ec^3            1   11.254  11.2539   390.29    0.000
  Bar Tc^2        1    0.324   0.3242    11.24    0.001
  Bar Tc*Ec       1    5.943   5.9428   206.10    0.000
  Bar Tc *Gc      1    0.286   0.2860     9.92    0.002
  Ec*Ac           1    1.235   1.2348    42.82    0.000
  Gc              1    0.505   0.5049    17.51    0.000
Error          1743   50.259   0.0288
  Lack-of-Fit  1742   50.241   0.0288     1.62    0.568
  Pure Error      1    0.018   0.0179
Total          1750  189.154


Model Summary

S          R-sq       R-sq(adj)   R-sq(pred)
0.169807   73.43%     73.32%      73.10%


Coefficients

Term           Coef  SE Coef  T-Value  P-Value
Constant     4.3552   0.0380   114.51    0.000
Ec          -0.7780   0.0771   -10.09    0.000
Ec^3         0.9369   0.0474    19.76    0.000
Ave Tc^2     0.0934   0.0278     3.35    0.001
Ave Tc*Ec   -0.3055   0.0213   -14.36    0.000
Ave Tc *Gc   0.0371   0.0118     3.15    0.002
Ec*Ac       -0.0994   0.0152    -6.54    0.000
Gc1         -0.0423   0.0101    -4.18    0.000
```

Figure 4.12 ANOVA and Model Summary for Returning with Age-Grade

In Figure 4.11 and Figure 4.12 it can be observed that models yield adjusted R-Squares of 56.79% for the new participants and 73.32% for returning participants. The presence of p-values below 0.10 show a confidence level of 90% for the predictors and their coefficients. Figure 4.13 and Figure 4.14 present the residual plots for the models are presented.

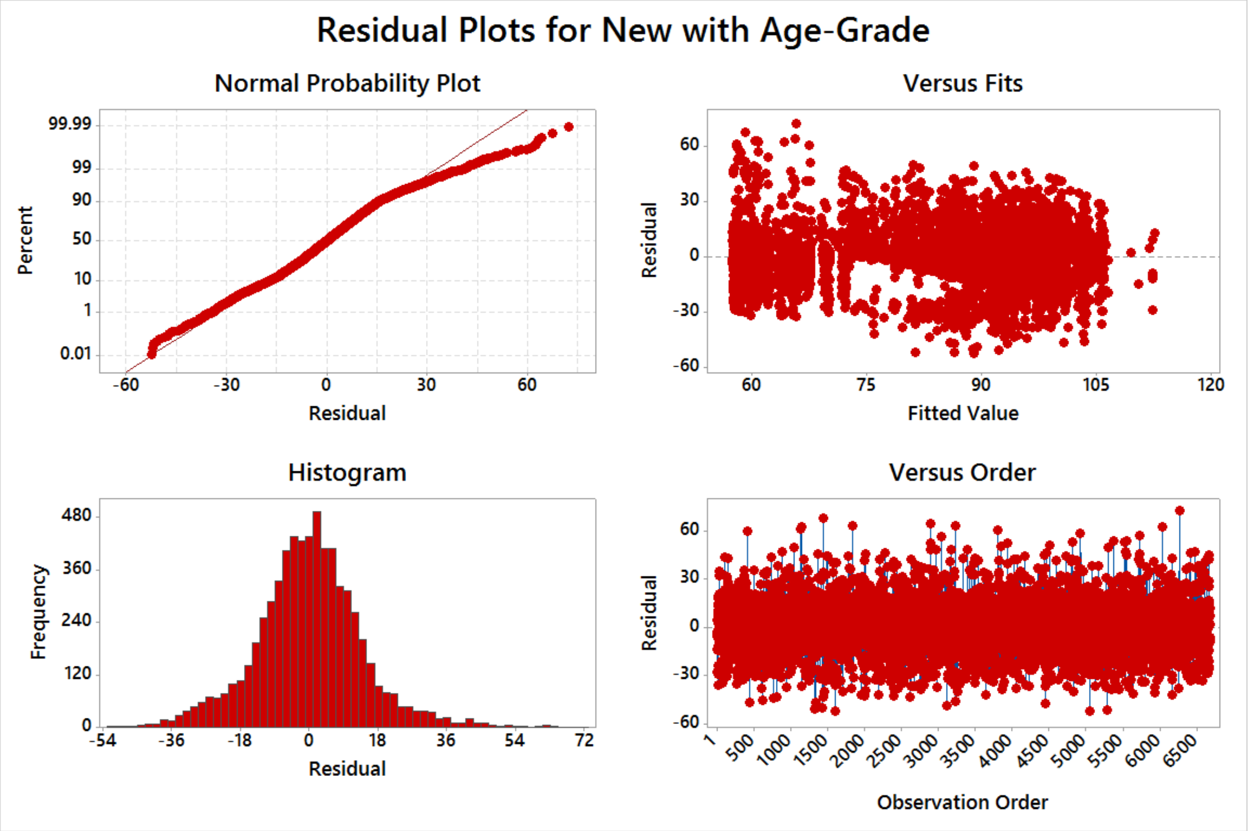


Figure 4.13 Residual Plots for New with Age-Grade

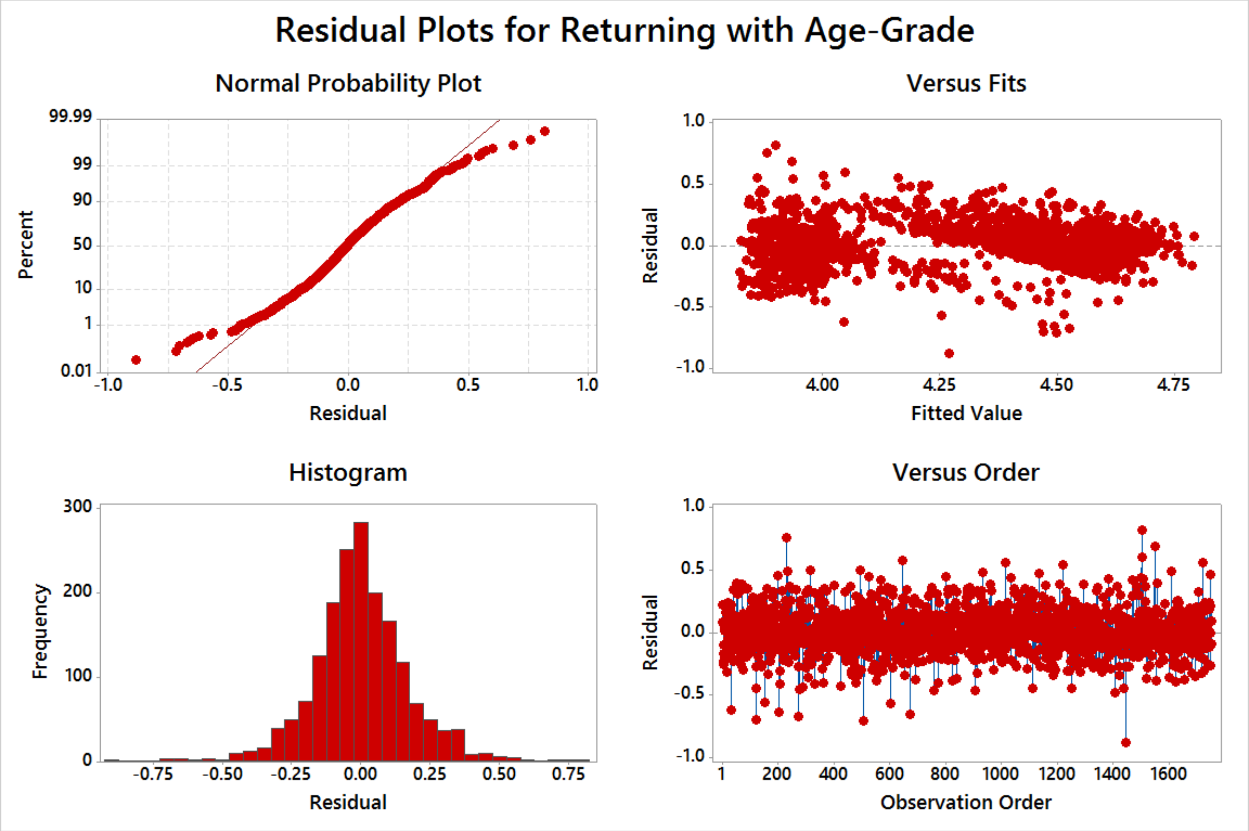


Figure 4.14 Residual Plots for Returning with Age-Grade

From Figure 4.13 and Figure 4.14 it can be observed that the normality assumption on both regressions is validated. The assumptions independence and distribution of the errors can also be validated with the plot of versus fits, this plot shows that the errors are evenly distributed above and below zero for each regression. Note that like in Section 4.5 four equations will result from this method.

$$\begin{aligned} New\ Female\ Participants \\ &= 90.15 - 26.51\,Ec - 15.63\,Ec^2 + 39.22\,Ec^3 + 1.78\,Ac \\ &- 2.05\,Ac^2 + 3.388\,PAc + 13.07\,PAc^2 \end{aligned} \tag{7}$$

$$\begin{aligned} New\ Male\ Participants \\ &= 88.17 - 26.51\,Ec - 15.63\,Ec\hat{}2 + 39.22\,Ec\hat{}3 + 1.78\,Ac \\ &- 2.05\,Ac\hat{}2 + 3.388\,PAc + 13.07\,PAc\hat{}2 \end{aligned} \tag{8}$$

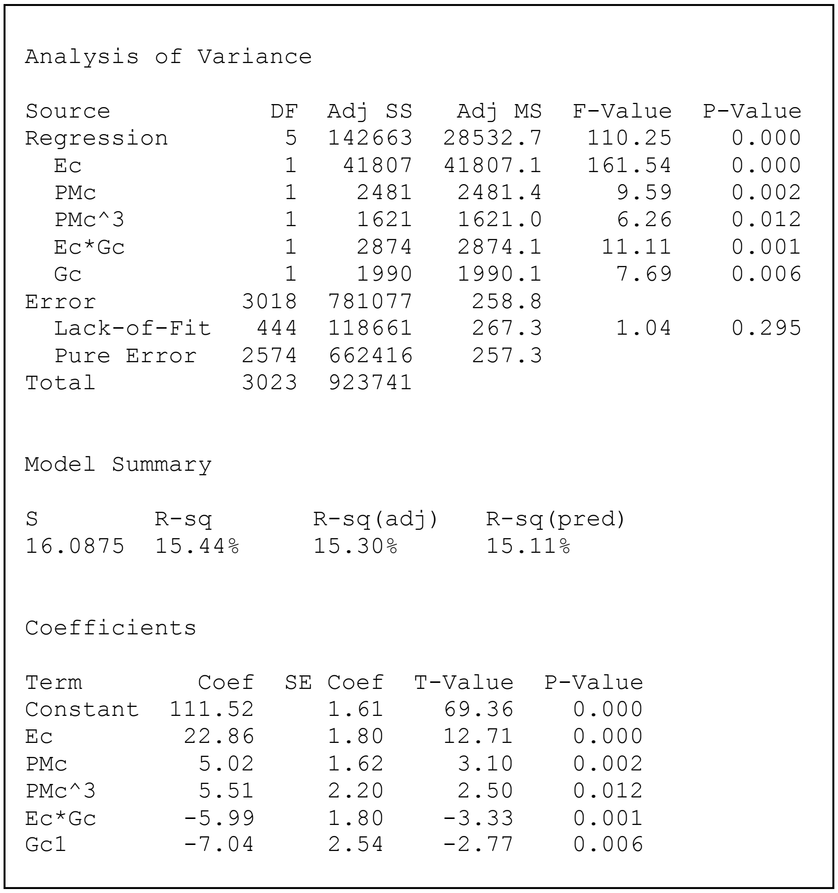
$$ln(Returning\ Female\ Participants) = 4.3552 - 0.7780\, Ec + 0.9369\, Ec^3 + 0.0934\, Ave\,Tc^2 - 0.3055\, Ave\,Tc * Ec + 0.0371\, Ave\,Tc * Gc - 0.0994\, Ec * Ac \quad (9)$$

$$ln(Returning\ Male\ Participants) = 4.3128 - 0.7780\, Ec + 0.9369\, Ec^3 + 0.0934\, Ave\,Tc^2 - 0.3055\, Ave\,Tc * Ec + 0.0371\, Ave\,Tc * Gc - 0.0994\, Ec * Ac \quad (10)$$

Equations 7 and 8 present the formulas to estimate the finish time for both male and female new participants. On equations 9 and 10 the formulas calculate the natural logarithm of the estimated finish times for male and female returning participants. For returning participants the age-grade did not prove to be significant while for new participant it is significant. This suggests that the age-grade make up for the lack of historical data for new participants.

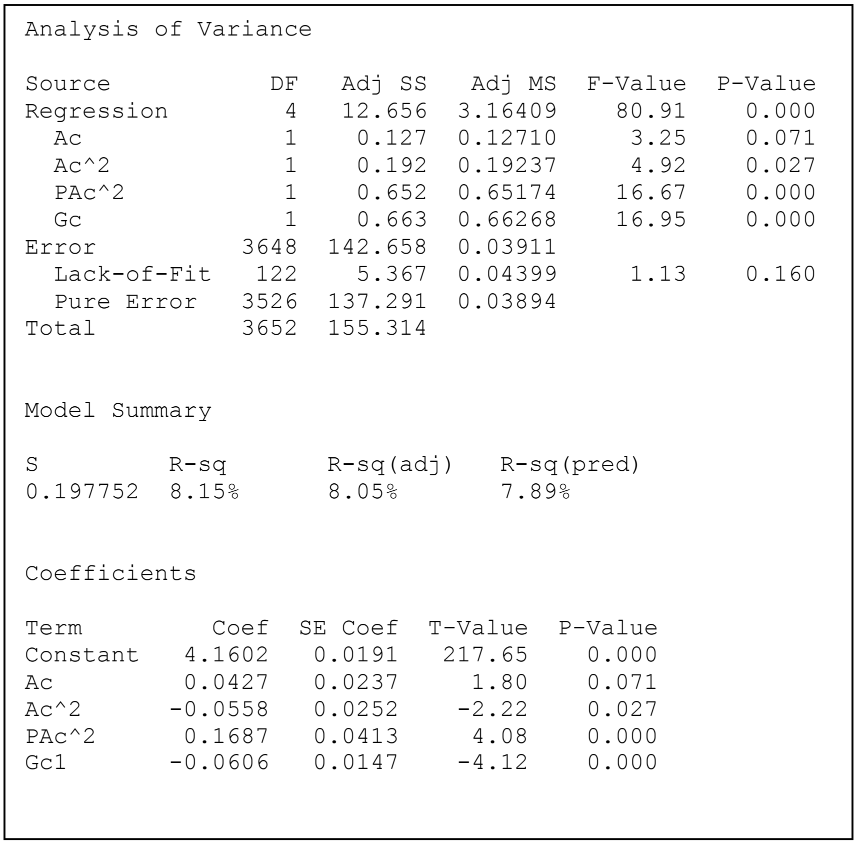
4.7 Method 4

The fourth method, Method 4, extends the third method by subdividing participants into two sub-groups: the ones with realistic estimates and non-realistic estimates. As a result this method is composed of four regressions; (1) new participants who provided good estimates (new realistic), (2) new participants who provided bad estimates (new non-realistic), (3) returning participants who provided good estimates (returning realistic), and (4) returning participants who provided bad estimates (returning non-realistic). This method is the equivalent to using categorical variables in the regression to handle non-realistic times. Figure 4.15 and Figure 4.16 presents the ANOVA and Model Summary for the new participants (new realistic and new non-realistic).



```
Analysis of Variance

Source           DF  Adj SS   Adj MS  F-Value  P-Value
Regression        5  142663  28532.7   110.25    0.000
  Ec              1   41807  41807.1   161.54    0.000
  PMc             1    2481   2481.4     9.59    0.002
  PMc^3           1    1621   1621.0     6.26    0.012
  Ec*Gc           1    2874   2874.1    11.11    0.001
  Gc              1    1990   1990.1     7.69    0.006
Error          3018  781077    258.8
  Lack-of-Fit   444  118661    267.3     1.04    0.295
  Pure Error   2574  662416    257.3
Total          3023  923741


Model Summary

S        R-sq       R-sq(adj)   R-sq(pred)
16.0875  15.44%     15.30%      15.11%


Coefficients

Term        Coef  SE Coef  T-Value  P-Value
Constant  111.52     1.61    69.36    0.000
Ec         22.86     1.80    12.71    0.000
PMc         5.02     1.62     3.10    0.002
PMc^3       5.51     2.20     2.50    0.012
Ec*Gc      -5.99     1.80    -3.33    0.001
Gc1        -7.04     2.54    -2.77    0.006
```

Figure 4.15 ANOVA and Model Summary for New Realistic



```
Analysis of Variance

Source           DF   Adj SS   Adj MS  F-Value  P-Value
Regression        4   12.656  3.16409    80.91    0.000
  Ac              1    0.127  0.12710     3.25    0.071
  Ac^2            1    0.192  0.19237     4.92    0.027
  PAc^2           1    0.652  0.65174    16.67    0.000
  Gc              1    0.663  0.66268    16.95    0.000
Error          3648  142.658  0.03911
  Lack-of-Fit   122    5.367  0.04399     1.13    0.160
  Pure Error   3526  137.291  0.03894
Total          3652  155.314


Model Summary

S          R-sq       R-sq(adj)   R-sq(pred)
0.197752  8.15%      8.05%       7.89%


Coefficients

Term         Coef  SE Coef  T-Value  P-Value
Constant   4.1602   0.0191   217.65    0.000
Ac         0.0427   0.0237     1.80    0.071
Ac^2      -0.0558   0.0252    -2.22    0.027
PAc^2      0.1687   0.0413     4.08    0.000
Gc1       -0.0606   0.0147    -4.12    0.000
```

Figure 4.16 ANOVA and Model Summary for New Non-Realistic

The regressions for the new participants yield an adjusted R-Squared of 15.30% for the ones with good estimates and 8.05% for the new participants with bad estimates. The p-values of the coefficients and predictors for the new participants with bad estimates are all below 0.10 indicating a significance with a confidence interval greater than 90%. On the other hand, the p-values of coefficients and predictors for the new participants with good estimates show significance with a confidence level greater than 83%. In Figure 4.17 and Figure 4.18 the residual plots for both regressions are presented.

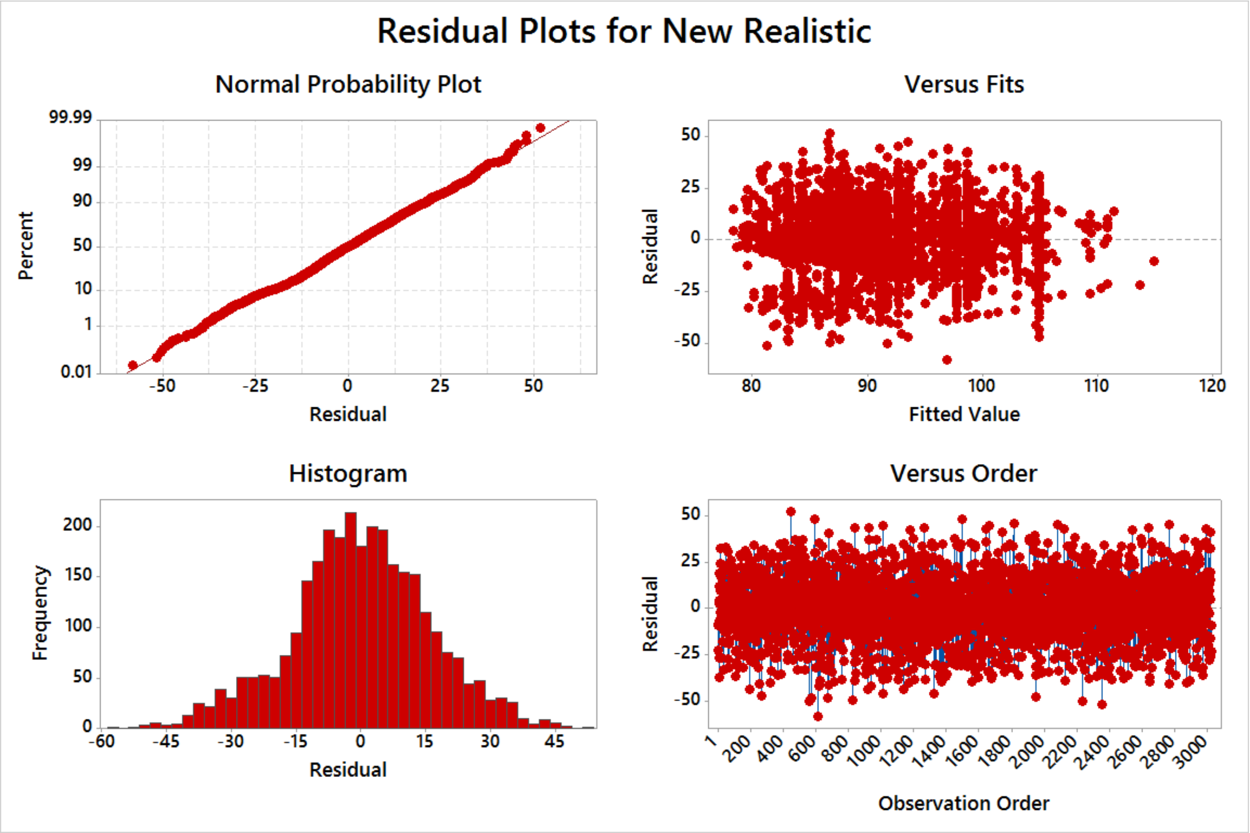


Figure 4.17 Residual Plots for New Realistic

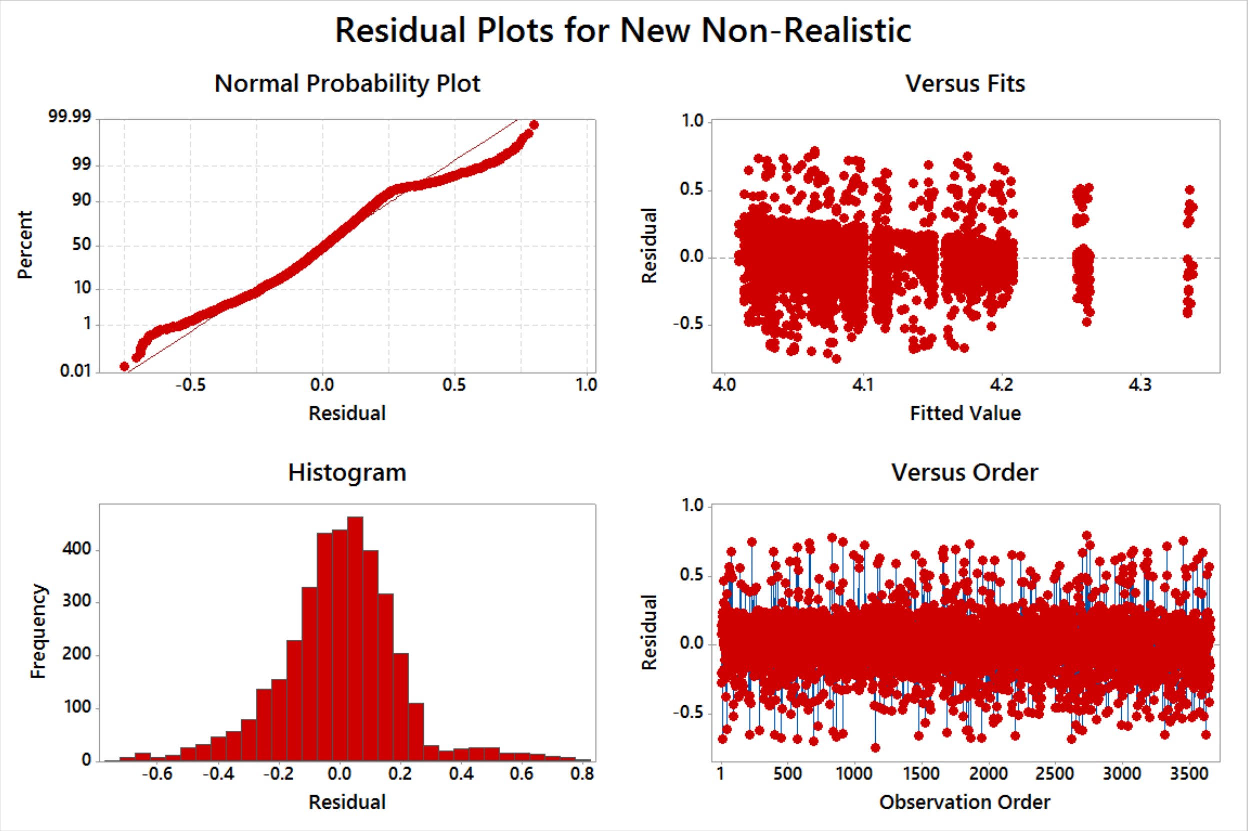
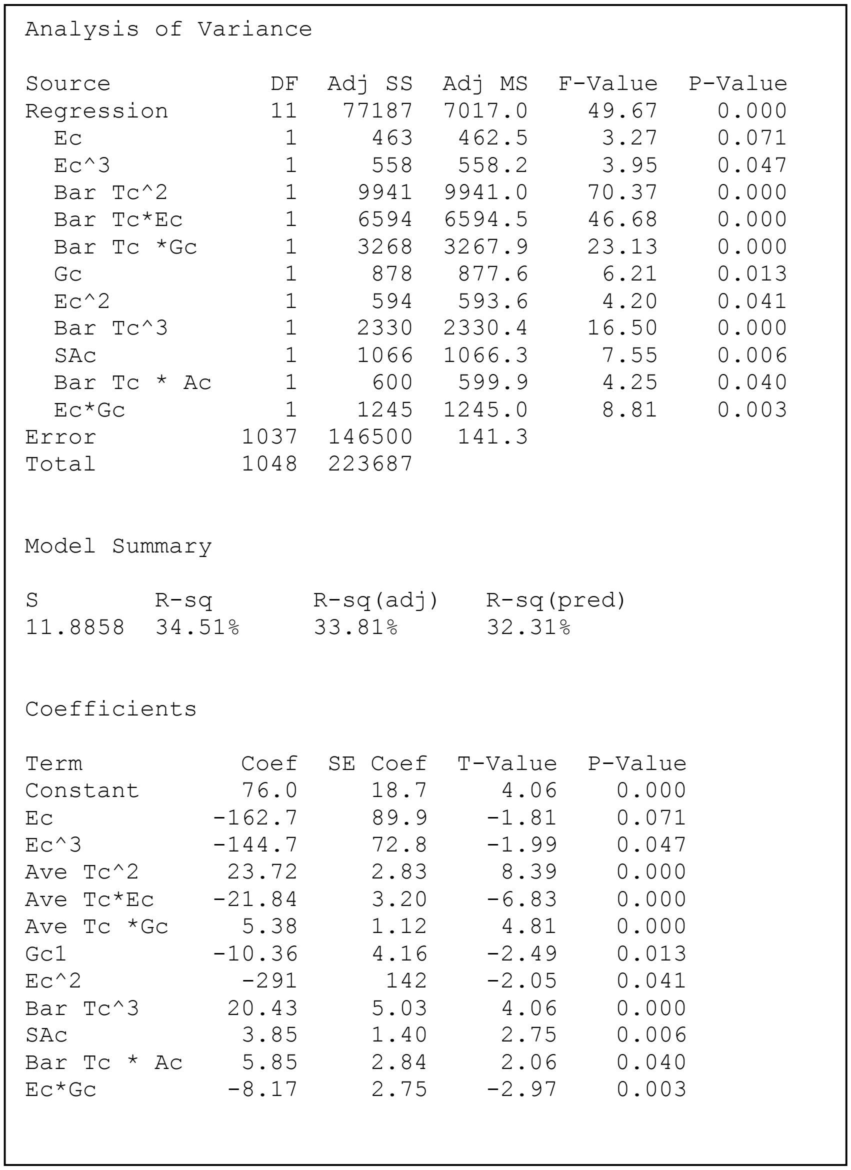


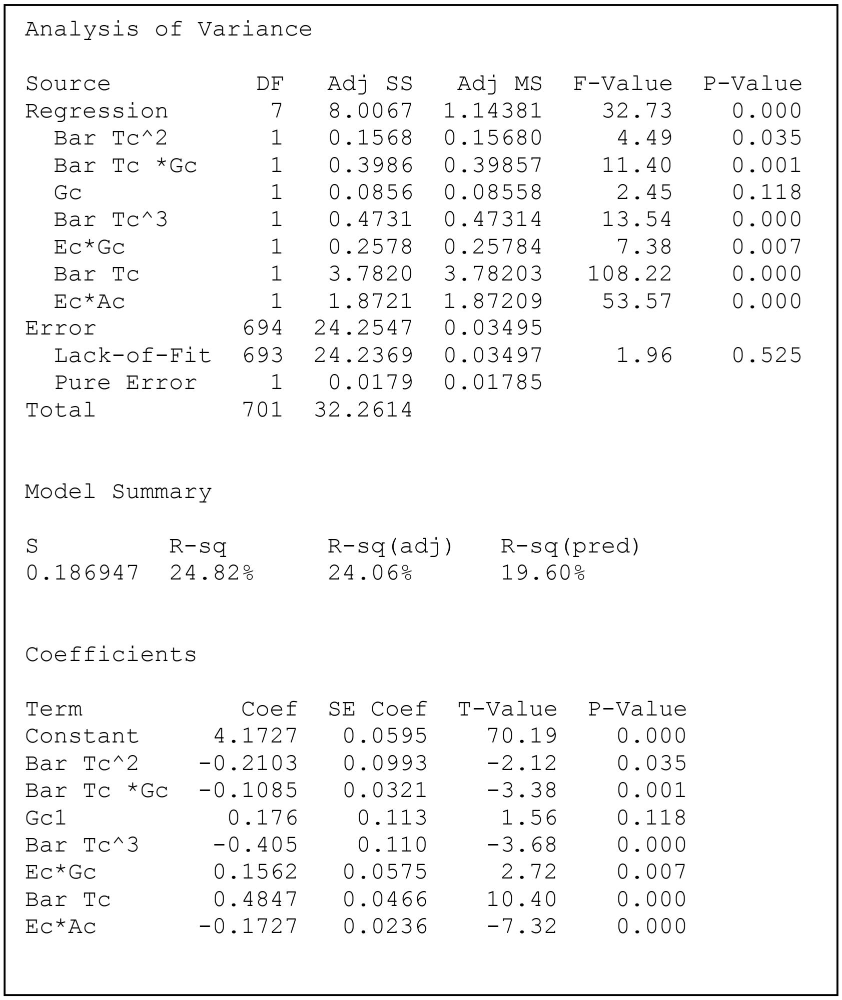
Figure 4.18 Residual Plots for New Non-Realistic

The normality assumption on both regressions are validated with the use of the normal probability plots and the histogram. The assumptions independence and distribution of the errors can also be validated with the versus fits, this plot shows that the errors are evenly distributed above and below zero for each regression. In Figure 4.19 and Figure 4.20 the ANOVA and Model Summary for the two sub-groups of the returning participants.



```
Analysis of Variance

Source           DF  Adj SS  Adj MS  F-Value  P-Value
Regression       11   77187  7017.0    49.67    0.000
  Ec              1     463   462.5     3.27    0.071
  Ec^3            1     558   558.2     3.95    0.047
  Bar Tc^2        1    9941  9941.0    70.37    0.000
  Bar Tc*Ec       1    6594  6594.5    46.68    0.000
  Bar Tc *Gc      1    3268  3267.9    23.13    0.000
  Gc              1     878   877.6     6.21    0.013
  Ec^2            1     594   593.6     4.20    0.041
  Bar Tc^3        1    2330  2330.4    16.50    0.000
  SAc             1    1066  1066.3     7.55    0.006
  Bar Tc * Ac     1     600   599.9     4.25    0.040
  Ec*Gc           1    1245  1245.0     8.81    0.003
Error          1037  146500   141.3
Total          1048  223687


Model Summary

S        R-sq       R-sq(adj)   R-sq(pred)
11.8858  34.51%     33.81%      32.31%


Coefficients

Term           Coef  SE Coef  T-Value  P-Value
Constant       76.0     18.7     4.06    0.000
Ec           -162.7     89.9    -1.81    0.071
Ec^3         -144.7     72.8    -1.99    0.047
Ave Tc^2      23.72     2.83     8.39    0.000
Ave Tc*Ec    -21.84     3.20    -6.83    0.000
Ave Tc *Gc     5.38     1.12     4.81    0.000
Gc1          -10.36     4.16    -2.49    0.013
Ec^2           -291      142    -2.05    0.041
Bar Tc^3      20.43     5.03     4.06    0.000
SAc            3.85     1.40     2.75    0.006
Bar Tc * Ac    5.85     2.84     2.06    0.040
Ec*Gc         -8.17     2.75    -2.97    0.003
```

Figure 4.19 ANOVA and Model Summary for Returning Realistic



```
Analysis of Variance

Source          DF   Adj SS   Adj MS  F-Value  P-Value
Regression       7   8.0067  1.14381    32.73    0.000
  Bar Tc^2       1   0.1568  0.15680     4.49    0.035
  Bar Tc *Gc     1   0.3986  0.39857    11.40    0.001
  Gc             1   0.0856  0.08558     2.45    0.118
  Bar Tc^3       1   0.4731  0.47314    13.54    0.000
  Ec*Gc          1   0.2578  0.25784     7.38    0.007
  Bar Tc         1   3.7820  3.78203   108.22    0.000
  Ec*Ac          1   1.8721  1.87209    53.57    0.000
Error          694  24.2547  0.03495
  Lack-of-Fit  693  24.2369  0.03497     1.96    0.525
  Pure Error     1   0.0179  0.01785
Total          701  32.2614


Model Summary

S          R-sq       R-sq(adj)   R-sq(pred)
0.186947  24.82%     24.06%      19.60%


Coefficients

Term           Coef  SE Coef  T-Value  P-Value
Constant     4.1727   0.0595    70.19    0.000
Bar Tc^2    -0.2103   0.0993    -2.12    0.035
Bar Tc *Gc  -0.1085   0.0321    -3.38    0.001
Gc1           0.176    0.113     1.56    0.118
Bar Tc^3     -0.405    0.110    -3.68    0.000
Ec*Gc        0.1562   0.0575     2.72    0.007
Bar Tc       0.4847   0.0466    10.40    0.000
Ec*Ac       -0.1727   0.0236    -7.32    0.000
```

Figure 4.20 ANOVA and Model Summary for Returning Non-Realistic

The models for the returning participants yield p-values below 0.07 for predictors and their coefficients indicating a significance with a confidence interval greater than 93%. The adjusted R-squared at a 33.81% for returning realistic participants and at a 24.06% for the returning participants with unrealistic estimates. Figure 4.21 and Figure 4.22 presents the residual plots for returning realistic and returning non-realistic.

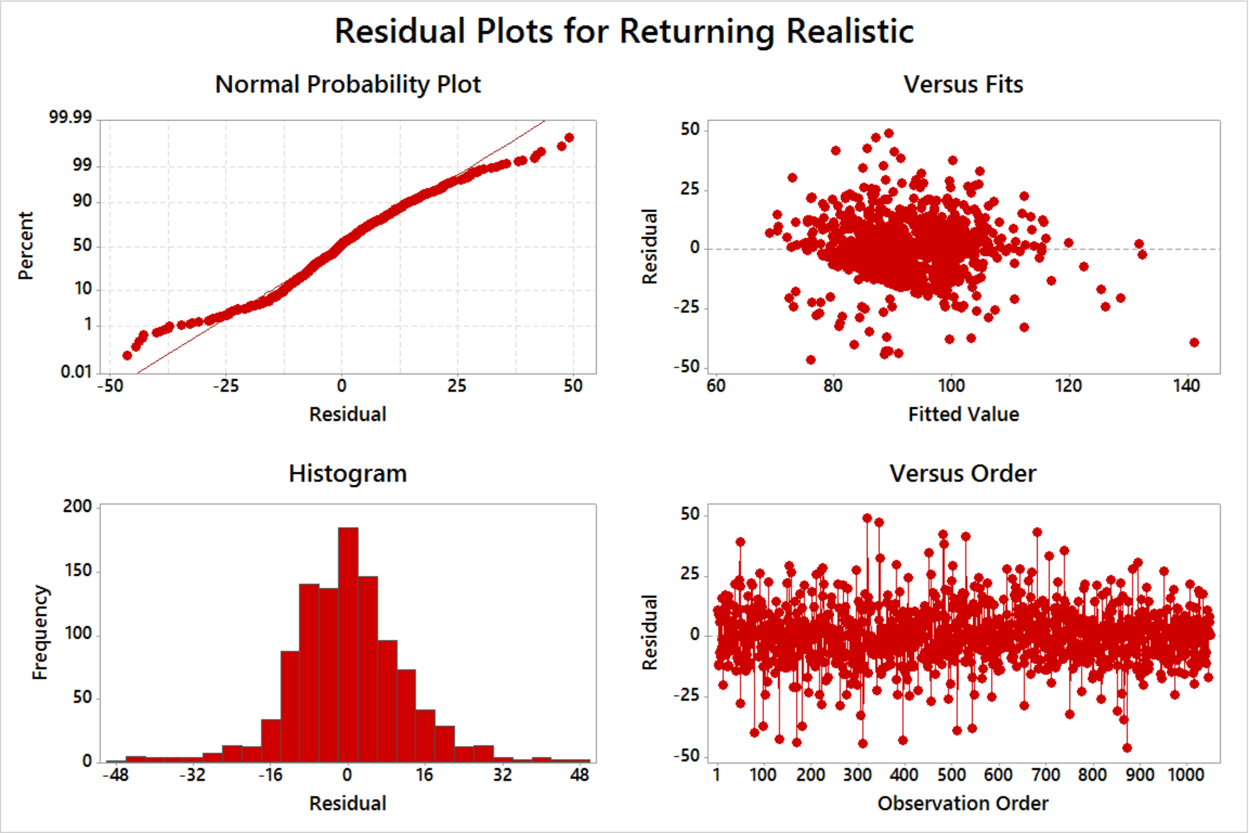


Figure 4.21 Residual Plots for Returning Realistic

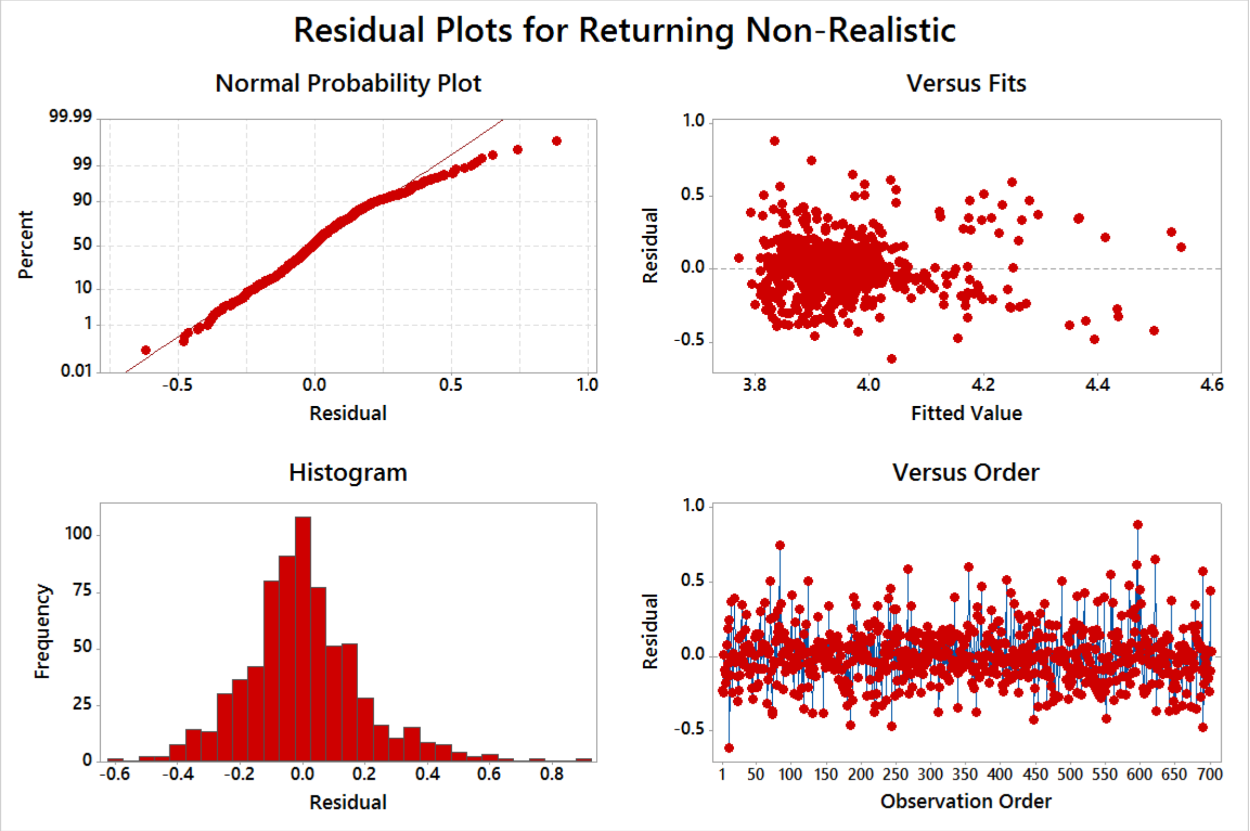


Figure 4.22 Residual Plots for Returning Non-Realistic

With the normal probability plots and histogram we can observe that the normality assumption of the errors is valid for the two regressions. The versus fits plots suggest a mean of zero for the errors, with an even distribution of the above and below zero. From this method eight formulas are obtain and are presented in equation 12 to 19.

$$New\ Realistic\ Females = 111.52 + 22.86\ Ec + 5.02\ PMc + 5.51\ PMc^3 - 5.99\ Ec * Gc \quad (12)$$

$$New\ Realistic\ Males = 104.48 + 22.86\ Ec + 5.02\ PMc + 5.51\ PMc\hat{}3 - 5.99\ Ec * Gc \quad (13)$$

$$ln(New\ Unrealistic\ Females) = 4.1602 + 0.0427\ Ac - 0.0558\ Ac^2 + 0.1687\ PAc^2 \quad (14)$$

$$ln(New\ Unrealistic\ Males) = 4.09961 + 0.0427\ Ac - 0.0558\ Ac^2 + 0.1687\ PAc^2 \quad (15)$$

$$\begin{aligned} &Returning\ Realistic\ Females \\ &\quad = 76.0 - 162.7\,Ec - 144.7\,Ec^3 + 23.72\,Ave\,Tc^2 \\ &\quad - 21.84\,Ave\,Tc * Ec + 5.38\,Ave\,Tc * Gc - 291\,Ec^2 \\ &\quad + 20.43\,Ave\,Tc^3 + 3.85\,SAc + 5.85\,Ave\,Tc * Ac - 8.17\,Ec \\ &\quad * Gc \end{aligned} \tag{16}$$

$$\begin{aligned} &Returning\ Realistic\ Males \\ &\quad = 65.7 - 162.7\,Ec - 144.7\,Ec^3 + 23.72\,Ave\,Tc^2 \\ &\quad - 21.84\,Ave\,Tc * Ec + 5.38\,Ave\,Tc * Gc - 291\,Ec^2 \\ &\quad + 20.43\,Ave\,Tc^3 + 3.85\,SAc + 5.85\,Ave\,Tc * Ac - 8.17\,Ec \\ &\quad * Gc \end{aligned} \tag{17}$$

$$\begin{aligned} &Ln(Returning\ Unrealist\ Females) \\ &\quad = 4.1727 - 0.2103\,Bar\,Tc^2 - 0.1085\,Ave\,Tc * Gc \\ &\quad - 0.405\,Ave\,Tc^3 + 0.1562\,Ec * Gc + 0.4847\,Ave\,Tc \\ &\quad - 0.1727\,Ec * Ac \end{aligned} \tag{18}$$

$$\begin{aligned} &Ln(Returning\ Unrealistic\ Males) \\ &\quad = 4.3492 - 0.2103\,Ave\,Tc^2 - 0.1085\,Ave\,Tc * Gc \\ &\quad - 0.405\,Ave\,Tc^3 + 0.1562\,Ec * Gc + 0.4847\,Ave\,Tc \\ &\quad - 0.1727\,Ec * Ac \end{aligned} \tag{19}$$

Note that all equations for unrealistic estimates, equations 14, 15, 18, and 19, needed to be transform to a natural logarithm in order to obtain a more reliable estimate from the regressions. Meanwhile the formulas for the realistic participants yield the estimated finish time. It can also be appreciated that for the new participants the age-grade proved to be significant in combination with the realistic estimate provided. For all unrealistic estimates the estimates did not prove to be significant and for this reason they were not included in the formulas.

4.8 Regression Methods

Figure 4.2 summarizes the different predictors considered for the regressions and identifies which one resulted statistically significant in each regression, correspondent to one of the methods by providing the sign of the Coefficient (C) as a positive (+) or a negative (-), and their p-value (p) – indicating the statistical significance of the terms. In addition Table 4.2

provides the adjusted R-square, the coefficient of determination, which gives in terms of percentage a metric of how well the regression line fits the finish times of 2014.

Table 4.2 Summary of Regression Models

	Method 1				Method 2				Method 3				Method 4							
													Returning				New			
	Return.		New		Return.		New		Return.		New		Realistic		Non-Realistic		Realistic		Non-Realistic	
Adjusted R-Squared	72.46%		55.47%		72.65%		55.81%		73.32%		56.79%		33.81%		24.06%		15.30%		8.05%	
	C	***p***	***C***	***p***	***C***	***p***	***C***	***p***	***C***	***p***	***C***	***p***	***C***	***p***	***C***	***p***	***C***	***p***	***C***	***p***
Estimate	-	0.047			-	0.000			-	0.000	-	0.000	-	0.071			+	0.000		
Estimate^2	-	0.580	-				-	0.041			-	0.000								
Estimate^3	+	0.000	+	0.061	+	0.000	+	0.000	+	0.000	+	0.010	-	0.047						
Age			+	0.000			+	0.000			+	0.092							+	0.071
Age^2											-	0.000							-	0.027
Age^3			-	0.000			-	0.000												
Gender					+	0.000	+	0.000	+	0.000	+	0.006	+	0.013	+	0.018	+	0.006	+	0.000
Age-grade Median																	+	0.002		
Age-grade Median^2																				
Age-grade Median^3																	+	0.012		
Age-grade Average											+	0.000	+	0.006						
Age-grade Average^2											+	0.000							+	0.000
Age-grade Average^3																				
Avg. Historical Time	+	0.000			+	0.000														
Avg. Historical Time^2	+	0.000			+	0.000			+	0.001			+	0.000	-	0.035				
Avg. Historical Time^3													+	0.000	-	0.071				
Gender and Avg. Previous Time									+	0.002			+	0.000	-	0.001				
Estimate and Avg. Previous Time									-	0.000										
Avg. Previous Time and Age	+	0.001											+	0.040						
Gender and Age	+	0.053	-	0.000			+	0.000												
Estimate and Age	-	0.000			-	0.000			-	0.000					-	0.000				
Estimate and Gender			-	0.00			-	0.002					-	0.003	+	0.000	-	0.001		
Age-grade Avg. and Estimate																				
Age-grade Avg. and Age																				
Age-grade Median and Estimate																				
Age-grade Median and Age																				
Age-grade Median and Avg.																				

From Table 4.2 it may be appreciated that the regression models for the returning participants have higher adjusted R-Squares than those of the new participants, which is expected given the additional predictor for the returning participants (*i.e.*, average previous time). Evaluating the R-Squares of the regressions it may be noted that the Method 1 has a better adjusted R-Squared than Methods 2 and 3 in both new and returning participants. Note that Method 4 cannot be directly compared with the other methods using the R-Squares due to the division of the data into four groups instead of two. Also from Table 4.2 it may be noted that when the average historical time is incorporated into the regressions the predictors yields p-values below 0.010 indicating that it is statistically significant with a confidence level of 99%.

4.9 Statistical Comparison

To determine the best method to estimate the participants' finish time we used the proposed regression models to estimate finish times for the 2014 edition of the WB10K. Figure 4.23 depicts how the WB10K participants were divided for the evaluation of the methods and the amount of non-elite participant in each category for the 2014 edition of the race.

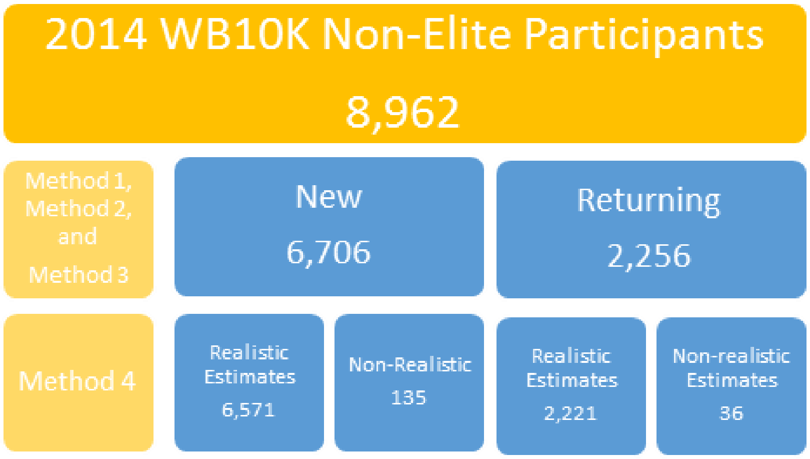


Figure 4.23 WB10K Classification for Non-Elite Participant of 2014

Without the 88 elite athletes that participated in 2014, the total participants in this evaluation are 8,962. The 2014 data was substituted into the four regression-based models to obtain finish the time estimates. The predicted estimate from each method is compared to the actual finish time registered for each participant at the 2014 edition of the WB10K. The Mean Squared Error (MSE) and the Mean Absolute Error (MAD) are used to evaluate the performance of each method. The respective formulas for MSE and MAD are presented in Eqns. (20) and (21) [22].

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(\widehat{Y_\iota} - Y_i\right)^2 \tag{20}$$

$$MAD = \frac{1}{n}\sum_{i=1}^{n}\left|\widehat{Y_\iota} - Y_i\right| \tag{21}$$

In Equation 1 and Equation 2, $\widehat{Y_\iota}$ represents the finish time for participant *i* in 2014, whereas Y_i represents the predicted finish time for participant *i* (resulting from the models). Both formulas quantify the error between the prediction and the time registered at the race (*i.e.*, $\widehat{Y_\iota} - Y_i$). The main difference between the two metrics is that the MAD indicates by how many units does the predictor deviates from the actual time on average. On the other hand, the MSE gives us the square of the average error. Squaring the error ensures that all errors are non-negative and strongly penalizes large estimation errors. Table 4.3 shows the MSE and MAD values of the methods.

Table 4.3 MSE and MAD for Each Method – 2014 data

	MSE	MAD
Method 1	559	21
Method 2	457	17
Method 3	1,404	29
Method 4	7,409,523	97
Status Quo	3,202	46

From Table 4.3 it may be concluded that the method with the least MSE and MAD is Method 2. It is interesting to note that the estimates provided by the participants are a better predictor than the Method 4. (Note that the fourth method is composed of four regressions rather than two, like Methods 1, 2 and 3.) On the other hand, if we compare each method's MSE and MAD for new and returning participants the results vary slightly. The comparison for new and returning participants is presented in Table 4.4 with the lowest MSEs are highlighted with a bold font.

Table 4.4 Method's MSE and MAD for New Participants vs. Returning Participants (2014)

	Method 1		Method 2		Method 3		Method 4		Status Quo	
	MSE	MAD	MSE	MAD	MSE	MAD	MSE	MAD	MSE	MAD
Returning	475	19	**344**	15	$1.6\text{x}10^4$	16	3,269	16	681	16
Returning Realistic Estimates	475	19	303	15	**295**	14	393	9	437	15
Returning Non-realistic Estimates	**520**	18	2,529	24	$1.1\text{x}10^6$	209	1,089	19	$1.4\text{x}10^4$	92
New	587	21	**499**	18	521	19	$9.9\text{x}10^6$	54	946	18
New Realistic Estimates	587	21	**444**	18	474	19	1,806	20	1,686	28
New Non-realistic Estimates	**595**	20	3,553	29	3,163	24	1,133	17	$1.9\ \text{x}10^4$	113

In Table 4.4 the different groups among the non-elite participants are labeled on the right. Non-elite participants are divided in to two main groups Returning and New, these two categories are sub-divided in to realistic or non-realistic (*i.e.*, based on the estimates provided).In each row the MSE and the MAD for the four method is presented. Table 4.4 illustrates the performance of each regression method with the participant's category or sub-category. When comparing the MSE and MAD for the four participants' categories (*i.e.*, sub-groups), Method 1 yields the best MSE for two of the four sub-groups, new and returning participants with non-realistic estimates. On the other hand, Method 2 yields the best alternative for new participants with realistic estimates Method 3 for returning participants with realistic estimates. These results suggest that even though the values of the estimates provided by participants are not realistic, they contribute to predict the finish time of the participants who provide them. If Methods 1, 2, and 3 are combined by selecting the best performing method in each sub-group the MSE and MAD (errors) for the predictions would improve. Note that the combined method, henceforth referred to as the Combined Method, would predict finish times of returning participants with realistic estimates using Method 3, the finish time of new participants with realistic estimates with Method 2, and the finish times of participants with non-realistic estimates with Method 1. Table 4.2 presents the MSE and MAD for the Method 2, who resulted in the best MSE overall in Table 4.5, and the combined Method.

Table 4.5 Method 2 vs. Combined Method

	MSE	MAD
Method 2	457	17
Combined Method	129	3

The MSE and MAD improvement from the Combined Method over Method 2 is 28.23% and 17.65%, respectively. Note that the regressions for the Combined Method segregate the participants who provide non-realistic estimates and incorporates them in Method 1, giving them the opportunity to assume values from -1 to 1. Meanwhile, using Method 2 for new participants provides the opportunity rigorously discriminate between the estimates and assigning more weight to the realistic estimates. Method 3 resulting in the best alternative for returning participants with realistic estimates suggests that the age-grade is a better predictor for returning than for new participants. When compared to Status Quo, the Combined Method would have an improvement of MSE and MAD for the 2014 edition of 700.66% and 270.59%, respectively.

5. Interferences Lower Bound

In this chapter a mathematical model is proposed to identify a lower bound for the total interferences during the WB10K.

5.1 *Identifying the Lower Bound*

In order to identify the minimum possible number of interferences during past WB10K editions, a mathematical model was developed. The resulting number of interferences from the mathematical model is a lower bound for the number of interferences. Given the times registered by participants at the checkpoints, and the time in which participants crossed the start line, the mathematical model will assign participants to starting positions to determine the lower bound.

The mathematical model assigns starting positions (represented by the guntimes at the 0K checkpoint) to participants (whose historical net time is registered) so that the total number of interferences during the race (from the 0K to the 10K) is minimized. The lower bound will correspond to a corral policy that has one participant per corral, which is impractical for the WB10K. However, the lower bound provides a benchmark for assessing the corral designs and starting policies. In summary, for the model, the historical time it took participants to reach the start line will be sorted and assigned to the different starting positions, where the smallest value will correspond to the first position and the largest to the last position. Once the participant is assigned to a starting position the time it takes to get to the start line will be added to the net time it took the participant to complete the intervals (0K to 3K, 3K to 8K, and 8K to 10K). After the starting positions are assigned the model will calculate the number of interferences based on that assignment. The model will evaluate various starting positions for all the participants until it can find a global minimum.

5.2 *Notation*

The notation used in the mathematical model is presented below, starting with the indexes used for the parameters and variables.

Indexes:

k: $checkpoints$ ($k = 1$ for 0K, $k = 2$ for 3K, $k = 3$ for 8K, $k = 4$ for 10K)

p: $starting\ positions\ p \in (1,2,, \dots, h)$

i, j: $participants\ \ \in (1,2, \dots, l)$

Note that the indexes presented above are used to identify the particular checkpoint, starting position, and individual. For example the checkpoints are identified as k=1 when evaluating the 0K checkpoint. On the other hand, the starting positions and the participants are identified from numbers 1 to the total number of starting positions (h) and participants (l). The following are the parameters of the model.

Parameters:

$g_p = guntime\ of\ position\ p\ to\ checkpoint\ 0K$

$n_{ik} = net\ time\ of\ runner\ i\ at\ checkpoint\ k$

The parameters have values that are considered as input to the model and do not change value as the constraints are evaluated. The gun times are taken from the historical time that participants took to arrive at the 0K from a given position (p). Meanwhile, the net times correspond to the time each participant (*i,j)* takes to arrive to a checkpoint (k). The following are the variables included in the mathematical model.

Variables:

$$s_{ip} = \begin{cases} 1, & if\ runner\ i\ is\ assigned\ to\ starting\ position\ p \\ 0, & otherwise \end{cases}$$

$$w_{ijk} = \begin{cases} 1, & if\ participants\ i\ reaches\ checkpoint\ k\ before\ runner\ j \\ 0, & otherwise \end{cases}$$

$$x_{ijk} = \begin{cases} 1, & if\ runner\ i\ passed\ participants\ j\ between\ chekpoints\ k\ and\ k+1 \\ 0, & otherwise \end{cases}$$

The variables used in the mathematical model are binary (*i.e.* can only take values of 0 or 1). The variable s_{ip} indicates which runner (i,j) is assign to a certain position (p). On the other hand, the variable w_{ijk} takes a value of one (1) if participant i arrives to checkpoint k before participant j. Lastly, the variable x_{ijk} is equal to one (1) when the the participant i interfered with the participant j in the interval demarked by checkpoint k and k+*1*.

5.3 *Problem Formulation*

The objective function of the model is to minimize the interferences between the participants of the WB10K in an ideal scenario, as presented in Equation 22.

$$Min\ z = \sum_{i=1}^{l} \sum_{j=1}^{l} \sum_{k=1}^{4-1} x_{ijk} \tag{22}$$

Note that Equation 22 minimizes the sum of interferences (x_{ijk}) between all participants (i,j) at the intervals denoted by the checkpoints (k). This objective function is subject to eight sets of constraints. The first two constraint sets are presented in Eqs. 23 and 24.

$$\sum_{i=1}^{l} S_{ip} = 1 \quad \forall\ p = 1, 2, \dots, h \tag{23}$$
$$\sum_{p=1}^{h} S_{ip} = 1 \quad \forall\ i = 1, 2, \dots, l \tag{24}$$

Equation 23 ensures that there is only one participant per starting position. Equation 24 allows only one starting position per participant. By including these constraints the model is making sure that no participant is on top of another, something that would not be feasible in reality. In order to verify the arrival sequence at the checkpoints two constraint sets were added, Equations 25 and 26.

$$\left(\sum_{p=1}^{h}(g_p * \mathrm{s}_{jp}) + n_{jk}\right)$$

$$-\left(\sum_{p'=1}^{h}(g_{p'} * \mathrm{s}_{ip'}) + n_{ik}\right) \leq M * w_{ijk} \quad \forall\, j, i \neq j \ \in 1, .., l, k = 1, \dots, 4 \tag{25}$$

$$w_{ijk} + w_{jik} = 1 \quad \forall\, j, i \neq j \ \ k = 1, \dots, 4 \tag{26}$$

The combination of Equation 25 and 26 registers which participant got to the checkpoint earlier. Equation 25 takes into account the time it takes from the corral to the start line. Note that if participant *i* reaches the checkpoint *k* before participant *j* then $\mathrm{w}_{\mathrm{ijk}}$ will take value of 1. Meanwhile, if the opposite happens, if participant *j* reaches the checkpoint *k* before participant *i* then $\mathrm{w}_{\mathrm{ijk}}$ will take value of 0. By multiplying the binary variable that identifies the starting position of a participant (s_{jp}) by the gun time parameter (g_p), the gun time that will be added to the net time (n_{jk}) of the participant will be the one corresponding to the position assigned. On the other hand, Equation 26 makes sure that one of them arrives first.

$$w_{ijk+1} - w_{ijk} \leq M\, x_{ijk} \quad \forall\, j, i \neq j, k = 1{,}2{,}3 \tag{27}$$

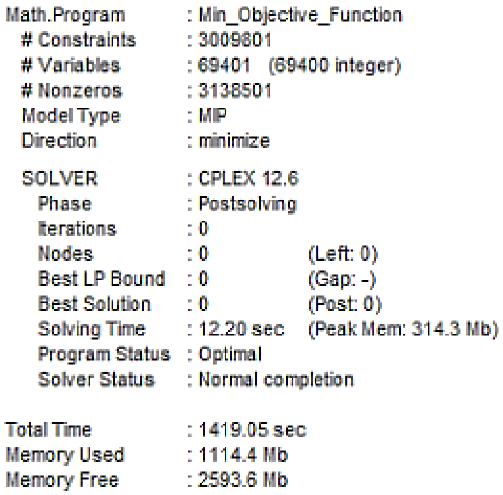
Equation 27 combined verifies if there was an interference between runners, with x_{ijk}. If participant *i* arrived before participant *j* at the checkpoint *k* and at the checkpoint *k+1* the participant *j* arrived first then x_{ijk} takes value of 1. If participant *i* arrives before participant *j* at both checkpoints then x_{ijk} takes valued of 0.

$$x_{ijk}, s_{ip}, w_{ijk} \in \{0,1\} \quad \forall\, j, i, \mathrm{k} \tag{28}$$

Equation 28 assures that the variables stay within their feasible region, declaring them as binary.

5.4 *Lower Bound for WB10K 2014 edition*

The lower bound model was programed in AIMMS version 4.0 64-bits. The model evaluates all starting positions for every participant to find the global minimum of interferences, which may result in insufficient memory resources when evaluating a large number of participants. To illustrate the model, a small internally generated instance with 100 participants was run in a MacBook Pro with 8Gb RAM and 2.5 GHz Intel Core i5 processor. Figure 5.1 shows the AIMMS result window.



```
Math.Program            : Min_Objective_Function
  # Constraints         : 3009801
  # Variables           : 69401   (69400 integer)
  # Nonzeros            : 3138501
  Model Type            : MIP
  Direction             : minimize

  SOLVER                : CPLEX 12.6
    Phase               : Postsolving
    Iterations          : 0
    Nodes               : 0             (Left: 0)
    Best LP Bound       : 0             (Gap: -)
    Best Solution       : 0             (Post: 0)
    Solving Time        : 12.20 sec     (Peak Mem: 314.3 Mb)
    Program Status      : Optimal
    Solver Status       : Normal completion

Total Time              : 1419.05 sec
Memory Used             : 1114.4 Mb
Memory Free             : 2593.6 Mb
```

Figure 5.1 AIMMS Result Window

From Fig. 5.1 it may be observed that the optimal solution is zero interference. Note that the model assumes an ideal scenario where only one participant is assigned per corral. While the current corral policy adds randomness to the actual starting position of the participant. This makes it hard to ensure that the fastest participant will be starting from a position nearest to the start line than the slower participants.

Also, Fig. 5.1 shows the number of variables that are generated, 69,401 integer variables. This model used 1114.4 Mb, which translates to approximately 1Gb, and took 1419.05

seconds, approximately 23 minutes. The memory resource usage is a value that will increase exponentially as participants are added. To evaluate the total number of participants during the WB10K 2014 edition the program would need to be installed in a computer with more than 100Gb or RAM.

Furthermore this mathematical model may be used to evaluate the effectiveness of the corral assignments by comparing the lower bound obtained with the total number of interferences during the corresponding WB10K edition. This tool could also aid in the corral assignment policy by using regressions to estimate the net times of the participants and running the model with those values. The results will indicate which position should be assign to each participant and this may be used to assign the fastest participants to the front corrals and so on. This model could also be adapted to other racing events and used as an evaluation tool.

6. Corral Design

6.1 Simulation

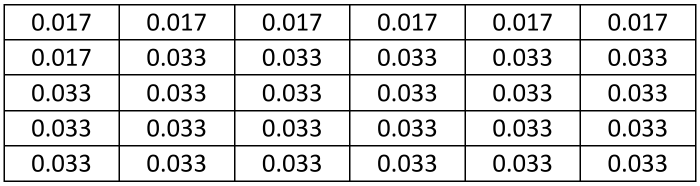
Determining the best corral design for the WB10K requires evaluating several combinations of quantity and size of corrals. Furthermore, additional policies regarding the release of corrals (e.g., releasing all corrals simultaneously or using waves) could also be contemplated for the WB10K. However, since the WB10K is a yearly event, and the logistics of implementing various policies for evaluation would be time consuming, real-life experimentation is impractical. Instead, a Monte Carlo simulation will be developed in Java to evaluate different corral designs and corral release policies with respect to the resulting interference between participants.

The Monte Carlo simulation takes as input the actual times, net times, estimate provided, and bib numbers from the participants of the WB10K. Additionally, the participants estimated times form the combined method presented on Section 4.9 may be also used as input for the simulation. On the other hand, the corral size, number of corrals, will be considered an instance of the different policies. The pseudo code for the simulation is presented in Figure 6.1.

```
Given:
$i$   = number of participants
$n$  = index for corrals
$x$  = width of corral
$y$  = height of corral
$m$ = index for replica
$w_n$ = index for Width of Corral N
$h_n$ = index for Height of Corral N

SORT position times in ascending order
SORT participants in ascending order based on participant’s estimated time
FOR $m$ = 1 to quantity of replicas
    SET $n$ = 1
    FOR $i$ = 1 to quantity of participants
        FOR $h_n$ = 1 to $y$ of Corral$_N$
            FOR $w_n$ = 1 to $x$ of Corral $n$
                IF  $h_n$ = Height of Corral $n$ NEXT $n$
                ASSIGN position time $i$ to location ($h_n$,$w_n$) of Corral $n$
                ASSIGN participant $i$ to location ($h_n$, $w_n$) Corral $n$
                ASSIGN participant $i$ net times to location ($h_n$, $w_n$) Corral $n$
            Next $w_n$
        Next $h_n$
    Next $i$

    FOR $n$ = 1 to quantity of corrals
        RANDOMIZE participants in corral $n$
    Next $n$
RUN FUNCTION calculate interferences as in Error! Reference source not found.
Next M
END
```

Figure 6.1 Pseudo Code for Monte Carlo Simulation

As shown in Figure 6.1 the code will sort the estimated times in order to identify which participants should be assigned to the front corrals and which should be assigned to the further ones. Meanwhile, the code will assign a position time (*i.e.,* the time it takes the participant to reach the starting line from the corresponding corral) to each starting position within the corrals. These starting positions are the registered times from the corral to the 0K during the race. Note that it is assumed that the smallest time is achieved by a participant departing from a corral nearest to the start line. Afterwards, participants will be assigned to a random position within the corral, these steps intend to recreate the fact that the precise start position within a corral is uncontrollable. The simulation represents the corrals as matrices, in a matrix the height and width dictate the number of elements it can hold (*i.e.*, in this case participants). Corral Matrix presents an example of a corral as it is considered in the simulation.



0.017	0.017	0.017	0.017	0.017	0.017
0.017	0.033	0.033	0.033	0.033	0.033
0.033	0.033	0.033	0.033	0.033	0.033
0.033	0.033	0.033	0.033	0.033	0.033
0.033	0.033	0.033	0.033	0.033	0.033

Figure 6.2Corral Matrix Position Times

As shown in Figure 6.2, the position times are assigned to a specific position within the corral. Note that this matrix is a 6×5 matrix, this indicates that 30 participants can be fitted inside this corral. Once everyone is assigned a position, a position time is added to their corresponding net time. With the data exported in an MS Excel File, the counter presented in Figure 3.2 will be executed to calculate the number of interferences produce by the particular corral policy.

6.2 *Validation*

In order to ensure that the model is a correct representation of the real system, the Monte Carlo simulation will be executed with the parameters that more closely represent the WB10K current corral policy. For this, the bib numbers assigned to the participants by the WB10K in the 2014 edition will serve to assign the 9,050 non-elite participants to the corrals. The logic behind it is that the simulation will treat the bib numbers as the estimated finish times, where the participant with the lowest bib numbers will be assigned to the corrals nearest to the starting line and the largest will be further from it. By using the bib numbers the current placement policy is evaluated by the simulation. Additionally to represent the corral size used during the WB10K (*i.e.*, as presented on Figure 3.1) the width of all corrals will be 10, this number is based on an observation of the participants departing from the corrals, and the length will be the number needed to achieve the real participant capacity per corral. The data from the 2014 edition will be used to evaluate the simulations. To validate the interferences between participants during the 2014 edition will be compared to the interferences yield by the simulation. Arbitrarily, ten replicas of the simulation will be ran in order to calculate the needed number of replicas. The interference results from the simulation are presented in Table 6.1.

Table 6.1 Number of Interferences Validation with BIB Numbers

	0K-3K	**3K-8K**	**8K-10K**	**Total**
Replica 1	32,849,532	3,178,473	2,369,317	38,397,322
Replica 2	32,903,910	3,176,276	2,377,766	38,457,952
Replica 3	32,852,103	3,173,423	2,373,947	38,399,473
Replica 4	32,805,018	3,181,314	2,375,933	38,362,265
Replica 5	32,870,297	3,172,315	2,376,784	38,419,396
Replica 6	32,890,298	3,172,908	2,378,453	38,441,659
Replica 7	32,891,299	3,183,685	2,374,687	38,449,671
Replica 8	32,873,199	3,182,399	2,371,159	38,426,757
Replica 9	32,788,071	3,171,322	2,377,834	38,337,227
Replica 10	32,866,287	3,177,689	2,375,638	38,419,614
Average	32,859,001	3,176,980	2,375,152	38,411,134
Std. Dev.	37,274.1	4,463.956	2,983.135	38,253.87

From Table 6.1 it can be observed that the total number of inferences in each replica surpasses thirty eight million. Comparing the results on Table 6.1 with Table 3.2, by observation, alone one may suspect that the model does not represent the real system. In order to correctly make conclusions, the number of replicas needed must be calculated. To determine the number of replicas the following equation is used.

$$n = \left(\frac{t_{\frac{\alpha}{2},n-1} * S}{e_r * \bar{x}} \right)^2 \tag{29}$$

The formula presented on Equation 29 is the common used formula to identify the sample size [23]. In this formula, *n* represents the sample size (number of replicas), *s* is the standard deviation of the sample, e_r is the relative error, $\bar{x}$ is the average of the sample and $t_{\alpha/2,\, n-1}$ is the two sided student-t distribution. To obtain the results presented in Table 6.1 an error of five percent from the average is used, as well as the average and standard deviation in Table 6.2.

Table 6.2 Sample Size for Bib Number Validation

	0K-3K	3K-8K	8K-10K	Total
Relative Error	1,642,950	158,849	118,757.6	1,920,557
t(0.95,10-1)	2.2621	2.2621	2.2621	2.2621
Sample Size	0.002634	0.004041	0.003229	0.00203

To achieve a 95% confidence interval with the desired error based on the initial sample of 10 simulation runs, the number of replicas needed is 0.004041. Since the number of replicas needed is less than the initial number of replicas generated (*i.e.*, 10 replicas), then those should be sufficient to calculate the confidence interval and verify if the model validates. Equation 30 will be used to calculate the confidence interval.

$$\bar{x} \pm t_{\frac{\alpha}{2},n-1} * \frac{s}{\sqrt{n}} \tag{30}$$

By adding $t_{\frac{\alpha}{2},n-1} * \frac{s}{\sqrt{n}}$ to the average, the upper limit is obtain and by subtracting $t_{\frac{\alpha}{2},n-1} * \frac{s}{\sqrt{n}}$ to the average the lower limit is calculated. The confidence interval for the bib number corral assignment simulation are presented on Table 6.3.

Table 6.3 Confidence Interval for Estimated Mean of Interferences

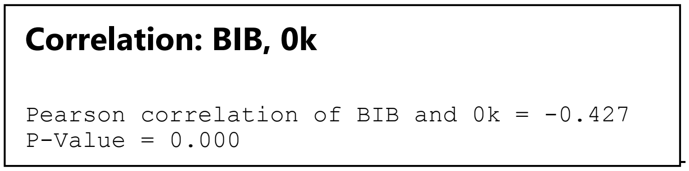
	0K-3K	**3K-8K**	**8K-10K**	**Total**
$\boldsymbol{t_{\frac{\alpha}{2},n-1} * \frac{s}{\sqrt{n}}}$	26,664.28	3,193.322	2,134.006	27,365.17
Upper Limit	32,885,666	3,180,174	2,377,286	38,438,499
Average	32,859,001	3,176,980	2,375,152	38,411,134
Lower Limit	32,832,337	3,173,787	2,373,018	38,383,768

The confidence intervals are used to compare the interferences during the WB10K 2014 with the simulation replicas. If the number of interferences yield with the real data falls within the confidence intervals it can be said that the simulation is an accurate representation of the system. Table 6.4 presents the number of interferences during the WB10K 2014 edition, as presented in Table 6.3.

Table 6.4 WB10K 2014 Interferences

0K-3K	**3K-8K**	**8K-10K**	**Total**
12,443,247	3,221,238	2,414,538	18,079,023

When comparing the interferences during the WB10K 2014 edition and the simulation, it can be noted that the results are not contained within the confidence intervals, this indicates that the simulation does not represent the system. In other words, the simulation results for the number of interference based on how the WB10K organization assigns participants to corrals (i.e., based on their estimated times) does not match actual number of interferences during the race. Note, however, that since there is no strict control policy in place during the WB10K to ensure that participants start from the assigned corral. Hence, in order to validate the 2014 edition results we should model how the participants ended up organizing themselves during the race. To test the hypothesis that participants do not adhere to their assigned corral, a Pearson Correlation test was conducted in Minitab 17 to verify the dependency between the bib number and the 0K registered time. Figure 6.3 presents the results.



Correlation: BIB, 0k

```
Pearson correlation of BIB and 0k = -0.427
P-Value = 0.000
```

Figure 6.3 Correlation Test for Bib Number and Position Time

From the correlation test on Figure 6.3 the p-value indicates that there is statistical evidence that there is correlation between the bib number and the time it takes a participant to reach the start line (0K). Hence, the test indicates that the relationship is negative (*i.e.,* as the bib number increases it takes less time to arrive to the starting line), of -0.427. In addition to the test, the marginal plot in Fig. 6.4 was obtained from Minitab 17 to better understand the relationship.

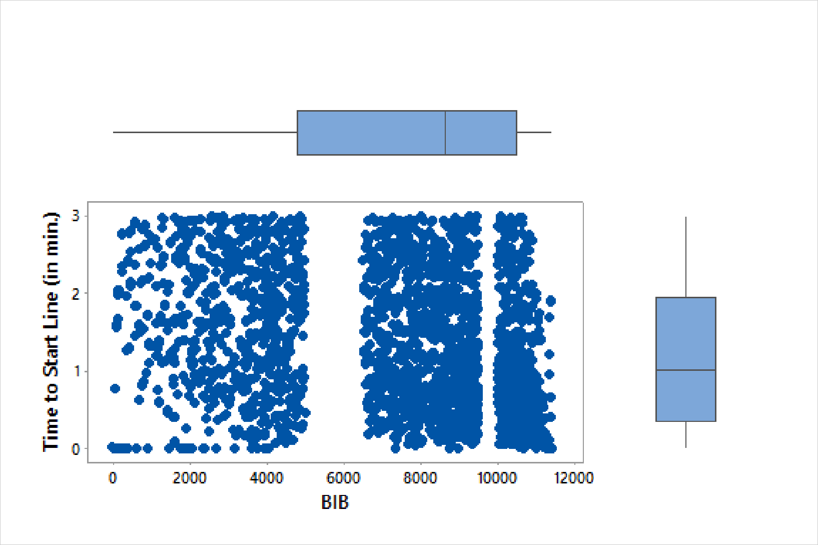


Figure 6.4 Marginal Plot of 0K vs. Bib Number

Figure 6.4 shows that there is no relationship between the bib number and the time it takes a participant to reach the start line. According to the current policy the participants with lower bib numbers should reach the start line in less time than participants with larger bib numbers. If that policy would be enforced there would be a positive correlation between the bib number and the time it takes a participant to reach the start line. This behavior may indicate that the participants do not follow the current corral assignment policy. Validating the model requires that the position assigned to the participant is within the same corral they started the 2014 race. In an effort to replicate this reality another simulation will be executed using the times the participants registered at the 0K as the corral indicator. In other words, participants with lower times at the 0K checkpoint will start from a corral closes to the start line. In Table 6.5 the results for ten replicas are presented.

Table 6.5 Validation with 0K Registered Times

	0K-3K	3K-8K	8K-10K	Total
Replica 1	12,584,272	3,211,787	2,412,949	18,209,008
Replica 2	12,398,583	3,211,640	2,440,365	18,050,588
Replica 3	12,308,411	3,215,815	2,439,968	17,964,194
Replica 4	12,359,598	3,288,993	2,413,233	18,061,824
Replica 5	12,547,957	3,212,594	2,413,133	18,173,684
Replica 6	12,364,877	3,223,483	2,441,489	18,029,849
Replica 7	12,469,092	3,214,832	2,413,233	18,097,157
Replica 8	12,401,716	3,214,832	2,413,233	18,029,781
Replica 9	12,517,341	3,214,832	2,413,233	18,145,406
Replica 10	12,426,822	3,214,832	2,420,946	18,062,600
Average	12,437,867	3,222,364	2,422,178	18,082,409
Std. Dev.	89,461.29	23,647.46	12,947.42	74,446.34

Using Eq. 30 the sample size required to obtain the confidence interval is calculated. Each confidence sample size is presented Table 6.6.

Table 6.6 Sample size for Validation with 0K Registered Times

	0K-3K	3K-8K	8K-10K	Total
Relative Error (interferences)	621,893.3	161,118.2	121,108.9	904,120.5
t(.95,10-1)	2.2621	2.2621	2.2621	2.2621
N	0.105897	0.110236	0.058487	0.034696

Note that from Table 6.6 it can be concluded that the 10 replicas obtained from the simulation are enough to calculate the confidence interval with a 95% confidence. Using Eq. 31 and the statistics from Table 6.4 the confidence intervals presented in the table below are calculated.

Table 6.7 Confidence Interval for Interferences with 0K Assignment

	0K-3K	3K-8K	8K-10K	Total
$\mathbf{t_{\frac{\alpha}{2},n-1} * \frac{s}{\sqrt{n}}}$	63,996.75	16,916.37	9,262.029	53,255.7
Upper Limit	12,501,864	3,239,280	2,431,440	18,135,665
Average	12,437,867	3,222,364	2,422,178	18,082,409
Lower Limit	12,373,870	3,205,448	2,412,916	18,029,153

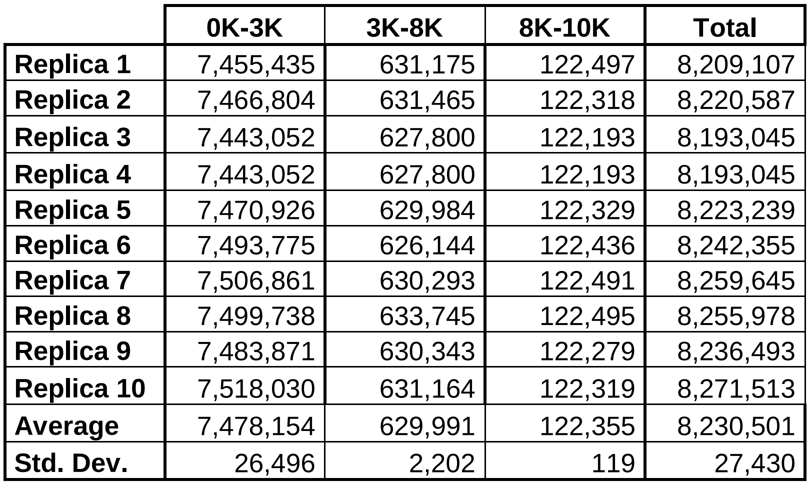
Comparing the WB10K 2014 interferences with Table 6.7 it may be observed that the real values are contained among the confidence interval. This indicates that the model is an accurate representation of the system and can be used to evaluate different corral assignment policies. Interestingly, the swarm intelligence of participants the day of the race (*i.e.,* policy 1) proves to be more effective than the honor based assignment (*i.e.,* policy 0).

In order to propose a corral policy that reduces participants' interferences, four additional policies will be evaluated. Policy 2: Status Quo, will calculate the expected number of interferences using the regression assignment method and the current corral sizes. Policy 3: Equal Size Distribution, will evaluate the performance of equally dividing the participants among the number of corrals. Policy 4: Ascending Size Distribution, will be based on an ascending corral with a percentage of difference among the number of corrals. Policy 5: Descending Size Distribution, will be constructed on a descending percentage of difference among the number of corrals. Policy 6: Waves, will evaluate the implementation of waves between corrals. In addition, given that there are only 8,962 participants in the WB10K 2014 edition for whom the data was available to calculate the regression based estimates the four additional policies will only take into account those participants.

6.3 *Policy 2: Current Corral Size*

Note that the previous policies were evaluated in section 6.2, Policy 1 is the honor bib based assignment and Policy 2 is the swarm intelligence modify assignment. The goal of Policy 2 is to evaluate the current corral division combined with the combined regression method proposed on Section 4.9. The 8,962 participants will be assigned to corrals as shown in Figure 3.1. The participants with the lowest expected finish time from the regression analyses will be assigned to the corrals closer to the front. The results from ten replicas are presented on Table 6.8

Table 6.8 Regression Method with Status Quo



	0K-3K	3K-8K	8K-10K	Total
Replica 1	7,455,435	631,175	122,497	8,209,107
Replica 2	7,466,804	631,465	122,318	8,220,587
Replica 3	7,443,052	627,800	122,193	8,193,045
Replica 4	7,443,052	627,800	122,193	8,193,045
Replica 5	7,470,926	629,984	122,329	8,223,239
Replica 6	7,493,775	626,144	122,436	8,242,355
Replica 7	7,506,861	630,293	122,491	8,259,645
Replica 8	7,499,738	633,745	122,495	8,255,978
Replica 9	7,483,871	630,343	122,279	8,236,493
Replica 10	7,518,030	631,164	122,319	8,271,513
Average	7,478,154	629,991	122,355	8,230,501
Std. Dev.	26,496	2,202	119	27,430

From Table 6.8 it can be observed that the total expected interferences for the current corral policy combined with the regression method is of 8,230,501 with a standard deviation of 27,430. This yields a much smaller number of interferences than the ones recorded in the WB10K 2014 edition, a reduction of more than 10,000,000 interferences by just using the combined regression method presented on Section 4.9 and using the same corral layout presented in Fig. 3.1.

6.4 Policy 3: Equal Size Distribution

The policy 3 equally divides all participants between the corrals. In other words, the total number of participants for a WB10K edition would be divided by the number of corrals ensuring that each corral contains the same amount of participants. Equation 31 illustrates the policy.

$$\frac{i}{n} = w_n * h_n \qquad (31)$$

In Eq. 32 the number of participants (*i*) is divided by the number of corrals. The mathematical division will result in the number of participants per corral, whish is define by the matrix dimensions width (w_n) and height (h_n). This policy seeks to have a uniform corral size. For this policy the width of the corral will be 10 and the length will be modify (*i.e.*, rounded up) to accommodate the corresponding number of participants per corral. In this policy we will evaluate 20 different instances and each instance will have one more corral than the previous. For example in instance 1 all participants are assigned to one corral, then in instance 2 participants will be divided among 2 corrals, later in instance 3 participants will be divided among 3 corrals, and so on. Table 6.9 presents the division of participants per each instance.

Table 6.9 Participants per Corral Policy 3

Number of Corrals	Width	Height
1	10	897
2	10	449
3	10	299
4	10	225
5	10	180
6	10	150
7	10	129
8	10	113
9	10	100
10	10	90
11	10	82
12	10	75
13	10	69
14	10	65
15	10	60
16	10	57
17	10	53
18	10	50
19	10	48
20	10	45

With the variables presented on Table 6.9 the simulation is initialized for each instance and ten replicas are run for each of them. The average and the standard deviation for each replica is presented on Table 6.10.

Table 6.10 Policy 3 Results

Number of Corrals		0K-3K	3K-8K	8K-10K	Total
1	**Average**	10,353,282	722,905	148,382	11,224,568
	Std. Dev.	91,429	5,168	716	95,260
2	**Average**	7,980,836	637,002	124,803	8,742,642
	Std. Dev.	55,425	2,221	311	56,418
3	**Average**	7,507,235	624,920	121,196	8,253,350
	Std. Dev.	37,450	855	243	38,314
4	**Average**	7,427,959	624,484	120,588	8,173,030
	Std. Dev.	21,163	1,114	121	21,309
5	**Average**	7,252,274	618,774	119,512	7,990,560
	Std. Dev.	18,992	861	121	19,062
6	**Average**	7,226,501	620,458	119,505	7,966,464
	Std. Dev.	13,148	479	93	13,241
7	**Average**	7,204,590	616,932	119,054	7,940,576
	Std. Dev.	15,067	732	80	15,336
8	**Average**	7,232,949	618,633	119,293	7,970,874
	Std. Dev.	6,280	388	47	6,374
9	**Average**	7,190,915	617,069	119,023	7,927,007
	Std. Dev.	13,718	431	36	14,049
10	**Average**	7,173,079	617,187	119,052	7,909,318
	Std. Dev.	4,295	419	49	4,597
11	**Average**	7,184,536	618,248	119,137	7,921,921
	Std. Dev.	7,534	442	51	7,577
12	**Average**	7,165,742	616,597	118,949	7,901,288
	Std. Dev.	10,334	354	54	10,582
13	**Average**	7,156,451	616,658	118,872	7,891,981
	Std. Dev.	8,186	349	39	8,459
14	**Average**	7,170,849	616,485	118,993	7,906,327
	Std. Dev.	4,910	457	45	4,919
15	**Average**	7,163,514	616,627	118,883	7,899,024
	Std. Dev.	7,007	403	46	7,360
16	**Average**	7,181,803	616,241	118,940	7,916,984
	Std. Dev.	4,082	296	48	4,203
17	**Average**	7,179,324	617,350	118,983	7,915,656
	Std. Dev.	5,276	283	27	5,497
18	**Average**	7,162,173	616,128	118,839	7,897,140
	Std. Dev.	5,942	266	23	6,084
19	**Average**	7,167,509	616,433	118,941	7,902,883
	Std. Dev.	5,954	235	38	6,071
20	**Average**	7,158,924	616,608	118,857	7,894,390
	Std. Dev.	5,789	200	29	5,822

Note that the lowest number of interferences is achieved with ten corrals, with a total number of interferences of 7,552,495 and a standard deviation of 313,249. With the exception of ten corrals, as the corrals increase the number of interferences decrease and a similar behavior may be observed in the standard deviation. Figure 6.5 illustrates the behavior of the policy evaluated.

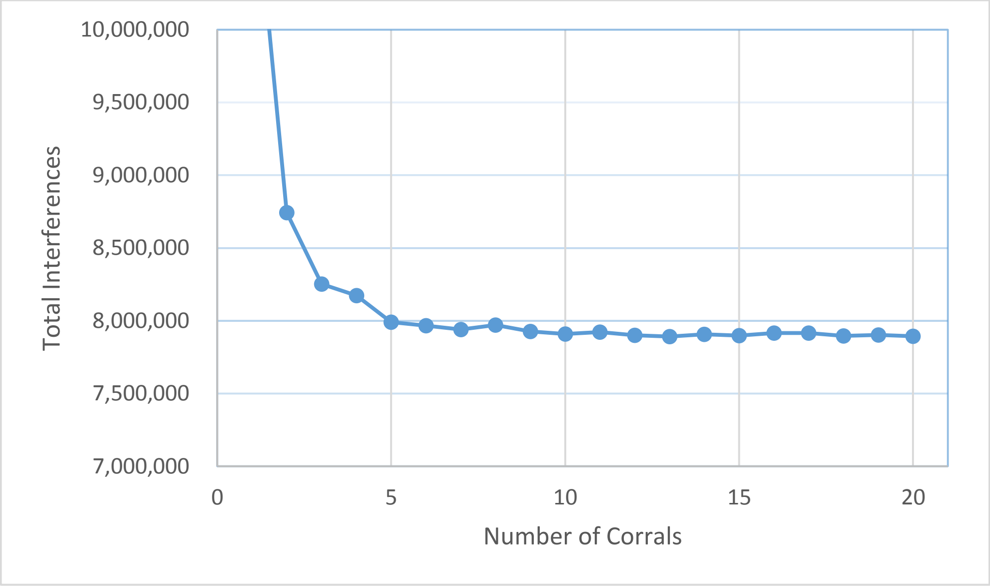


Figure 6.5 Total Interferences for Policy 3

Figure 6.5 may suggest that the best number of corrals for an equal size distribution for the 2014 edition of the WB10K would be twenty. Twenty corrals yield a smaller standard deviation and it results in 7,894,390 of interferences with a standard deviation of 5,822. The following figure better illustrates the confidence interval for each instance.

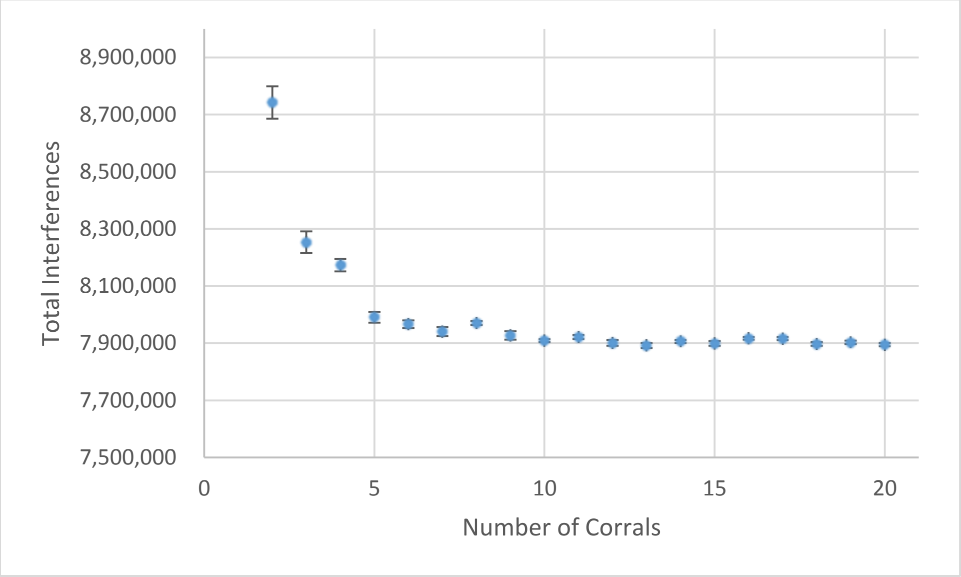


Figure 6.6 Policy 3 Confidence Intervals

In Figure 6.6 it may be observed that twenty corrals yield a smaller margin of errors for the total number of interferences. On the other hand, the intervals overlap each other, for example values obtained with ten corrals are contained within the interval for twenty corrals.

6.5 *Policy 4: Ascending Size Distribution*

With policy 4 each corral will be larger than the one before by a predetermined percentage. For example, if there are three corrals, the second corral will be larger than the first corral by a percentage *P*, and the third corral will be larger than the second corral by the same percentage *P*. A system of equations was constructed to aid in the determination of each corral size and is presented below.

$$\sum_{i=1}^{n} x_i = T \quad (32)$$

$$x_i = x_{i-1} + (P * x_{i-1}) \ \forall \ i \quad (33)$$

$$x_n \geq 0 \tag{34}$$

Equation 32 ensures that the sum of participants per corral (x_i) equals the total number of participants (T). The following equation takes into account a percentage increase (P) of the corral capacity of the previous corral (x_{n-1}). Finally, Eq. 34 does not allow a creation of a corral that can accommodate zero or less participants. If any corral size results in a decimal place it will be rounded to the nearest integer. This policy will evaluate a percentage increase of 5% in each instance. Similarly to Policy 3, there will be twenty instances evaluated for this policy (from one to twenty corrals). The results from Policy 4 are presented on Table 6.11.

Table 6.11 Policy 4 Results

Number of Corrals		0K-3K	3K-8K	8K-10K	Total
N2	**Average**	8,031,415	636,702	124,617	8,792,734
	Std. Dev.	39,057	2,461	307	40,840
N3	**Average**	7,574,213	626,192	121,267	8,321,672
	Std. Dev.	36,037	1,023	200	36,744
N4	**Average**	7,458,331	624,822	120,626	8,203,779
	Std. Dev.	23,908	1,324	109	24,999
N5	**Average**	7,284,279	619,174	119,528	8,022,980
	Std. Dev.	17,985	1,003	121	18,214
N6	**Average**	7,262,478	620,353	119,535	8,002,366
	Std. Dev.	15,423	659	65	15,793
N7	**Average**	7,202,540	618,569	119,241	7,940,351
	Std. Dev.	12,371	723	69	12,780
N8	**Average**	7,201,854	617,634	119,050	7,938,537
	Std. Dev.	14,042	689	44	14,323
N9	**Average**	7,210,338	617,471	119,047	7,946,856
	Std. Dev.	11,348	541	62	11,560
N10	**Average**	7,221,459	617,543	119,055	7,958,057
	Std. Dev.	10,574	563	65	10,821
N11	**Average**	7,197,780	616,603	118,989	7,933,372
	Std. Dev.	9,991	348	48	10,102
N12	**Average**	7,179,761	617,031	118,931	7,915,723
	Std. Dev.	6,091	499	50	6,457
N13	**Average**	7,170,174	616,955	118,901	7,906,029
	Std. Dev.	2,652	314	47	2,493
N14	**Average**	7,187,718	617,013	119,018	7,923,749
	Std. Dev.	7,283	345	45	7,604
N15	**Average**	7,160,336	616,509	118,764	7,895,609
	Std. Dev.	6,998	308	37	6,965
N16	**Average**	7,164,502	616,915	118,892	7,900,309
	Std. Dev.	6,448	219	47	6,671
N17	**Average**	7,172,302	616,015	118,929	7,907,246
	Std. Dev.	4,879	248	30	4,874
N18	**Average**	7,163,119	616,252	118,878	7,898,249
	Std. Dev.	4,943	248	30	5,028
N19	**Average**	7,158,253	616,591	118,924	7,893,768
	Std. Dev.	7,707	264	35	7,837
N20	**Average**	7,151,782	616,314	118,834	7,886,930
	Std. Dev.	4,759	164	37	4,847

Note that instance 1 (*i.e.*, one corral) is not evaluated for this policy given that it would have the same behavior as shown in the previous scenario (Policy 3). From Table 6.11 it may be observed that as the number of corrals increases the interferences decrease. Figure 6.7 illustrates the tendency of the results.

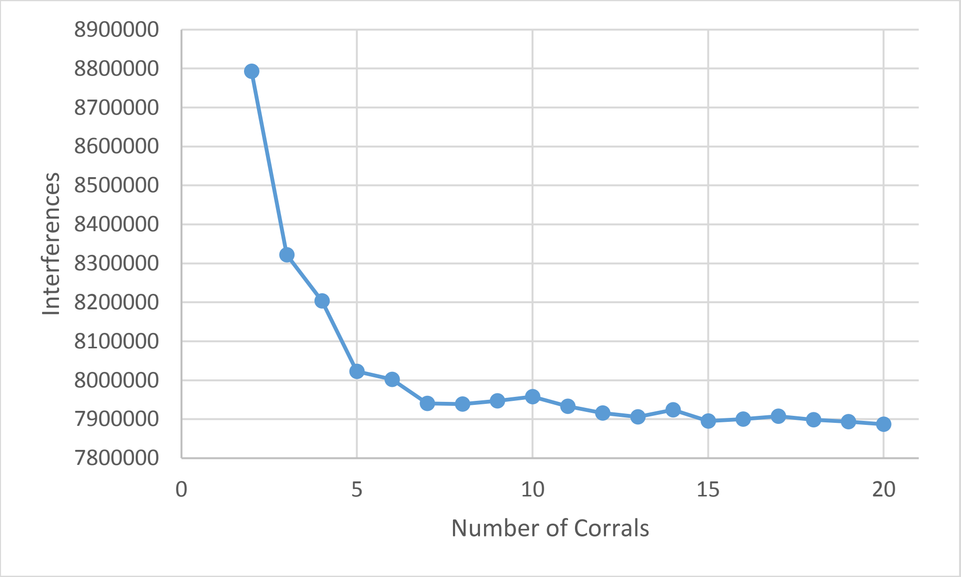


Figure 6.7 Total Interferences Policy 4

From Fig. 6.7 the relationship between the number corrals and interferences may be better appreciated. Hence, the number of corrals that reduces the participants' interferences, among the instances evaluated, is twenty corrals with an expected total interferences of 7,886,930 and a standard deviation of 4,847. To better appreciate the margin of error of each instance the confidence intervals are plotted on Figure 6.8.

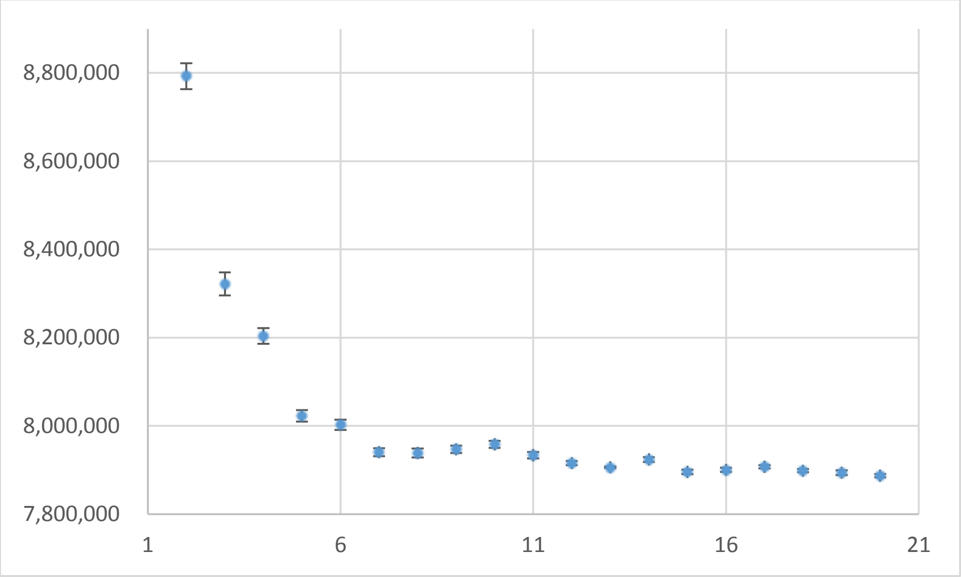


Figure 6.8 Policy 4 Confidence Intervals

From Fig. 6.8 it can be observed that the confidence intervals for each instances overlap between each other. Even though the margins of errors of smaller corrals may contain values yield by the twenty corrals instance it still provides the best results.

6.6 *Policy 5: Descending Size Distribution*

The policy evaluated in this section is based on a decreasing behavior in the total of participants per corral. Each corral will be smaller than the one in front by a predetermined coefficient. A system of equations similar to the one in policy 4, presented on Section 6.5, is used to determine the corral size.

$$x_i = x_{i-1} - (P * x_{i-1}) \ \forall \, i \qquad (35)$$

Using Equations 32 and 34 from Section 6.5 and replacing Equation 33 for Equation 35 will ensure that the participants per corral follow the policy described. Note that this policy will result in the same number of participants per corral as in Policy 4, nevertheless the positions of the larger corrals are inverted. The summarized results of ten replicas per instance is presented in Table 6.12.

Table 6.12 Policy 5 Results

Number of Corrals		0K-3K	3K-8K	8K-10K	Total
1	Average	7,842,313	635,083	124,262	8,601,658
	Std. Dev.	54,339	1,752	223	54,711
2	Average	7,511,302	626,356	121,631	8,259,289
	Std. Dev.	35,751	1,276	188	36,603
3	Average	7,511,302	626,356	121,631	8,259,289
	Std. Dev.	35,751	1,276	188	36,603
4	Average	7,387,980	624,264	120,399	8,132,644
	Std. Dev.	20,526	1,059	138	20,422
5	Average	7,256,119	620,665	119,874	7,996,658
	Std. Dev.	16,633	980	118	16,882
6	Average	7,232,421	620,960	119,551	7,972,932
	Std. Dev.	10,481	666	99	10,788
7	Average	7,226,306	618,922	119,261	7,964,488
	Std. Dev.	10,433	404	75	10,229
8	Average	7,183,814	617,359	119,102	7,920,274
	Std. Dev.	10,925	542	108	11,131
9	Average	7,173,469	618,036	119,077	7,910,581
	Std. Dev.	7,484	712	80	7,782
10	Average	7,176,964	618,427	119,059	7,914,450
	Std. Dev.	6,311	588	48	6,017
11	Average	7,172,638	617,163	119,040	7,908,842
	Std. Dev.	9,765	526	63	10,057
12	Average	7,177,994	616,570	119,022	7,913,586
	Std. Dev.	7,411	380	45	7,505
13	Average	7,179,346	617,700	118,977	7,916,022
	Std. Dev.	8,793	447	46	9,009
14	Average	7,174,794	617,104	118,961	7,910,859
	Std. Dev.	6,620	332	49	6,627
15	Average	7,160,044	616,591	118,944	7,895,579
	Std. Dev.	4,747	469	62	4,843
16	Average	7,160,845	617,155	118,990	7,896,989
	Std. Dev.	4,731	480	41	5,089
17	Average	7,167,137	616,436	118,901	7,902,474
	Std. Dev.	4,932	373	64	5,013
18	Average	7,152,062	616,030	118,783	7,886,874
	Std. Dev.	3,617	284	44	3,573
19	Average	7,172,889	616,955	119,000	7,908,844
	Std. Dev.	5,620	304	35	5,510
20	Average	7,158,320	616,343	118,807	7,893,469
	Std. Dev.	4,376	231	32	4,445

From Table 6.12 it may be appreciated that the same descending behavior form policy 3 and 4 is present. As the number of corrals increase the number of interferences decrease. To better appreciate the tendency the data was plotted on the graph in Fig. 6.9.

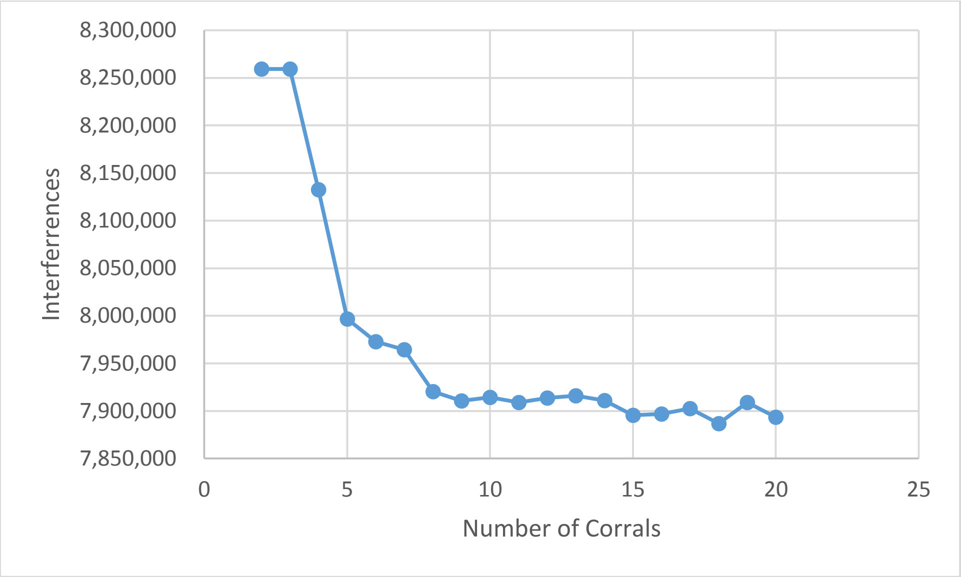


Figure 6.9 Total Interferences Policy 5

In the graph, Fig. 6.9, it may be observed that the more corrals the less interferences between participants. This suggest that having twenty corrals will in fact yield less interferences during the race. The minimum number of interferences is achieved with eighteen corrals with a mean of 7,886,474 and a standard deviation of 3,573. Figure 6.10 shows the confidence intervals for each instance.

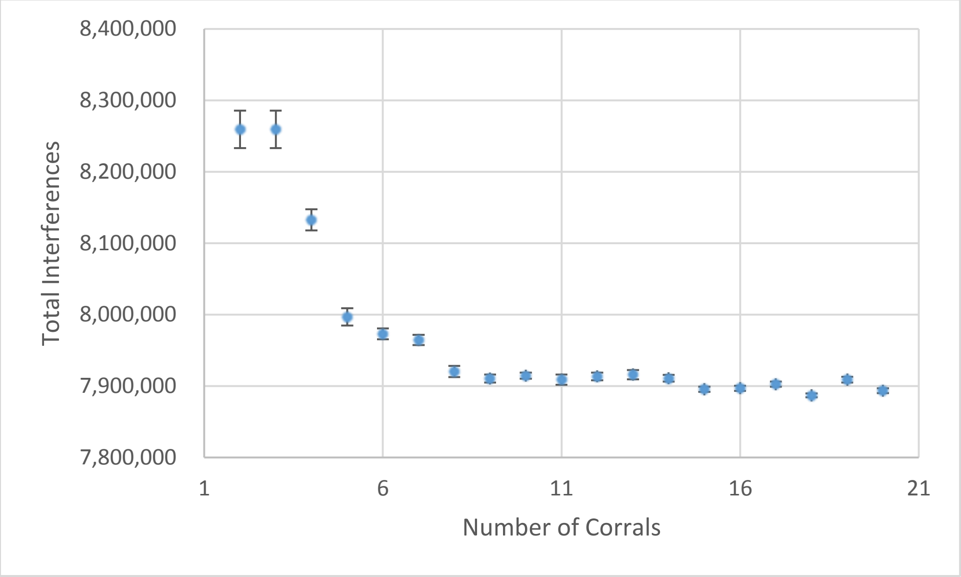


Figure 6.10 Policy 5 Confidence Intervals

In Fig. 6.10 it may be appreciated that the confidence intervals overlap each other. Particularly, in the eighteen corral instance, the results are also contained within the twenty corral instance.

6.7 *Policy 6: Waves*

For this policy each corral will be release at a different gun time this particular policy is used at other large race events. The current corral assignment policy presented on Policy 2 will be evaluated with waves of 1, 2, 3, 4, 5, 10, 15, 20, 25, and 30 minutes.

Table 6.13 Waves with Status Quo and Regression Method

Time Between Corrals		0K-3K	3K-8K	8K-10K	Total
1 min.	**Average**	7,126,422	607,744	118,671	7,852,836
	Std. Dev.	26,082	1,507	160	26,760
2 min.	**Average**	6,852,087	588,349	115,446	7,555,881
	Std. Dev.	24,189	1,412	129	24,763
3 min.	**Average**	6,617,076	569,870	112,367	7,299,312
	Std. Dev.	28325	1289	170	28804
4 min.	**Average**	6,405,483	552,168	109,510	7,067,162
	Std. Dev.	32066	916	157	32177
5 min.	**Average**	6,269,116	541,883	107,586	6,918,584
	Std. Dev.	28,633	1,468	144	29,675
10 min.	**Average**	5,541,036	465,152	95,242	6,101,430
	Std. Dev.	28,750	2,009	154	29,626
15 min.	**Average**	5,117,960	405,838	85,383	5,609,182
	Std. Dev.	26,512	1,261	132	26,736
20 min.	**Average**	4,842,827	360,756	77,579	5,281,162
	Std. Dev.	47,430	2,329	118	48,963
25 min.	**Average**	4,694,217	327,828	71,200	5,093,245
	Std. Dev.	30,893	1,406	120	31,259
30 min.	**Average**	4,634,950	304,864	66,281	5,006,096
	Std. Dev.	36,595	1,045	76	36,810

From Table 6.13 it can be observed that the waves minimize the interferences between participants. In Figure 6.11 the effect of the waves over the total interferences can be appreciated.

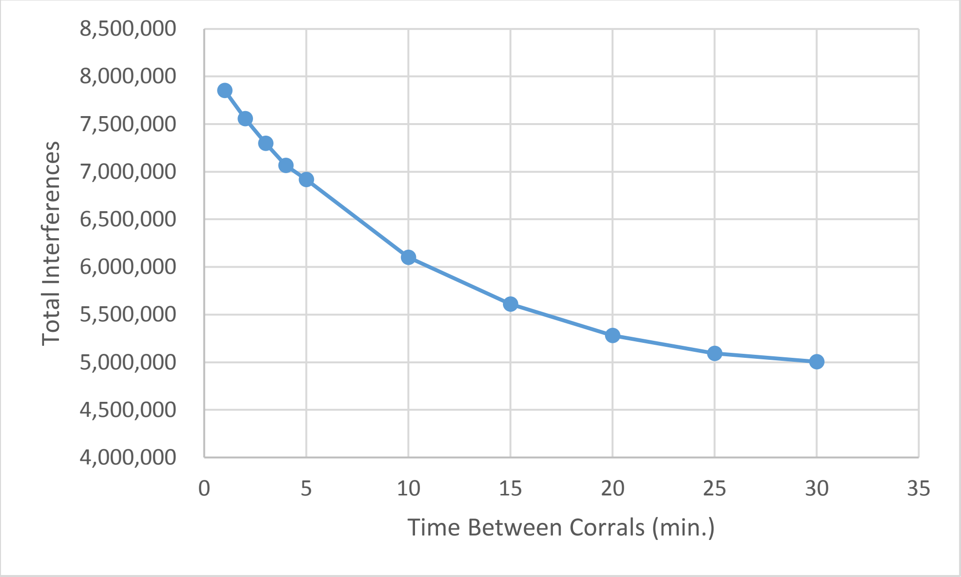


Figure 6.11 Total Interferences with Waves

Figure 6.11 illustrates that the larger the time between corrals the less interferences among participants. To better observe the margin of error for each instance the confidence intervals are shown in Fig 6.12.

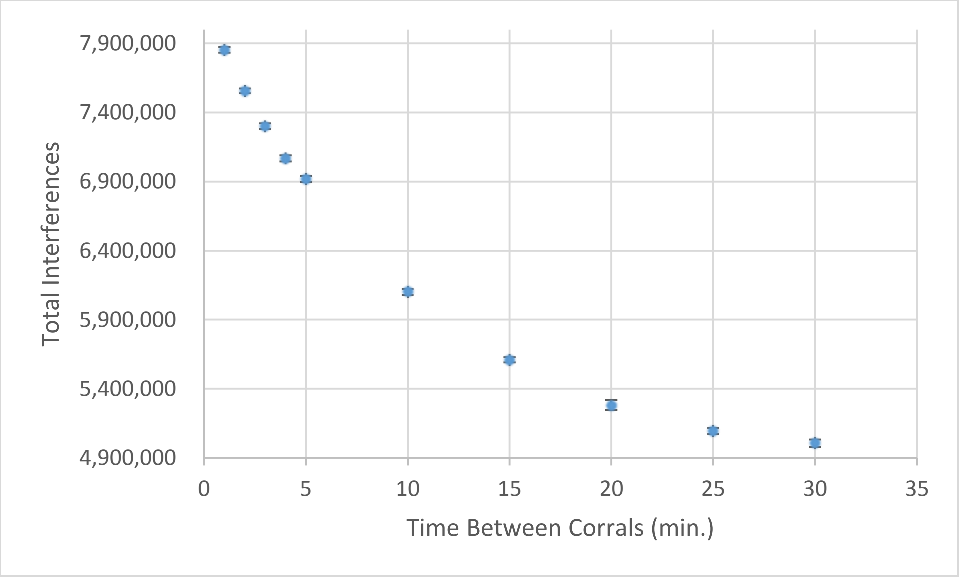


Figure 6.12 Policy 6 Confidence Intervals

Figure 6.12 suggest that results yield by the different wave sizes do not overlap each other. This suggest that the results obtained by implementing a 30 minute wave cannot be achieved with any other wave size. The downside of implementing large wave times is that it will make the event last longer. Table 6.14 shows the additional time each wave length would add to the race.

Table 6.14 Additional Time and Expected Interferences

Wave Length (in min.)	Additional Time (in min.)	Expected Interferences
1	4	7,852,836
2	8	7,555,881
3	12	7,299,312
4	16	7,067,162
5	20	6,918,584
10	40	6,101,430
15	60	5,609,182
20	80	5,281,162
25	100	5,093,245
30	120	5,006,096

Note that as the wave length increases the additional time also increases, nonetheless the expected number of interferences decrease. Implementing 30 minute waves may have a negative effect on the participation quorum by extending the event two hours, as mention previously the WB10K takes place on a Sunday and ends near to sun down. An option to address the effect would be to start the event earlier in order for the participants to get to their homes or hotels at a reasonable time. Furthermore, if the wave policy is compared to the status quo, hence four corrals with no waves, it can be observed that the one minute waves yield a smaller number of interferences. With one minute waves the total interferences range among 7,852,836 meanwhile without the waves it is among 8,230,501. Adding a one minute wave only adds 4 minutes to the duration of the WB10K and reduces interferences between participants. Additional simulations with Policies 3, 4, and 5 are executed in order to observe the effect of adding one minute waives between corrals, Table 6.15 summarizes the results for Policy 3.

Table 6.15 One Minute Wave Policy 3

Number of Corrals		0K-3K	3K-8K	8K-10K	Total
2	**Average**	7,825,364	627,704	122,991	8,576,059
	Std. Dev.	44,688	1,882	219	45,431
3	**Average**	7,228,631	608,532	118,283	7,955,445
	Std. Dev.	25,920	1,617	112	25,446
4	**Average**	7,035,419	602,382	116,549	7,754,349
	Std. Dev.	14,028	638	96	14,226
5	**Average**	6,754,692	590,393	114,420	7,459,504
	Std. Dev.	19,208	767	119	19,241
6	**Average**	6,638,104	586,281	113,536	7,337,920
	Std. Dev.	9,813	848	81	9,931
7	**Average**	6,520,889	578,016	112,125	7,211,029
	Std. Dev.	10,851	441	80	10,881
8	**Average**	6,450,630	573,976	111,357	7,135,963
	Std. Dev.	13,161	552	73	13,420
9	**Average**	6,323,324	566,552	110,126	7,000,001
	Std. Dev.	7,049	277	43	7,088
10	**Average**	6,223,001	560,981	109,220	6,893,202
	Std. Dev.	7,873	258	30	7,837
11	**Average**	6,149,652	556,559	108,373	6,814,584
	Std. Dev.	6,917	410	38	6,940
12	**Average**	6,049,505	549,736	107,300	6,706,541
	Std. Dev.	6,120	249	40	6,202
13	**Average**	5,962,785	543,957	106,237	6,612,979
	Std. Dev.	6,964	318	36	7,085
14	**Average**	5,909,714	540,000	105,594	6,555,308
	Std. Dev.	5,183	220	38	5,247
15	**Average**	5,814,603	533,686	104,498	6,452,787
	Std. Dev.	3,400	125	30	3,419
16	**Average**	5,761,859	529,625	103,814	6,395,298
	Std. Dev.	5,019	226	21	5,176
17	**Average**	5,680,634	524,145	102,852	6,307,630
	Std. Dev.	5,845	168	25	5,930
18	**Average**	5,595,475	517,992	101,805	6,215,272
	Std. Dev.	7,218	238	35	7,395
19	**Average**	5,549,508	514,568	101,275	6,165,350
	Std. Dev.	5,008	115	19	5,073
20	**Average**	5,442,607	507,434	99,985	6,050,025
	Std. Dev.	4,597	121	22	4,571

From Table 6.15 it can be observed that as the number of corral increases the interferences decreases, as seen on the results of Policy 3. To better appreciate this effect the results are plotted with their margin errors in Figure 6.13.

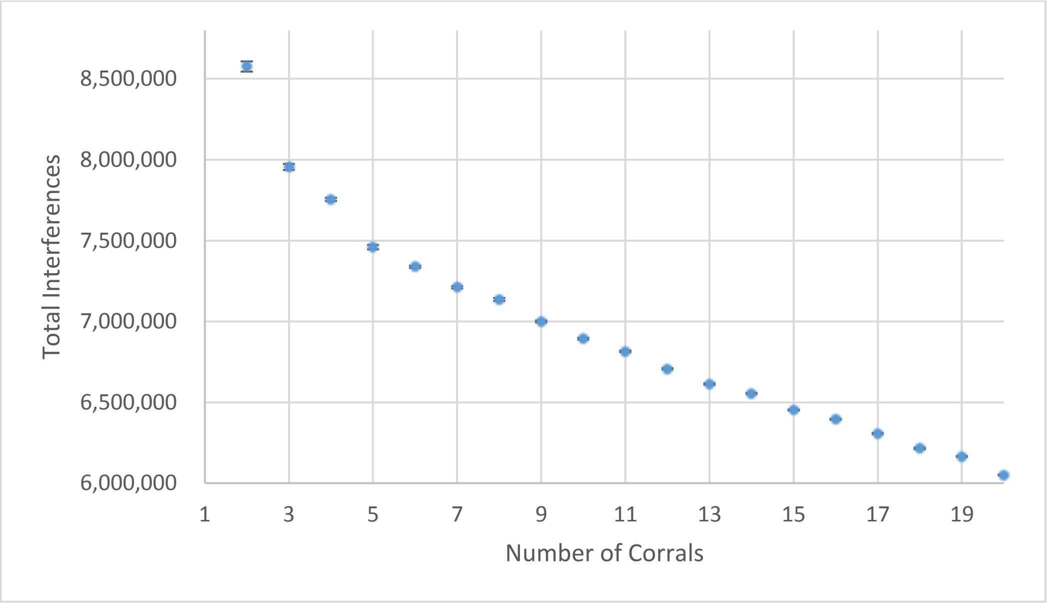


Figure 6.13 One Minute Wave Policy 3: Margin of Errors

Hence, if we compared the results obtained in Policy 3 with 20 corrals with the addition of a one minute wave the numbers go from 7,894,390 to 6,050,025. The effect of adding a one minute wave any Policy with 20 corrals will add 20 minutes to the total duration of the race. Similar results are obtained from the addition of one minute waves to Policy 4, see Table 6.16.

Table 6.16 One Minute wave Policy 4

Number of Corrals		0K-3K	3K-8K	8K-10K	Total
N2	**Average**	7,626,775	579,144	114,948	8,320,867
	Std. Dev.	49,330	1,500	254	50,772
N3	**Average**	7,273,184	609,766	118,344	8,001,293
	Std. Dev.	33,847	1,093	161	34,346
N4	**Average**	7,061,330	602,379	116,613	7,780,321
	Std. Dev.	23,020	742	102	23,508
N5	**Average**	6,796,136	591,101	114,436	7,501,673
	Std. Dev.	15,898	733	95	16,242
N6	**Average**	6,671,589	586,292	113,539	7,371,420
	Std. Dev.	10,553	481	75	10,753
N7	**Average**	6,533,873	578,871	112,251	7,224,995
	Std. Dev.	18,118	420	65	18,317
N8	**Average**	6,438,243	572,464	111,148	7,121,855
	Std. Dev.	13,385	326	52	13,508
N9	**Average**	6,363,091	566,774	110,194	7,040,059
	Std. Dev.	9,528	279	40	9,608
N10	**Average**	6,282,417	561,538	109,264	6,953,219
	Std. Dev.	7,783	335	38	7,829
N11	**Average**	6,182,357	555,369	108,317	6,846,043
	Std. Dev.	7,586	299	66	7,888
N12	**Average**	6,083,769	550,946	107,339	6,742,055
	Std. Dev.	11,652	335	49	11,900
N13	**Average**	5,999,615	545,063	106,371	6,651,049
	Std. Dev.	8,939	230	36	8,929
N14	**Average**	5,949,951	540,742	105,726	6,596,419
	Std. Dev.	11,218	204	37	11,308
N15	**Average**	5,856,443	534,691	104,523	6,495,657
	Std. Dev.	6,484	168	21	6,545
N16	**Average**	5,792,818	530,490	103,886	6,427,194
	Std. Dev.	4,304	161	20	4,376
N17	**Average**	5,744,850	525,828	103,207	6,373,885
	Std. Dev.	4,938	167	30	5,013
N18	**Average**	5,667,831	520,608	102,256	6,290,695
	Std. Dev.	7,048	163	27	7,129
N19	**Average**	5,607,750	517,052	101,577	6,226,379
	Std. Dev.	8,407	79	14	8,445
N20	**Average**	5,532,418	511,098	100,530	6,144,045
	Std. Dev.	3,902	142	22	3,962

When comparing the results from Table 6.16 the least number of interferences is yielded by 20 corrals for a total of 6,144,045 that represents a reduction in comparison with Policy 4 alone (Policy 4 with 20 corrals yielded 7,886,930 interferences). The following figure presents the margin of errors.

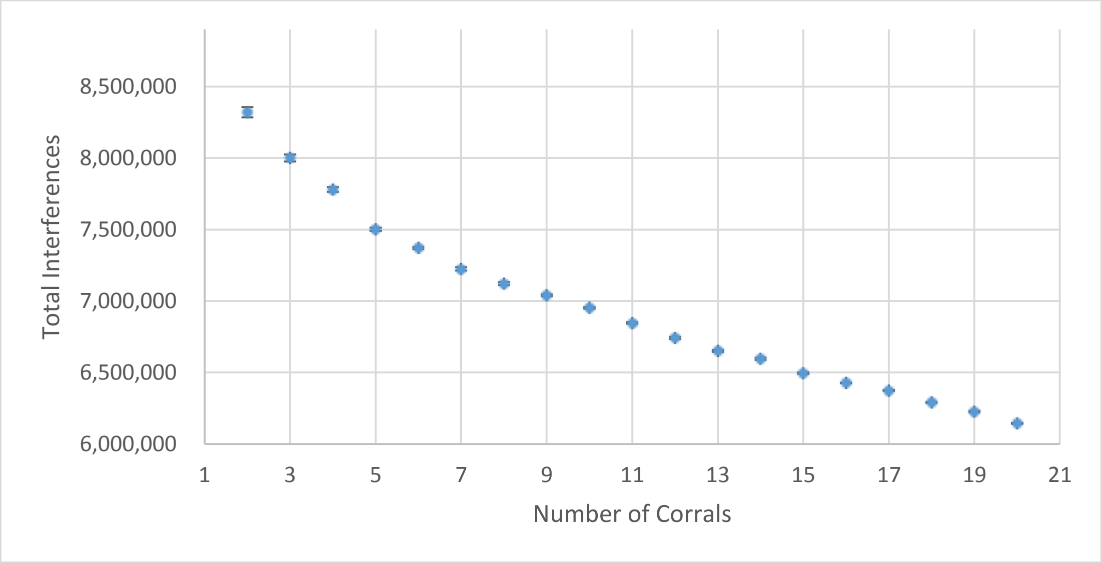


Figure 6.14 One Minute Waves Policy 4: Margin Errors

From Figure 6.14 it is shown that, just like with the original policy, as the number of corrals increase the number of interferences decreases. The following table presents the results from adding one minute waves to Policy 5.

Table 6.17 One Minute Wave Policy 5

Number of Corrals		0K-3K	3K-8K	8K-10K	Total
N2	**Average**	7,714,761	625,500	122,654	8,462,915
	Std. Dev.	50,911	1,910	314	51,545
N3	**Average**	7,249,027	555,459	118,702	7,923,188
	Std. Dev.	32,707	173,810	146	200,553
N4	**Average**	7,013,193	602,561	116,490	7,732,245
	Std. Dev.	23,801	1,405	118	24,835
N5	**Average**	6,757,123	591,731	114,756	7,463,610
	Std. Dev.	13,743	449	65	13,731
N6	**Average**	6,630,861	587,353	113,547	7,331,761
	Std. Dev.	11,890	558	73	12,252
N7	**Average**	6,541,469	579,868	112,376	7,233,713
	Std. Dev.	12,597	670	56	12,753
N8	**Average**	6,405,039	572,708	111,219	7,088,965
	Std. Dev.	13,067	545	69	13,173
N9	**Average**	6,313,624	567,908	110,292	6,991,824
	Std. Dev.	11,028	507	48	11,205
N10	**Average**	5754897	471431	93494	6319822
	Std. Dev.	13874	358	32	14086
N11	**Average**	6,129,754	556,298	108,409	6,794,461
	Std. Dev.	4,817	267	51	5,017
N12	**Average**	6,063,809	550,505	107,536	6,721,850
	Std. Dev.	7,269	341	31	7,397
N13	**Average**	5,979,113	545,843	106,566	6,631,521
	Std. Dev.	7,595	334	38	7,735
N14	**Average**	5,897,177	540,668	105,720	6,543,565
	Std. Dev.	8,928	134	44	8,959
N15	**Average**	5,805,825	534,756	104,819	6,445,400
	Std. Dev.	3,554	195	26	3,584
N16	**Average**	5,740,419	530,824	104,135	6,375,377
	Std. Dev.	5,970	216	32	6,084
N17	**Average**	5,672,140	525,935	103,265	6,301,340
	Std. Dev.	6,065	274	20	6,262
N18	**Average**	5,590,625	521,155	102,393	6,214,172
	Std. Dev.	4,435	274	33	4,568
N19	**Average**	5,541,877	517,449	101,835	6,161,161
	Std. Dev.	6,482	209	18	6,507
N20	**Average**	5,456,243	511,523	100,723	6,068,489
	Std. Dev.	2556	222	27	2517

The results from Table 6.17 are presented graphically in Figure 6.15 to better observe the tendency and the margins of error of each point.

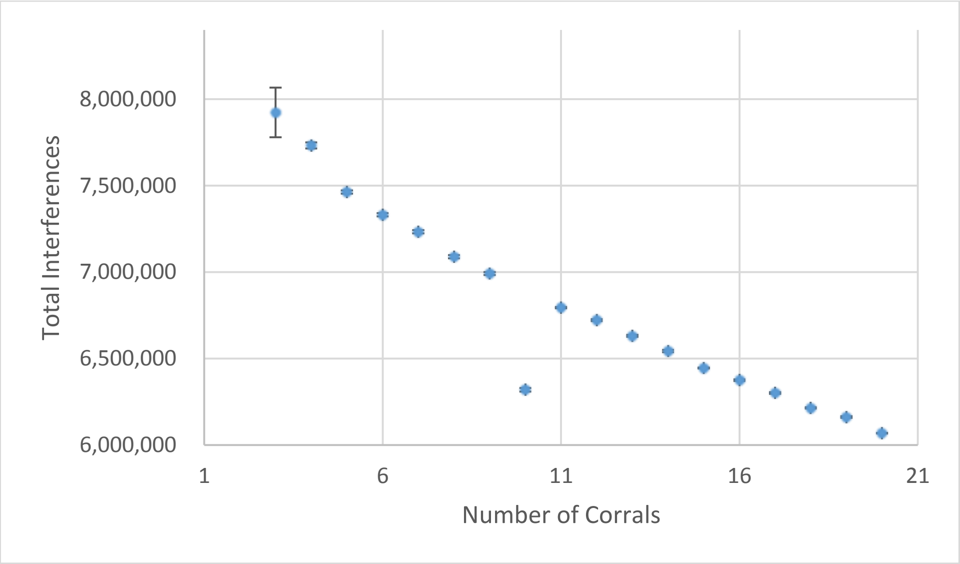


Figure 6.15 One Minute Waves Policy 5: Margin of Errors

The same behavior observed with Polies 3 and 4 is observed in Policy 5. The addition of a one minute wave reduces the number of interferences between participants. Particularly in Policy 5 it reduces it from 7,893,469 to 6,068,489. Even though one minute waves are not the waves that yield the less number of interferences in the WB10K they reduce the number of interferences even further when combined with other policies. Nonetheless better results may be obtained with 30 minute waves but adopting this practice may damage the appeal of the WB10K to participants.

6.8 *Policies Results Comparison*

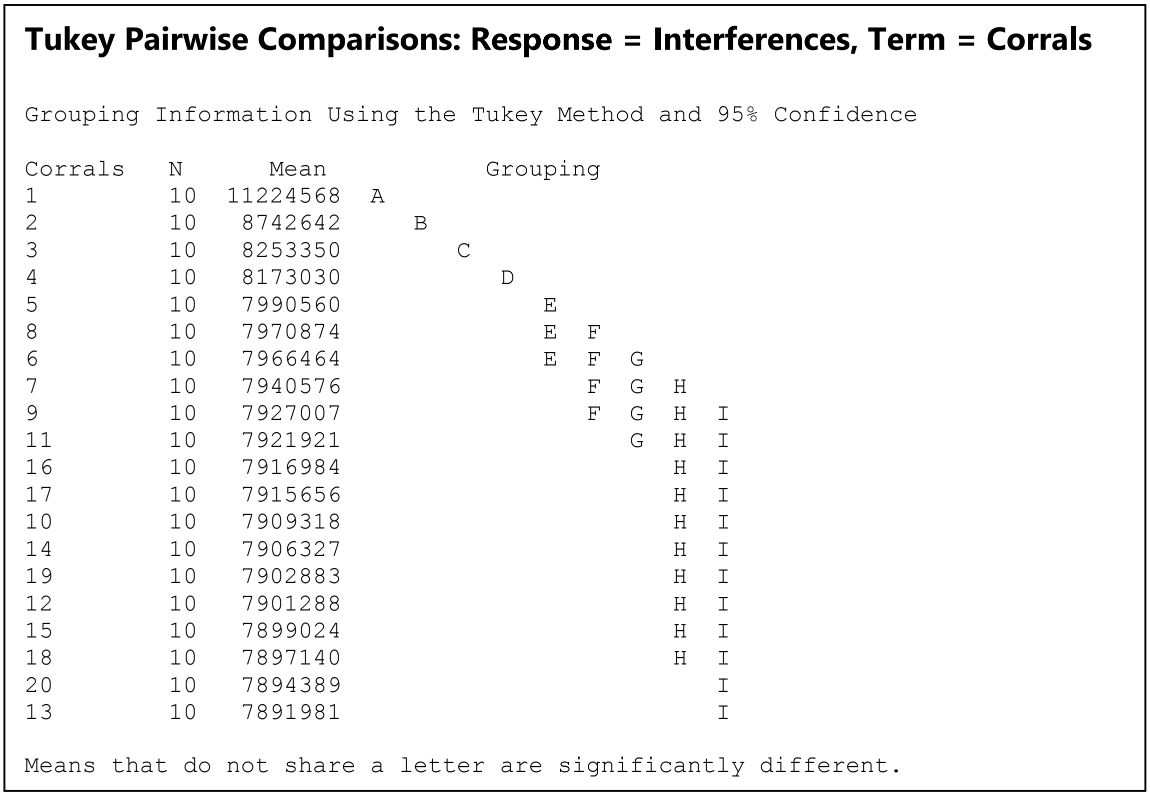
All the different corral assignment policies evaluated in this chapter minimized the number of interferences between participants by more 10,000,000 when compared to the WB10K 2014 edition. The results for the different policies are summarized in Table 6.18 and are compared with the current policy, honor based bib assignment.

Table 6.18 Policies Results

	Number of Corrals	Total Interferences	Standard Deviation	Percentage of difference
Policy 0: Honor Based	4	38,411,134	38,254	
Policy 1: Swarm Intelligence	4	18,082,409	74,446	-53%
Policy 2: Current Size	4	8,230,501	27,430	-79%
Policy 3: Equal Size	13	7,891,981	8,459	-79%
Policy 4: Ascending Size	18	7,886,874	3,573	-79%
Policy 5: Descending Size	20	7,886,930	4,847	-79%
Policy 6: Waves	4	5,006,096	36,810	-87%

From Table 6.18 it may be observed that the participants swarm intelligence is reduces interferences by 53%, proving to be better than the honor based policy. Note that the policy that yield the lowest interferences is policy 6 (*i.e.*, the implementation of waves). The 5,006,096 result from the same corral design used in previous WB10K editions nonetheless it uses the regression method to assign bib numbers. Policy 6 yields an 87% decrease of interferences when compared to the current policy used by the WB10K (*i.e.*, honor based). Table 6.18 may also suggest that a combination between Policy 3 and Policy 6 would yield a lower number of interferences. From Sections 6.4 to 6.6 it was observed that the more corrals were implemented the less interferences would occur. Creating more corrals attacks the random position effect upon the total interferences.

In order to ensure that the different number of corrals have a statistical significant difference between them a Tukey Pairwise Comparison at 95% a confidence level is presented. Figure 6.16 presents the Tukey analysis for Policy 3 is evaluated.

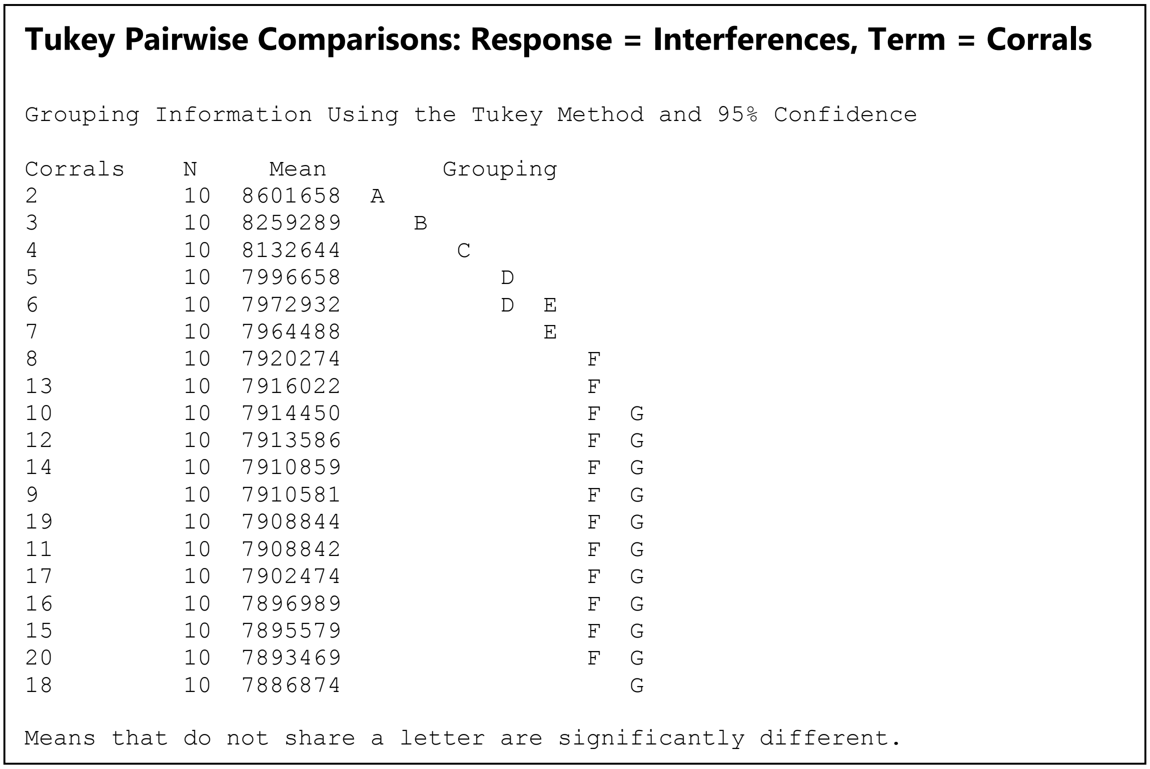


Tukey Pairwise Comparisons: Response = Interferences, Term = Corrals

```
Grouping Information Using the Tukey Method and 95% Confidence

Corrals   N      Mean           Grouping
1         10  11224568  A
2         10   8742642     B
3         10   8253350        C
4         10   8173030           D
5         10   7990560              E
8         10   7970874              E  F
6         10   7966464              E  F  G
7         10   7940576                 F  G  H
9         10   7927007                 F  G  H  I
11        10   7921921                    G  H  I
16        10   7916984                       H  I
17        10   7915656                       H  I
10        10   7909318                       H  I
14        10   7906327                       H  I
19        10   7902883                       H  I
12        10   7901288                       H  I
15        10   7899024                       H  I
18        10   7897140                       H  I
20        10   7894389                          I
13        10   7891981                          I

Means that do not share a letter are significantly different.
```

Figure 6.16 Tukey Pairwise for Policy 3

From the Tukey comparison test it can be observed that adding a corral does not necessarily has a statistically significant impact in the number of interferences. In that same contest it may be concluded that there is a significant difference between having 1, 2, 3, and 4 corrals. Meanwhile there is no statistical difference between having 9 corrals or up to twenty. In other words instead of implementing Policy 3 with 20 corrals it may be implemented with 9 corrals and obtain similar results. The same test is evaluated with the results of the Policy 4 simulations, see Figure 6.17.

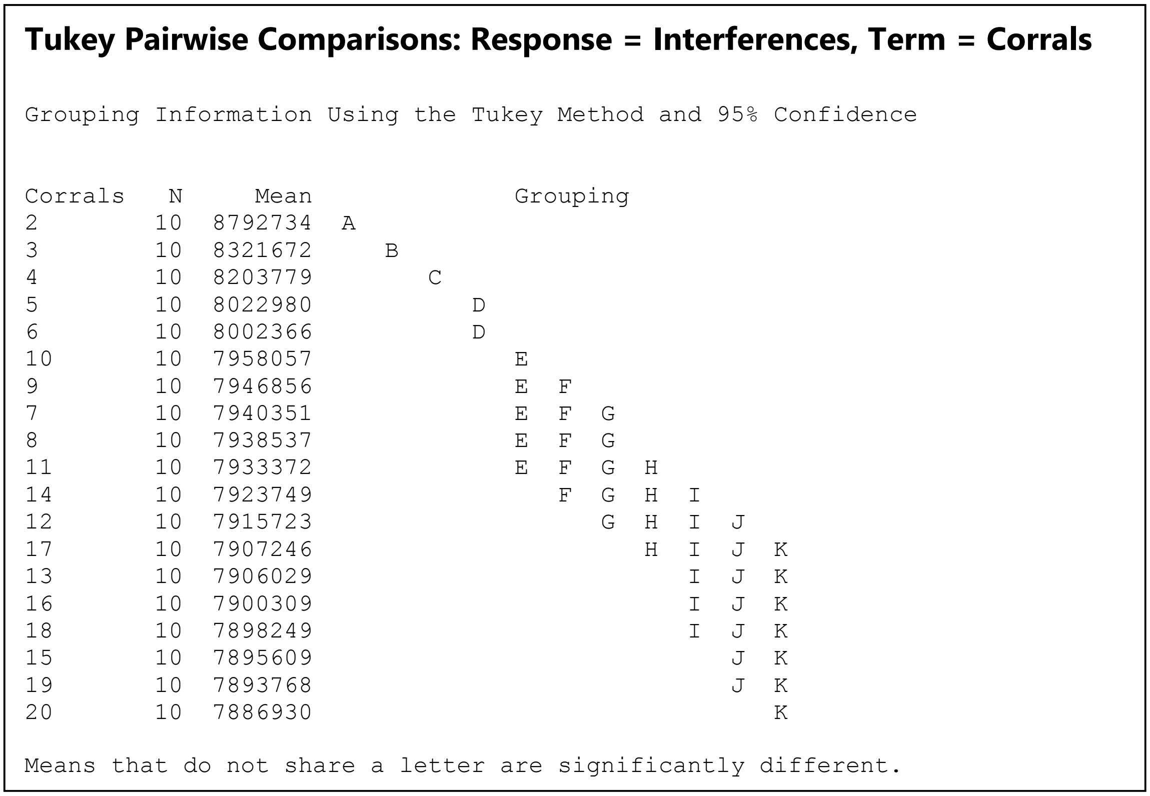


Tukey Pairwise Comparisons: Response = Interferences, Term = Corrals

```
Grouping Information Using the Tukey Method and 95% Confidence

Corrals    N     Mean        Grouping
2          10  8601658  A
3          10  8259289     B
4          10  8132644        C
5          10  7996658           D
6          10  7972932           D  E
7          10  7964488              E
8          10  7920274                 F
13         10  7916022                 F
10         10  7914450                 F  G
12         10  7913586                 F  G
14         10  7910859                 F  G
9          10  7910581                 F  G
19         10  7908844                 F  G
11         10  7908842                 F  G
17         10  7902474                 F  G
16         10  7896989                 F  G
15         10  7895579                 F  G
20         10  7893469                 F  G
18         10  7886874                    G

Means that do not share a letter are significantly different.
```

Figure 6.17 Tukey Stepwise for Policy 4

From Fig. 6.17 it can be appreciated that with fewer than 20 corrals a desirable number of interferences may be obtain with Policy 4. In fact with 9 corrals the resulting interferences do not show a statistical significant difference when compare with scenarios that include corrals. In Figure 6.18 the Tukey comparison is presented for the different number of corrals following Policy 5.

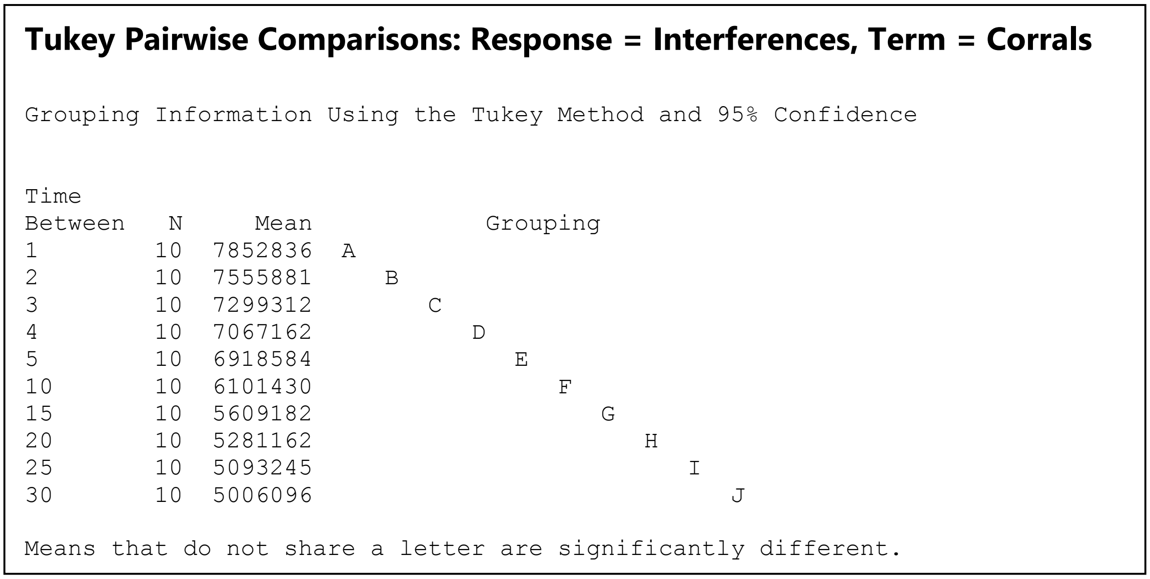


Tukey Pairwise Comparisons: Response = Interferences, Term = Corrals

```
Grouping Information Using the Tukey Method and 95% Confidence


Corrals   N     Mean              Grouping
2        10  8792734  A
3        10  8321672     B
4        10  8203779        C
5        10  8022980           D
6        10  8002366           D
10       10  7958057              E
9        10  7946856              E  F
7        10  7940351              E  F  G
8        10  7938537              E  F  G
11       10  7933372              E  F  G  H
14       10  7923749                 F  G  H  I
12       10  7915723                    G  H  I  J
17       10  7907246                       H  I  J  K
13       10  7906029                          I  J  K
16       10  7900309                          I  J  K
18       10  7898249                          I  J  K
15       10  7895609                             J  K
19       10  7893768                             J  K
20       10  7886930                                K

Means that do not share a letter are significantly different.
```

Figure 6.18 Tukey Stepwise for Policy 5

From Fig. 6.18 it can be seen that having 13 corrals will not yield interferences that represent a statistical difference when compared with having 20 corrals. Finally, a Tukey test was also executed to identify the statistical differences between the wave sizes and is presented in Figure 6.19.

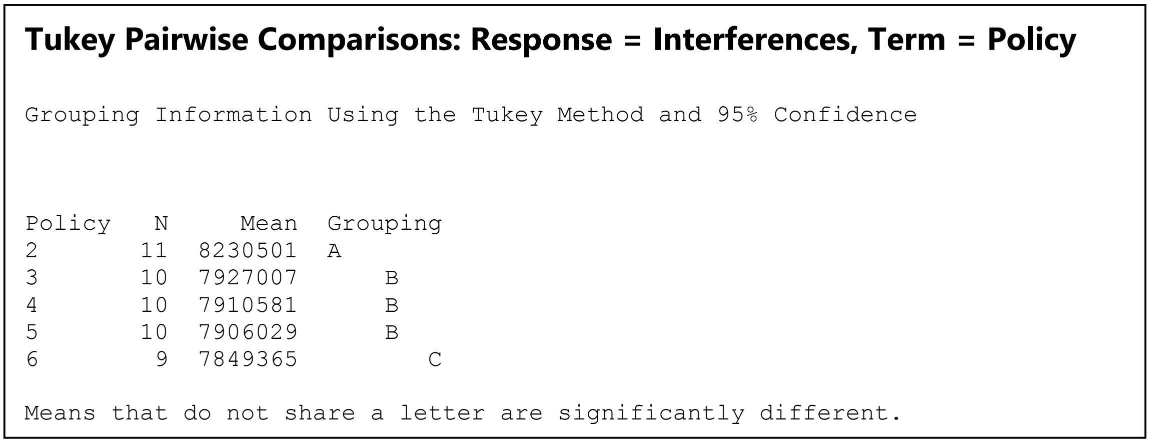


Tukey Pairwise Comparisons: Response = Interferences, Term = Corrals

```
Grouping Information Using the Tukey Method and 95% Confidence


Time
Between   N     Mean            Grouping
1        10  7852836  A
2        10  7555881     B
3        10  7299312        C
4        10  7067162           D
5        10  6918584              E
10       10  6101430                 F
15       10  5609182                    G
20       10  5281162                       H
25       10  5093245                          I
30       10  5006096                             J

Means that do not share a letter are significantly different.
```

Figure 6.19 Tukey Stepwise for Policy 6

Contrary to the other Policies the implementation of waves present a statistical difference between the different time intervals. This means that none of the waves studied will produce results similar with each other. From these statistical analyses it can be concluded that a smaller corral in Policies 3, 4, and 5 may produce results that do not present any statistical difference from the implementation of 20 corrals in each case. In order to determine which policy represents the statistical significant best alternative a Tukey comparison will be made between the fewest number of corrals needed in each policy and this will be compared to the current size of corrals and the waves implementation.



Tukey Pairwise Comparisons: Response = Interferences, Term = Policy

```
Grouping Information Using the Tukey Method and 95% Confidence



Policy   N     Mean  Grouping
2       11  8230501  A
3       10  7927007     B
4       10  7910581     B
5       10  7906029     B
6        9  7849365        C

Means that do not share a letter are significantly different.
```

Figure 6.20 Stepwise Between Policies

From the analysis presented on Figure 6.20 it can be observed that there is no significant difference between Policies 3, 4, and 5. This suggests that adopting either ascending, descending or equal size corrals would result on similar results. Nonetheless, either policy (3, 4, and 5) would yield a better result than the current corral size (Policy 2). Hence, Policy 6 proves to be statistically better that the other polices, even when compared to implementing only one minute waves. Note that it was proven that the larger the time between corrals the less the interferences, with a statistical significance. By adding waves in either Policy 3, 4, or 5 should improve the results.

7. Control Methods

It is important to highlight that even though participants are assigned to a corral during the WB10K, there is no guarantee that participants will actually present themselves to the correct corral. In Section 6.2 it was observed that the participants did not follow the current policy. In fact the positions selected by the participants yield better results than the honor based bib assignment. From Chapter 6 it was concluded that having a regression method bib assignment and assuring everyone follows the policy reduces participants' interference. To avoid situation where participants try to ignore their corral assignment, disqualifying any participant that passes the 0K checkpoint before their assigned corral has been cleared to start, can serve as a control method. Additionally an inscription penalty may be implemented for those participants who started on a different corral position than the assigned on the previous WB10K edition.

Also, from Chapter 6 it was observed that as the number of corrals increased the interferences were reduce. This is an effect of ensuring that the corral assignment is follow. In addition the smaller the corral size, the less room for randomness within the corral. By having smaller corrals during the WB10K, a strict corral assignment may be enforced given that there will be smaller groups to monitor. To ensure that everyone enters their correct corral staff may be placed at the entrance of each corral or the front corrals to verify that participants with the correct bib number enter those corrals.

8. Conclusions

This study proposes a new method to assign WB10K participants to corrals in order to reduce the interference between participants during the race. Interference is measured by the number of passes between participants. By carefully studying WB10K historical data, it is established that most interference occurs between the first checkpoints. Hence, it is concluded that the current method used to assign participants to corrals propitiates interference between participants. Further analysis confirms that the current method for predicting participants finish time is inaccurate.

This study compares three different regression-based methods to predict the race finish time of participants and the current method used in WB10K to assign runners to corrals. The finish time prediction methods can be used to assign participants to starting corrals in a way that interference between participants is minimized. Data from the 2011 and 2012 WB10K editions is used to fit regression models for the 2013 results. Then, these regression-based models are used to predict finish times for the 2014 WB10K edition. The MSE and MAD metrics are used to quantify the errors associated with the predictions for each method.

It is concluded that Method 2, which yielded a MSE of 457 and a MAD of 17, is the best alternative out of the three proposed methods, whereas the Status Quo yields a MSE of 3,202 and a MAD of 46. Method 2 outperformed the other methods when consolidating all the sub-groups, proving better than the Status Quo by 700.66% and 270.59% in the MSE and MAD metrics. When evaluating the methods for the subgroups, Method 1 and 3 yield lower MSE and MAD than Method 2 in three categories. A combined method with the best performing regression models from Methods 1, 2, and 3 further reduced the MSE and MAD of Method 2 by 28.23% and 17.65%, respectively. The Combined Method improves upon the Status Quo by 2,482.17% with respect to the MSE and 1,533.33% when considering the MAD. Either Method 2 or the Combined Method may be used to assign bib numbers to non-elite athletes in future WB10K editions.

A mathematical model was proposed in Chapter 5 to identify a lower bound for interferences between participants. The lower bound would aid in the evaluation process of corral policies. This model evaluates an ideal case were each participant may be assigned to an exclusive single corral, which in reality is not a feasible scenario. Given the number of participants in the WB10K, it is estimated that more than 100 GB of memory are needed to obtain the lower bound.

In Chapter 6, five different corral policies were evaluated. All the methods evaluated with the simulation assumed that the participants start the race from the corral assigned with the regression method. From this analyses it was observed that there was a direct relation between the number of corral and the total interferences. As the number of corrals increase the total number of interferences decreases. Policies without a wave implementation reduce participant's interferences by more than 70%. Nevertheless, the implementation of waves yield a lower expected number of interferences, with 5,006,096, an 87 % decrease in interferences. Hence, a wave implementation will require more logistic improvements and will increase the duration of the event. The WB10K may still use their existent corral policy and combined it with the regression method. In order for the implementation to be successful a control method needs to be implemented. Chapter 7 proposes that participants be disqualified if they reach the start line before the corral assigned is clear to start.

8.1 Limitations

The main limitation of this work is related to the accuracy of the data as it is provided voluntarily by participants without verification (*i.e.* age, gender, and estimates). Any inconsistencies on how names are written (including accents, initials, middle names and second last names) difficult matching participants using historical data.

8.2 Implementation

The assignment of bib numbers for elite athletes at the WB10K edition will remain unchanged. As in previous editions, the yellow corral (the first corral) is reserved for elite athletes. On the other hand, the bib number assignment for non-elite participants may be determined using the regression method described in Chapter 4.

Given the participants data (provided upon registration) for the 2014 edition, non-elite participants will be classified into two groups: those with registered times in 2011 or 2012 (i.e. returning participants), and those that do not have registered times (i.e., new participants). Then, the expected finish time will be calculated for each participant based on the corresponding regression equation. Next, non-elite participants will be sorted in ascending order, based on the computed expected finish time. The bib numbers for non-elite participants will be assigned based on the sorted expected finish time estimates so that the lowest bib number for a non-elite participant is assigned to the individual with the lowest expected finish time. The corral assignment for each non-elite participant will be made by grouping participants by bib number, depending on the size of each corral, starting with the front corrals.

If the WB10K wishes to provide bib numbers to participants as they register the top percentiles may be used as a guide line to assign participants to corrals. With this method the corrals may be delimited by the fastest runner among the corral and the fastest runner in the next corral. The participant should be assigned to the corral where his or her time estimate falls above the top percentile but is below the next corral percentile.

8.3 Future Work

Future work will focus on assigning bib numbers and corrals to participants during the 2016 WB10K edition. After the race the accuracy of the proposed regression models and the level of interference can be determined. Furthermore, the interferences may be classified as interferences between runners of the same corral and runners from different corrals. In addition, it would be interesting to study which other information can be asked to participants upon registration that would help improve the regression methods. Moreover, a survey may be conducted to quantify the effect of implementing the suggested methods and explore their content with the status quo of the event.

9. References

[1] "World's Best 10K Overview." *World's Best 10K*. AllSportCentral, AutoPistas De Puerto Rico & Puente Teodoro Moscoso. Web. 10 Jan 2014. <http://www.wb10k.com/content.cfm?contentID=68&sNavID=2&lang=esp>.

[2] "ABOUT THE IAAF." International Association of Athletics Federations - IAAF. Web. 1 Jan. 2015. <http://www.iaaf.org/about-iaaf>.

[3] "RouteMap."World'sBest10K.AllSportCentral,AutoPistasDePuertoRico&PuenteTeodoroMoscoso. Web. 19 Jan 2014. <http://www.wb10k.com/content.cfm?contentID=110&sNavID=11&lang=>.

[4] "Walt Disney World® Marathon Weekend." *Corral and Bib Assignment*. N.p.. Web. 22 Jan 2014. <http://www.rundisney.com/disneyworld-marathon/runner-info/>.

[5] "Chicago Marathon." *Start Corrals*. N.p.. Web. 22 Jan 2014. <http://www.chicagomarathon.com/participant-information/start-corrals/>.

[6] "PHILADELPHIA MARATHON." *Pre-race Email to registrants*. N.p.. Web. 20 Jan 2014. <http://philadelphiamarathon.com/for-runners/pre-race-email-to-registrants>.

[7] *"The Start.*" TCS New York City Marathon. Web. 26 May 2014. <http://www.tcsnycmarathon.org/race-day/the-start-0>.

[8] "Qualifying Standards." Boston Athletic Association. Web. 26 May 2014. <http://www.baa.org/races/boston-marathon/participant-information/qualifying/qualifying-standards.aspx>.

[9] Frederick, C.M. and Ryan, R.M. (1993), "Differences in motivation for sport and exercise and their relations with participation and mental health", Journal of Sport Behavior, Vol. 16 No. 3, pp. 124-46.

[10] Griffin, M. (2010), "Setting the scene: hailing women into a running identity", Qualitative Research in Sport and Exercise, Vol. 2 No. 2, pp. 153-74.

[11] Hallmann, Kirstin, and Pamela Wicker. "Consumer Profiles of Runners at Marathon Races." International Journal of Event and Festival Management 3.2 (2012): 171-87.

[12] Knechtle, Beat, Christoph Alexander Rüst, Thomas Rosemann, and Romuald Lepers. "Age-related Changes in 100-km Ultra-marathon Running Performance." AGE 2012.34 (2011): 1033–1045.

[13] Rodriguez, E., G. Espinosa-Paredes, and J. Alvarez-Ramirez. "Convection–diffusion Effects in Marathon Race Dynamics." Physica A 2014.393 (2014): 498–507.

[14] Alvarez-Ramirez, Jose, and Eduardo Rodriguez. "Scaling Properties of Marathon Races." Physica A 365.2006 (2006): 509–520.

[15] Sime, Jonathan D. (1999) "Crowd facilities, management and communications in disasters", Facilities, Vol. 17 Iss: 9/10, pp.313 – 324.

[16] Desmet, Antoine, and Erol Gelenbe. "Graph and Analytical Models for Emergency Evacuation." Future Internet 2013.5 (2013): 46-55.

[17] Al-Kodmany, Kheir. "Crowd Management and Urban Design: New Scientific Approaches." URBAN DESIGN International 18.4 (2013): 282–295.

[18] Usher, John M., and Strawderman, Lesley. "Emergent Crowd Behavior from the Microsimulation of Individual Pedestrians." Proceedings of the 2008 Industrial Engineering Research Conference (2008).

[19] Usher, John M., Kolstad, Eric, and Strawderman, Lesley. "Simulating Pedestrian Navigation Behavior Using a Probabilistic Model." Proceedings of the 2009 Industrial Engineering Research Conference (2009).

[20] Bekker, J., and W. Lotz. "Planning Formula One Race Strategies Using Discrete-event Simulation." Journal of the Operational Research Society 2009.60 (2008): 952-61.

[21] Atuahene, Isaac, Sawhney ,Rupy, Li, Xueping, and Aikens, Charles. "An Optimal-Model of Entry, Ticketing and Seating of Neyland Stadium" IERC. Reno, NV. May. 2011.

[22] Warpole, Ronald E., Raymond H. Myers, Sharon L. Myers, and Keying Ye. "Chapter 9: One-and-Two Sample Estimation Problems." Probability & Statistics for Engineers & Scientists. 9th ed. Boston: Prentice Hall, 2012. Print.

[23] Warpole, Ronald E., Raymond H. Myers, Sharon L. Myers, and Keying Ye. "Chapter 8: Fundamental Sampling Distributions and Data Description." Probability & Statistics for Engineers & Scientists. 9th ed. Boston: Prentice Hall, 2012. Print.

10. Appendix A: Programing Codes

```
package wb10K;
import java.awt.GridLayout;
import java.util.Arrays;
import java.util.Comparator;
import java.util.Random;
import java.io.File;
import jxl.*;
import jxl.write.Label;
import jxl.write.Number;
import jxl.write.WritableSheet;
import jxl.write.WritableWorkbook;
import javax.swing.Box;
import javax.swing.BoxLayout;
import javax.swing.JFileChooser;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JTextField;
import javax.swing.SwingConstants;

public class Main {
        //User Input
        static int runnerWidth;
        static int runnerHeight;
        static int numberOfCorrals;
        static int iterations;
        static int searchNameColumn;
        static int predictionColumn;
        static int positionTimesColumn;
        static int netTimesFirstColumn;
        static int howManyNetTimesFirstColumn;
static String sheetName;
static String sheetName2;

public static void main(String[] args) throws Exception{
        // TODO Auto-generated method stub
        //Get Default Values

        JTextField numberOfCorralsField = new JTextField(5);
        JTextField iterationsField = new JTextField(5);
        JPanel myPanel = new JPanel();
        myPanel.setLayout(new BoxLayout(myPanel, BoxLayout.Y_AXIS));
        myPanel.add(Box.createVerticalBox()); // a spacer
        myPanel.add(new JLabel("Number Of Corrals:"));
        myPanel.add(numberOfCorralsField);
        myPanel.add(Box.createHorizontalStrut(15)); // a spacer
        myPanel.add(new JLabel("Iterations:"));
        myPanel.add(iterationsField);

        int result = JOptionPane.showConfirmDialog(null, myPanel,
                        "Please Enter The Required Info", JOptionPane.OK_OPTION);
        if (result == JOptionPane.OK_OPTION) {
                numberOfCorrals = Integer.parseInt(numberOfCorralsField.getText());
                iterations = Integer.parseInt(iterationsField.getText());
        }

        //code de autogenerate jtextfield
        JTextField[][] tfs = new JTextField[numberOfCorrals][2];
        myPanel = new  JPanel( new GridLayout(numberOfCorrals,2) );

        for (int j = 0; j < tfs.length; j++) {
                tfs[j][0] = new JTextField();
                tfs[j][1] = new JTextField();
                int numberDisplay = j+1;
        myPanel.add(new JLabel("Corral-"+numberDisplay+"            Width:"));
        myPanel.add(tfs[j][0]);
        myPanel.add(new JLabel("Height: ",SwingConstants.RIGHT));
        myPanel.add(tfs[j][1]);
}

int[][] corralsDimension = new int[numberOfCorrals][2];
result = JOptionPane.showConfirmDialog(null, myPanel,
                "Please Enter The Required Info", JOptionPane.OK_OPTION);
if (result == JOptionPane.OK_OPTION) {
        for (int j = 0; j < corralsDimension.length; j++) {
                corralsDimension[j][0] = Integer.parseInt(tfs[j][0].getText());//Width
                corralsDimension[j][1] = Integer.parseInt(tfs[j][1].getText());//Height
        }
}

//search an Excel file to read from
JFileChooser fileChooser = new JFileChooser();
fileChooser.setFileSelectionMode(JFileChooser.DIRECTORIES_ONLY);
try {
        Thread.sleep(500);
}
catch(Exception e) {
}
fileChooser.showSaveDialog(null);

WritableWorkbook finalWorkbook = Workbook.createWorkbook(new
        File(fileChooser.getCurrentDirectory()+"/finalInterferenceCalculation.xls"));

WritableWorkbook positionTimesWorkbook = Workbook.createWorkbook(new
        File(fileChooser.getCurrentDirectory()+"/Java-positionTimesCorrals.xls"));

for (int iter = 0; iter < iterations; iter++) {
        int iterationsSheetNames = iter+1;
```

```
//read the workbook based on file provided
Workbook x = Workbook.getWorkbook(new
    File(fileChooser.getCurrentDirectory()+"/EstimatedTimes.xls"));
//Get sheet to read from, column from Name and column for Time Estimate
sheetName = "Sheet1";
searchNameColumn = 1;
predictionColumn = 2;
Sheet sheet = x.getSheet(0);


final String[][] estimatedTimes = new
    String[sheet.getColumn(searchNameColumn).length-1][2];
final String[][] randomizedEstimatedTimes = new
    String[sheet.getColumn(searchNameColumn).length-1][3];


for(int i = 1; i < sheet.getColumn(searchNameColumn).length; i++ ){
    estimatedTimes[i-1][0]= sheet.getCell(searchNameColumn,i).getContents();
    estimatedTimes[i-1][1] = sheet.getCell(predictionColumn,i).getContents();
}


Arrays.sort(estimatedTimes, new Comparator<String[]>() {
    @Override
    public int compare(final String[] entry1, final String[] entry2) {

        Double time1 = 0.00;
        Double time2 = 0.00;
        if (entry1[1] != null && entry1[1].length() > 0) {
            try {
                time1 = Double.parseDouble(entry1[1]);
            } catch(Exception e) {
                time1 = -1.00;
            }
        }
        if (entry2[1] != null && entry2[1].length() > 0) {
            try {
                time2 = Double.parseDouble(entry2[1]);
            } catch(Exception e) {
                time2 = -1.00;
            }
        }

        return time1.compareTo(time2);
    }
});

WritableWorkbook workbook = Workbook.createWorkbook(new
    File(fileChooser.getCurrentDirectory()+"/Java-SortedEstimatedTimes.xls"));
WritableSheet sheet0 = workbook.createSheet("Sorted", 0);

for ( int i = 0; i < estimatedTimes.length; i++){
    sheet0.addCell(new Label(0,i,estimatedTimes[i][0]));
    sheet0.addCell(new Label(1,i,estimatedTimes[i][1]));
}

workbook.write();
workbook.close();
workbook = Workbook.createWorkbook(new
    File(fileChooser.getCurrentDirectory()+"/Java-RandomEstimatedTimes.xls"));
sheet0 = workbook.createSheet("random", 0);


//get runnerHeight and Runner Width and save them to two dimension array.

int nextfloorRunnerWidth = 0;
int previousceilingRunnerWidth = 0;
int totalCorralSize = 0;
outerloop:
    for ( int i = 0; i < numberOfCorrals; i++){
```

```
totalCorralSize += corralsDimension[i][0]*corralsDimension[i][1];

int floorRunnerWidth = nextfloorRunnerWidth ;

int ceilingRunnerWidth =
            corralsDimension[i][0]*corralsDimension[i][1]
                        +nextfloorRunnerWidth;

nextfloorRunnerWidth = ceilingRunnerWidth;


String[][] randomizer = new
            String[corralsDimension[i][0]*corralsDimension[i][1]][2];


if(ceilingRunnerWidth > estimatedTimes.length-1){

            ceilingRunnerWidth = estimatedTimes.length;


            if (floorRunnerWidth >= estimatedTimes.length ) {

                        System.out.println("Breaking");

                        break outerloop;

            }

            randomizer = new String[estimatedTimes.length-
                        floorRunnerWidth][2];

}


int randomizerLocation = 0;

for(int j = floorRunnerWidth; j< ceilingRunnerWidth; j++){

            randomizer[randomizerLocation] = estimatedTimes[j];

            randomizerLocation++;

}


shuffle(randomizer);


int sheetCellLocation = 0;

for(int a = floorRunnerWidth; a< ceilingRunnerWidth; a++){


            sheet0.addCell(new
                        Label(0,a,randomizer[sheetCellLocation][0]));

            sheet0.addCell(new
Label(1,a,randomizer[sheetCellLocation][1]));
```

```
                        randomizedEstimatedTimes[a] =
randomizer[sheetCellLocation];
                        sheetCellLocation++;
                    }


                }



            workbook.write();
            workbook.close();


            //search an Excel file to read from


            //read the workbook based on file provided
            x = Workbook.getWorkbook(new
File(fileChooser.getCurrentDirectory()+"/PositionTimes.xls"));
            //Get sheet to read from, column from Name and column for Time Estimate


            sheetName = "Position Times";
            positionTimesColumn = 3;


            sheet = x.getSheet(0);
            int positionTimesColumnLenght = sheet.getColumn(positionTimesColumn).length;


            sheet0 = positionTimesWorkbook.createSheet("Corrals-"+iterationsSheetNames, iter);


            int positionPLuPLus = 1; //use to move in the single column of position times
            final String[] singleRowPositionTimesArray = new String[totalCorralSize]; //In the last
position save the estimatedTimes and the positionTimes
            int nextfloorRunnerHeight = 0;
            outerloop:


                for ( int n = 0; n < numberOfCorrals; n++){
                    sheet0.addCell(new Label(0,nextfloorRunnerHeight,"Corral: "+(n+1)));


                    for(int i = 0; i < corralsDimension[n][1]; i++ ){ //height

                        for(int j = 0; j < corralsDimension[n][0]; j++ ){ //width

                            if (positionPLuPLus > positionTimesColumnLenght || positionPLuPLus > randomizedEstimatedTimes.length) {
                                System.out.println("Breaking");
                                break outerloop;
                            }


                            singleRowPositionTimesArray[positionPLuPLus-1] = sheet.getCell(positionTimesColumn, positionPLuPLus).getContents();
                            sheet0.addCell(new Label(j+1,i + nextfloorRunnerHeight,sheet.getCell(positionTimesColumn,positionPLuPLus).getContents()));
                            sheet0.addCell(new Label(j+corralsDimension[n][0]+2,i + nextfloorRunnerHeight,randomizedEstimatedTimes[positionPLuPLus-1][1]));


                            positionPLuPLus++;
                        }
                    }


                    nextfloorRunnerHeight += corralsDimension[n][1];


                }



            //get Results 2014 sheet
            //pair the vlookup name with the randomizedEstimatedTimes
            //search an Excel file to read from
```

```
                        //read the workbook based on file provided

                        x = Workbook.getWorkbook(new
File(fileChooser.getCurrentDirectory()+"/Results.xls"));

                        //Get sheet to read from, column from Name and column for Time Estimate



                        sheetName = "Results";

                        searchNameColumn = 5;

                        netTimesFirstColumn = 35;

                        howManyNetTimesFirstColumn = 3;


                        sheet = x.getSheet(0);


                        workbook = Workbook.createWorkbook(new
File(fileChooser.getCurrentDirectory()+"/Java-FinalNetTimes.xls"));

                        sheet0 = workbook.createSheet("NetTimes", 0);

                        sheet0.addCell(new Label(0,0,"VLookUp")); //Corral 0

                        sheet0.addCell(new Label(1,0,"Estimated Times")); //Corral 0

                        sheet0.addCell(new Label(2,0,"Corral to 0K")); //Corral 0

                        sheet0.addCell(new Label(3,0,"Other Checkpoints")); //Corral 0



                        for ( int n = 0; n < singleRowPositionTimesArray.length && n <
randomizedEstimatedTimes.length; n++){


                             int netTimesRow =
searchSheet(randomizedEstimatedTimes[n][0],sheet,searchNameColumn);


                             sheet0.addCell(new Label(0,n+1,randomizedEstimatedTimes[n][0])); //Name

                             sheet0.addCell(new Label(1,n+1,randomizedEstimatedTimes[n][1]));
//EstimatedTime

                             sheet0.addCell(new Label(2,n+1,singleRowPositionTimesArray[n])); //Corral 0


                             if(netTimesRow == -1){
                }
                else{
                    for ( int m = 1; m <= howManyNetTimesFirstColumn; m++){
                        double netColumnValue =
Double.parseDouble(sheet.getCell(netTimesFirstColumn+m,netTimesRow).getContents());
                        double positionTimeValue =
Double.parseDouble(singleRowPositionTimesArray[n]);
                        double positionplusnetcolumn = netColumnValue +
positionTimeValue;
                        sheet0.addCell(new Number(m+2,n+1,
positionplusnetcolumn)); //Corral 0
                    }


                }





            }


            workbook.write();
            workbook.close();


            /// Final Counter Code


            //read the workbook based on file provided
            x = Workbook.getWorkbook(new File(fileChooser.getCurrentDirectory()+"/Java-
FinalNetTimes.xls"));
            searchNameColumn = 0;
            netTimesFirstColumn = 2;
            //Get sheet to read from, column from Name and column for Time Estimate


            sheet = x.getSheet(0);
```

```
sheet0 = finalWorkbook.createSheet("Interference-"+iterationsSheetNames, iter);
sheet0.addCell(new Label(0, 0, "VLookUp"));
sheet0.addCell(new Label(1, 0, "0K-3K: Passes"));
sheet0.addCell(new Label(2, 0, "3K-8K: Passes"));
sheet0.addCell(new Label(3, 0, "8K-10K: Passes"));
sheet0.addCell(new Label(5, 0, "0K-3K: Passed by"));
sheet0.addCell(new Label(6, 0, "3K-8K: Passed by"));
sheet0.addCell(new Label(7, 0, "8K-10K: Passed by"));

int totalRunners = sheet.getColumn(0).length;
Double tki = 0.0;  // tiempo del corredor i en el km k
Double tkj = 0.0;  // tiempo del corredor j en el km k
Double tki1 = 0.0; // t_k+1 i el siguiente checkpoint del corredor i
Double tkj1 = 0.0; // t_k+1 j el siguiente checkpoint del corredor j
int Cijk = 0;      // corredor j le paso a corredor i en el km k
String Data ="";   // ????
// Dim r As Range 'Rango del array
int Cm = 0;         // Pases totales
int Cp = 0;

System.out.println("Counter");
for(int k= 0; k < howManyNetTimesFirstColumn; k++){ //For k = 1 To x

        for(int i=1;i<totalRunners;i++){ //For i = 1 To y
                Cm = 0;         // Pases totales
                Cp = 0;
                for(int j=1;j<totalRunners;j++) { //For j = 1 To y
                        if(i==j){}//If i = j Then
                        else{

                                //Cijk = 0;
        if(!sheet.getCell(netTimesFirstColumn+k,i).getContents().isEmpty() &&
!sheet.getCell(netTimesFirstColumn+k,j).getContents().isEmpty() &&
!sheet.getCell(netTimesFirstColumn+k+1,i).getContents().isEmpty() &&
!sheet.getCell(netTimesFirstColumn+k+1,j).getContents().isEmpty()){


                                        tki =
Double.parseDouble(sheet.getCell(netTimesFirstColumn+k,i).getContents()); //Cells(1 + i, 1 + k).Value //1+i dado
que el titulo ocupa el lugar uno, lo mimo ocurre en k

                                        tkj =
Double.parseDouble(sheet.getCell(netTimesFirstColumn+k,j).getContents());//Cells(1 + j, 1 + k).Value

                                        tki1 =
Double.parseDouble(sheet.getCell(netTimesFirstColumn+k+1,i).getContents());//Cells(1 + i, 2 + k).Value

                                        tkj1 =
Double.parseDouble(sheet.getCell(netTimesFirstColumn+k+1,j).getContents());//Cells(1 + j, 2 + k).Value //tengo que
poner otro if para saber si tiene dato o no el k+1


                                        //Este Cp y Cm tiene que ser por cada i, y
imprimirlo a un nuevo sheet

                                        if(tki<tkj && tki1 > tkj1){//If t_ki < t_kj Then
//Si corredor i llega primero que j al checkpoint inicial


                                            Cm++;


                                        }


                                        if(tki > tkj && tki1 < tkj1){//If t_ki < t_kj
Then //Si corredor i llega primero que j al checkpoint inicial


                                            Cp++;


                                        }

                                    }

                                }//End iIF

                            } //Next j


                            if(k==0){

                                sheet0.addCell(new Label(0, i,
sheet.getCell(searchNameColumn,i).getContents()));
```

```
                    }
                    sheet0.addCell(new Number((1+k), i, Cp ));
                    sheet0.addCell(new  Number((2+k)+howManyNetTimesFirstColumn,  i, Cm ));



                }// Next i

            }//Next k

            System.out.println("FIN");


        }
        positionTimesWorkbook.write();
        positionTimesWorkbook.close();
        finalWorkbook.write();
        finalWorkbook.close();




    }
    //searches the sheet for the given item
    //returns null if item isn't found
    //returns cell adjacent to item if found
    public static int searchSheet(String item, Sheet sheet, int Column){
        int i;
        int result = -1;
        for ( i = 0; i < sheet.getRows(); i++){
            if(sheet.getCell(Column,i).getContents().equals(item)){
                result = i;//if return is placed here, will stop after first instance of the searched item
            }
        }
        //      System.out.println("Searched: "+ i + " items for " + item);
        return result;
    }
    static void shuffle(String[][] a) {
        Random random = new Random();

        for (int i = a.length - 1; i > 0; i--) {
            int m = random.nextInt(i + 1);

            String[] temp = a[i];
            a[i] = a[m];
            a[m] = temp;
        }

    }


}
```