Determining the impact of training load on distance running performance using machine learning techniques

Bongani Babeli & Tendai Musendo

Report presented in partial fulfilment
of the requirements for the degree of
BComHons **(Statistics**)
at the University of Stellenbosch

Supervisor: Dr Morne Lamont

Degree of confidentiality: A

November 2021

PLAGIARISM DECLARATION

1. Plagiarism is the use of ideas, material and other intellectual property of another's work and to present it as my own.
2. I agree that plagiarism is a punishable offence because it constitutes theft.
3. I also understand that direct translations are plagiarism.
4. Accordingly, all quotations and contributions from any source whatsoever (including the internet) have been cited fully. I understand that the reproduction of text without quotation marks (even when the source is cited) is plagiarism.
5. I declare that the work contained in this assignment, except otherwise stated, is my original work and that I have not previously (in its entirety or in part) submitted it for grading in this module/assignment or another module/assignment.

Student number: **22121099 &** **20941714**	**Signature** B.Babeli. T.P.Musendo
Initials and surname **B.E. Babeli &** **T.P. Musendo**	**Date: 05 November 2021**

Copyright © 2021 Stellenbosch University
All rights reserved

Acknowledgements

We would like to pass our heartfelt gratitude and appreciation to our Supervisor Dr Morne Larmont for being with us throughout the research, the support and patience he had for us especially during confusion times. We are also grateful for the time and knowledge he spared for us in his tight schedule and most importantly, the genuine and very helpful guidance we got from him that helped us produce this piece of work.

We would also like to express our appreciation to the department of statistics and actuarial science for giving us this opportunity to conduct this research under their umbrella, we are grateful for the valuable and insightful knowledge we received throughout our study years. Lastly, we would like to thank our families for giving us endless support, our friends and fellow Statistics classmates for their presence and their unconditional support, we are grateful indeed.

Abstract

Marathon runners are becoming more reliant on expert advice as well as anecdotal advice when constructing their training programs. Athletes also rely on anecdotal advice from their coaches and peers. However, this advice is not necessarily backed up by scientific investigations, and as a consequence, it is not always trustworthy. Scientific studies enable athletes and coaches to use an ‘evidence’ based criteria when planning their training programs. For this reason, a scientific approach to sports performance has become increasingly popular. The establishment of a training scheme that improves physiological qualities with the goal of enhancing athletic performance is one of the most essential jobs of a fitness coach. To attain this purpose, coaches design training programs that effectively control training duration, intensity and volume over the course of the year. Coaches can improve sports performance by quantifying training loads and reactions. The purpose of this report is to give statistically based insight on the effects that training load has on the performance of marathon runners. This will assist coaches and runners to improve their training regimes in order to achieve their performance goals. Different regression modelling techniques are utilised and a comparison of their performance measures is done by examining the MAE and RMSE. The results of the variable selection and model performance give an indication as to which training load factors affect performance and by how much. A combination of R, SPSS®, SAS® are used to do the various statistical computations whose coding and output is included in the APPENDIX.

Key words:

Regression; Modelling, Training Load, Athletes, Random Forest, LASSO, Elastic-Net, Performance Measures

Table of Contents

List of Tables

List of Figures

List of appendices

List of abbreviations and/or acronyms

ANOVA	Analysis of variance
ATP	Adenosine triphosphate
IQR	Inter quartile range
LASSO	Least absolute shrinkage and selection operator
LQ	Lower quartile range
MAE	Mean absolute error
MANOVA	Multivariate analysis of variance
OLS	Ordinary least squares
RMSE	Root mean square error
RSS	Residual sum of squares
UQR	Upper quartile range
$\dot{V}O_{2max}$	Velocity at which maximal oxygen uptake

CHAPTER 1
INTRODUCTION

The focus of this research project is to use statistical analysis to investigate how training load affects the performance of marathon runners. The main tool for this analysis will be multiple regression. Multiple regression is a technique that has been developed to investigate the relationship between a continuous response variable and a number of predictor variables (Pallant, 2010).The details about the variables and their interactions will be explored in the chapters to follow.

An exploratory analysis is undertaken to investigate the distribution of the raw data. The analysis starts by discussing how a questionnaire along with collected athlete records were used to construct the dataset. Followed by an explanation of the stages and steps taken to clean the data in order to achieve the final dataset that was used for the analysis. Performance of marathon runners is determined by many different factors, which include genetics, age, gender, nutrition, training regimes. Athletes alter and tailor the intensity and length of their training sessions to achieve their performance goals. The Medical Research Council (MRC) recommended that the important variables to assess the effect of training load on marathon runners include the following: The number of years an individual has been a recreational/distance runner, the number of times they trained during the weeks leading to the marathon; the distance they trained for and their training pace during these sessions. However, these are just a handful of the variables available for analysis but provide a solid starting point. An analysis of these variables as well as other variables will be done in the exploratory analysis. A general correlation analysis using variance-covariance matrices and scatterplots will be used to gain insight on the interactions between variables. This will provide insight that will be used to help select an appropriate regression technique and interpret differences in the models produced by these different regression techniques.

In statistics, there are different regression techniques that have been fine-tuned to deal with various problems that may arise from the interaction of predictor variables. The different regression techniques have their respective advantages and disadvantages. Each model will be assessed for significance according to the aim of the research. A critical comparison of the different models produced, considering the difference in computing methods. Before any recommendations can be made, the final model needs to be tested. This process will include splitting the data into two datasets: a training dataset and testing data set. The models are the fitted on the training and testing data and use several performance metrics to evaluate the performance of the models generated by the different regression techniques. This will be the final stage of the analysis, after which a conclusion will be drawn and recommendations made to athletes and coaches.

1.1 LITERATURE REVIEW

The goal of this project is to gain statistical insight on how training load affects an athlete's performance. The results from the analysis will be evaluated and recommendations will be given to athletes and coaches that will hopefully improve their training programs and subsequently achieve better marathon performance. This literature review examines the training load factors that influence the performance of athletes. Additionally, it illustrates how the separate components of training load affect performance as well as to what extent these factors affect performance. Many scientists investigated and showed that about 50% of the variation of the training response was due to heredity (Hopkins, 2001). Other academics criticized their findings and methodology of their findings.

The focus of this literature review is to investigate prior findings and synthesize them with other findings to get a solid starting point for this research project that attempts to answer the question – "Does training load affect performance?" There are several methods of quantifying training load as well as athletic performance. Smith (2003) developed a framework to try and understand the training process associated with elite athletic performance and he suggests that athletes should prepare themselves by going through a training regimen with the physiological goal of improving an athlete's performance. The method of training entails doing the same exercises over and over again to automate the execution of skills and to establish anatomical and metabolic functions that contribute to improved physical performance.

Overload training includes subjecting an athlete to higher levels of stress than they have previously endured in order to stimulate adaptation and supercompensation (Smith, 2003). When overload training and recuperation are properly balanced, supercompensation occurs, resulting in a performance surge (Budgett, 1998).

When looking at factors associated with sport performance, Smith (2003) further argues that an athlete's physical capacity, athletic shape, and sport skills will grow through many years of organized systematic training, culminating in a high level of performance with planned peaking. In this research, one of the variables in question is the number of years an athlete has been a recreational or distance runner. The research will investigate how that variable affects the finish time and race pace of the athletes. Training load (characterized by the frequency, duration, and intensity of training) fluctuates and should steadily rise during the training plan in accordance with workout-induced adaptation of various physical components (Kallus, 1992). This is in conjunction with the independent variables selected for this research where marathon runners were asked through a questionnaire, during the last 12 months leading up to the marathon: (1) Their average weekly training frequency (frequency); (2) Their average training distance in kilometres per week (duration) and (3) Their average training pace speed(intensity). The intensity of training is a qualitative factor that is determined by the type of activities completed in a given amount of time. In the questionnaire, runners gave an indication of the amount of time dedicated to different types of training, namely; the amount of time spent training the treadmill, training on tar/brick road and gravel. These three types of training provide a varied level of intensity and this research will examine if these variables have an impact on performance and how much the influence performance.

The maximum rate of oxygen the body can utilise during exercise is known as $\dot{V}O_{2max}$ which is a commonly accepted measure of intensity (Bouchard *et al.*, 1995). The usage of oxygen, which is a critical component of the respiratory process, is required for breathing. The lungs take in oxygen and transform it to adenosine triphosphate (ATP), a form of energy. VO_{2max} is directly proportional to amount of oxygen the body can absorb, and the higher the VO_{2max}, the more efficiently oxygen can be utilized to oxygen to generate ATP energy. This will lead to the body being able to better handle any sort of aerobic activity (Bacon *et al.*, 2013). Although high-intensity training has improved VO_{2max} in distance runners (Mikesell & Dudley, 1984), significant improvements in performance have been observed in athletes that did not increase their $\dot{V}O_{2max}$ (Martin *et al.*, 1986). Why could this be the case? Smith (2003) attempts to explain this with reference to an article by Hopkins (2001) that up to half of the difference in performance between individuals is assumed to be due to genetic factors, with the other half being due to training. Continuously increasing training volume is one of the most important aspects of modern training, especially for aerobic sports (Bompa & Buzzichelli, 2019) and concurrently (Foster *et al.*, 1996) established that training volume to have a substantial impact on performance in recreational runners.

1.2 CONCLUSION

The limits of human performance are constantly being pushed, thanks to consistent training throughout the year and a continual improvement knowledge of training methodology, among other things. The improvement of performance is accomplished by a systemic modification in training that enhances training adaptation and prevents overtraining through the use of appropriate loads and suitable rest times. Comprehensive athlete monitoring is required, especially during periods of training overload, in order for a coach to make informed decisions about the consequences and subsequent training plans. Ron Trent, a reviewer of the article titled ‘Genes and Training for Athletic Performance’ by Hopkins (2001), made the following comment, “I don’t think that there is any doubt that genes contribute to performance. You have cited evidence for 50%. I am not sure I believe 50%, but 50% is in the right ballpark. In the next decade, the exact contribution will be known using technology that is coming on track through the Human Genome Project." Although some academics may contradict each other's' findings and some may coincide, the progress made in this topic by the various academics and scientists have come a long way into giving insight on the question at hand, that is - How does training load influence athletic performance? This research will attempt to gain further insight on this question whilst baring all these previous findings in mind.

The layout of the study is a as follows. An exploratory data analysis and multivariate analysis of variance will be conducted in Chapters 2 & 3 respectively to investigate whether there are any significant differences in the response variables (Finish Time and Race Pace) for the different categorical variables (Age and Gender). The goal here is to see whether the data needs to split up further into the various age categories, for example, before the modelling stage. If the means differ significantly then the data will have to split up to avoid incorrect statistical inferencing. Chapters 4 & 5 will focus on the regression modelling and variable selection respectively. The modelling includes 3 statistical learning methods: LASSO, Elastic-net and Random Forest methods. The performance of each of these modelling techniques will be assessed using the MAE and RMSE. Theses performance measures, in conjunction with the results from the Variable Importance Plot are used to evaluate which predictor variables are best at predicting the response variable. Chapter 6 synthesizes is a summary of the results the research where the main goals of the study are revised using the scientific evidence to give coaches and athletes recommendations on how training load factors affect performance. Furthermore, recommendations will also be given to those who plan do to similar studies to this one and how they can improve on this research report.

CHAPTER 2
EXPLORATORY DATA ANALYSIS

2.1 INTRODUCTION

Data collection steps form the basis of data creation while sample sizes and summary statistics are important in showing the nature of collected data however, due to the nature of the study, data cleaning is equally essential to avoid misleading and wrong conclusions. These steps will be discussed in detail in this Chapter. The first step in determining if there is a relationship between the response and the predictor variables is to have an insight of the distribution of the data and how it behaves in respect to the response variables. Graphical representations of the data are therefore important in the analysis of the data and will be help in understanding existing patterns overall behavior of the data.

2.2 DATA CREATION

Figure A.1 is a flow diagram that illustrates the data collection procedure for before, during and after the marathon race. An online entry system is made available for runners who are interested in the race, these runners fill in their details on the system. When interested runners are done with entering their details, the following step is a pre-screening questionnaire on an alternate website. Interested participants who pass are deemed fit and are not prone to any risk or illness are redirected to the entry webpage where finalization of the entry process is made and the final payment is thus made. As for interested runners who are deemed high risk and prone to illness, a further educational intervention is made that assess the level of risk and reach a verdict of whether they can participate in the race or not. During race day, environmental factors are assessed if they are conducive for the race to take place or not, if conducive then the race continues otherwise the race is stopped. The medical team is also present during the race day for health emergencies before, during and after the race. Entrants start the race and some do not, race entrants who do not start the race may be led to taking this decision based on multiple factors such as their well-being, for runners who started the race, some of these runners are able to finish the race while some are not able to due to conditions such runner health conditions nonetheless, for runners who did not finish the race due to medical encounters, a medical intervention is made to assess if the underlying medical emergency is due to injury or illness. Some of the runners who finished the race may also have medical encounters that may be caused by injury or illness. For all runners that started the race and finished, race performance is recorded which is their finish time. Post-race day, a follow up email is sent to all race participants to capture any potential medical encounters that were not reported during race day.

2.3 SAMPLE SIZES

There are two race types, runners who participate in the half marathon race type cover a running distance of 21.1 km while runners in the ultra-race type cover a running distance of 56 km as seen on Table 2.1. The dataset consists of 84 variables and *106 743* observations. Table 2.1 is a summary of the numbers of runners who gave consent of participating in the study by gender, race type and age together with their proportions. Table A.2, A.3 and A.4 are summaries of runners who entered the race and those who did not participate in the study together with summaries of those who finished and those who did not finish the race.

Table 2.1: Summary of study participants who entered the race by age, gender and race type.

	All Race entrants		Study Participants	
Race type	**N**	**%**	**N**	**%**
21.1 km	**64740**	**60,65035**	**47069**	**61,40449**
56 km	**42003**	**39,34965**	**29585**	**38,59551**
Total	**106743**		**76654**	
Sex				
Males	**61815**	**57,91012**	**44042**	**57,45558**
Females	**44928**	**42,08988**	**32612**	**42,54442**
Age Groups				
<=30	**27710**	**25,95955**	**20168**	**26,31043**
31-40 years	**35049**	**32,83494**	**25045**	**32,67279**
41-50 years	**26964**	**25,26067**	**19340**	**25,23026**
>=50 years	**17020**	**15,94484**	**12101**	**15,78652**

Variables that are not related to training load are discarded for the analysis. Training load variables are chosen based on their relevance. Some runners who participated in the marathon have been running for recreational purposes for some time, this factor fits well with the training load definition given in Chapter 1. The number of years in distance running can also be classified as a training load factor for runners therefore is considered in the study. The training amount per week for runners defines the training amount an individual is engaged with per week, the representation of training frequency per week can also be described as training load. Training distance is simply the training distance in kilometers covered per week by runners for training purposes, this variable is directly related to training load an individual engages with.

Training pace represents an individual's pace in training, this variable can also be described as speed/power and individual attains while training and this variable well describes training load.

Some runners spend some of their time training on a treadmill, some on gravel road while some train on tar trails, these three variables also describe an individual's training load and are included as training load variables. Finish times in minutes for runners in the race is used as the response variable and race paces of runners recorded in minutes is the second response variable in the analysis, these two response variables are analyzed on separate basis. Race pace in minutes is the second response variable that defines performance, race pace is measured in minutes and just like finish time, it is investigated within different training load variables. The dataset is divided by race type for analysis purposes, half marathon runners have a sperate analysis form the ultra-marathon runners. The runners are also grouped by four age categories however these age categories are not separated during deeper analysis because that is not the main focus of the study. Table A.1 shows the summary statistics for the training load variables.

2.4 DATA CLEANING

During data collection, there can be a chunk of unnecessary recorded data that can be cause by recording errors, duplicate values and missing values. The abovementioned errors may cause several problems throughout the analysis and may also lead to misleading results and wrong conclusions. It is therefore vital to deal with unnecessary data to avoid working with unclean data and having impaired results. In this section, a thorough inspection of the data will be made to detect any irrelevant data observations and variables. After detecting flagged data observations and variables, data cleaning will be conducted to remove flagged data. A final inspection verifies that outliers are removed see Figure A.2.

2.4.1 Inspection

Table 2.4 give evidence that there is a significantly large amount of missing values and it is crucial to deal with such data anomalies before any further analysis can be carried out in order to avoid limited results. All detected anomalies in the data are dealt with in Section 2.4.2 where cleaning and verification of the data is discussed in detail. Multicollinearity is when there is a significant correlation between predictor variables, this can impair the prediction accuracy of the model and lead to models that produce misleading results if not dealt with. From Figure 2.1. the correlation matrix of training load variables, it is evident that number of years in recreational running and distance running are highly correlated $\rho = 0.86$. Multicollinearity may cause models with unstable coefficient estimates hence hard to interpret. Multicollinearity can also cause overfitting which is not attractive when a new data is introduced to test the model.

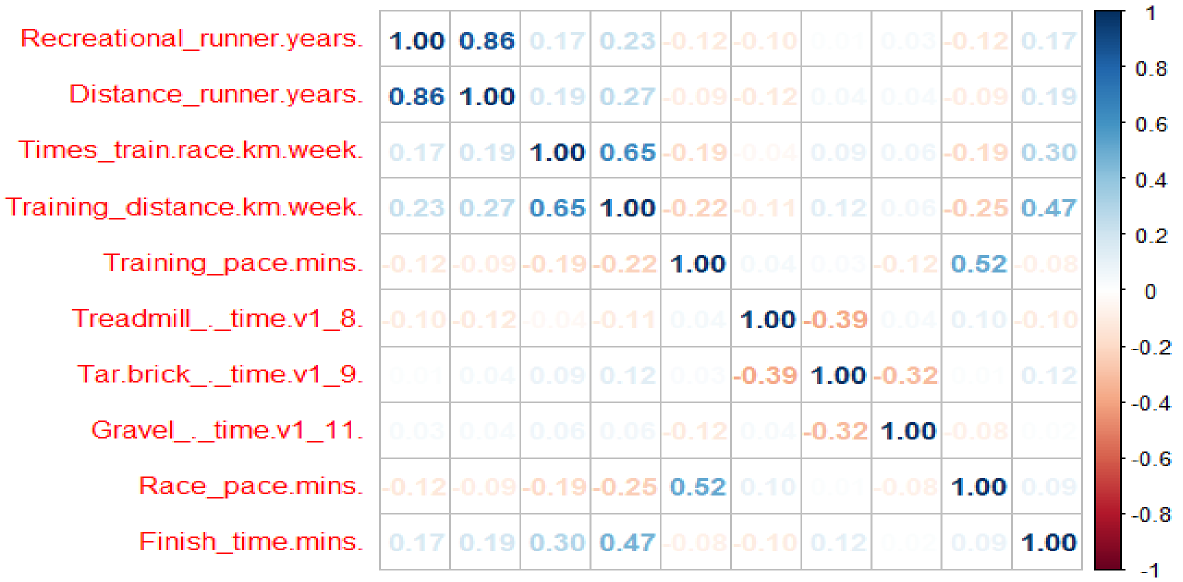


Figure 2.1: Scatterplot matrix of training load variables with a correlation coefficient scale of range of -1 to 1.

Table 2.3: summary of missing values for training load variables. The right column “Not asked” represents the number of observations where participants were not asked to fill in responses.

Summary of missing values		
Training load variable	Number of missing values	Not asked
Years of Recreational running	215	0
Years of Distance running	215	0
Times train race (km/week)	5178	0
Training distance (km/week)	5178	0
Training pace(mins)	583	0
Treadmill time	388	14890
Tar/brick time	394	14890
Gravel time	383	14890
Race pace(mins)	9174	0
Finish time(mins)	9842	0

Outliers can also be critical in the analysis and taking them out is an important step. Figure 2.2 is a visualization of the data in pursuit of detecting outliers utilizing boxplots. It is evident that there are present outliers in the data hence the inner boxplots may display misleading representations of the true distribution of the data.

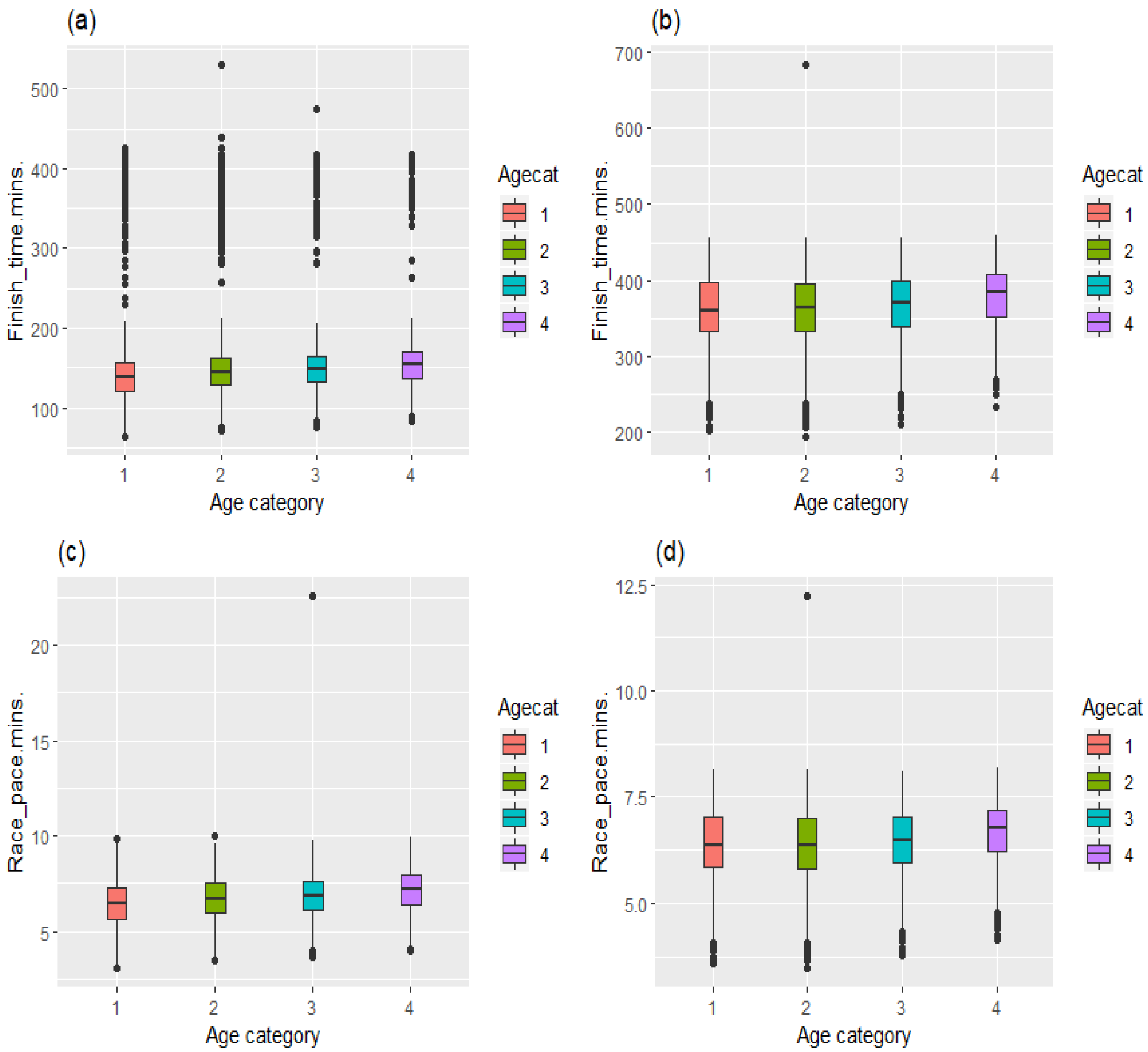


Figure 2.2: Boxplots of age categories finish time and race pace prior to removal of outliers. Panel (a) represents the half-marathon race with finish time response variable, panel (b) represents the ultra-marathon race and finish time response variable, panel (c) shows the half-marathon race with race pace response variable while panel (d) shows the ultra-marathon race and finish time response variable. The orange, green, blue and purple colours represent age categories 1,2,3 and 4 respectively plotted against finish time and race pace in minutes.

2.4.2 Cleaning and verifying

In the previous section, a summary of missing values was discussed and it consist of a huge portion however available data is still sufficient to produce a sensible analysis therefore missing values were taken out throughout the analysis. Another method of dealing with missing values apart from dropping them is imputing them, this method can be cumbersome because it may require different approaches such as the mean, mode, median, maximum and minimum for different variables. Fortunately, the missForest R Software package can do the whole imputation. Outliers were detected and removed using the following procedure; all data points that fall below $LQ - 1.5 * IQR$ and those that lie above $UQ + 1.5 * IQR$ are deemed as outliers and are therefore removed from the dataset. Figure 8.2 is a visualization of the data through boxplots after removing outliers, the boxplots verify that a huge portion of outliers is dropped.

There are some runners who participated more than once in the marathon racing, these runners are identified by entries of duplicate personal codes. For runners who participated more than once in the marathon race, the average is taken for all participated events while the other observations are dropped hence new response; finish time and race pace are created. The dropping of duplicate values is another significant factor that led to the drop of observations in the final dataset. During 2012, the following 3 variables; percentage time spent training on treadmill, percentage time spent training on tar/brick and percentage time spent training on gravel roads do not have values for the year 2012 because runners were not asked to give a response on them therefore since these variables are deemed important and used in the analysis, all 2012 observations for every variable in the dataset are dropped.

There are 38258 observations in the dataset after the above-mentioned data cleaning is done. Even though number of years in recreational running variable and number of years in distance running variables are highly correlated, the safest option is to drop one of the variables however, the underlying problem is that we don’t know which variable to drop between the two based on their level of importance therefore at this point there is not enough information at hand to help determine which one to drop. Fortunately, the fitted regression models have interesting properties of feature selection even for correlated variables therefore these models automatically handle this problem. The variable selection by the random forest regression in the variable selection section will help in determining the level of importance between these two variables while none of them will be dropped.

2.5 EXPLORATORY DATA ANALYSIS

Exploratory data analysis mainly focuses on exploring the general behavior of the explanatory/predictor variables and the response variable, more generally how they are related. The main idea is to see if there is a relationship between performance and training load variables. A few crucial variables that we believe accurately represent training load will be investigated in this section with graphical representations. The two response variables will also be explored.

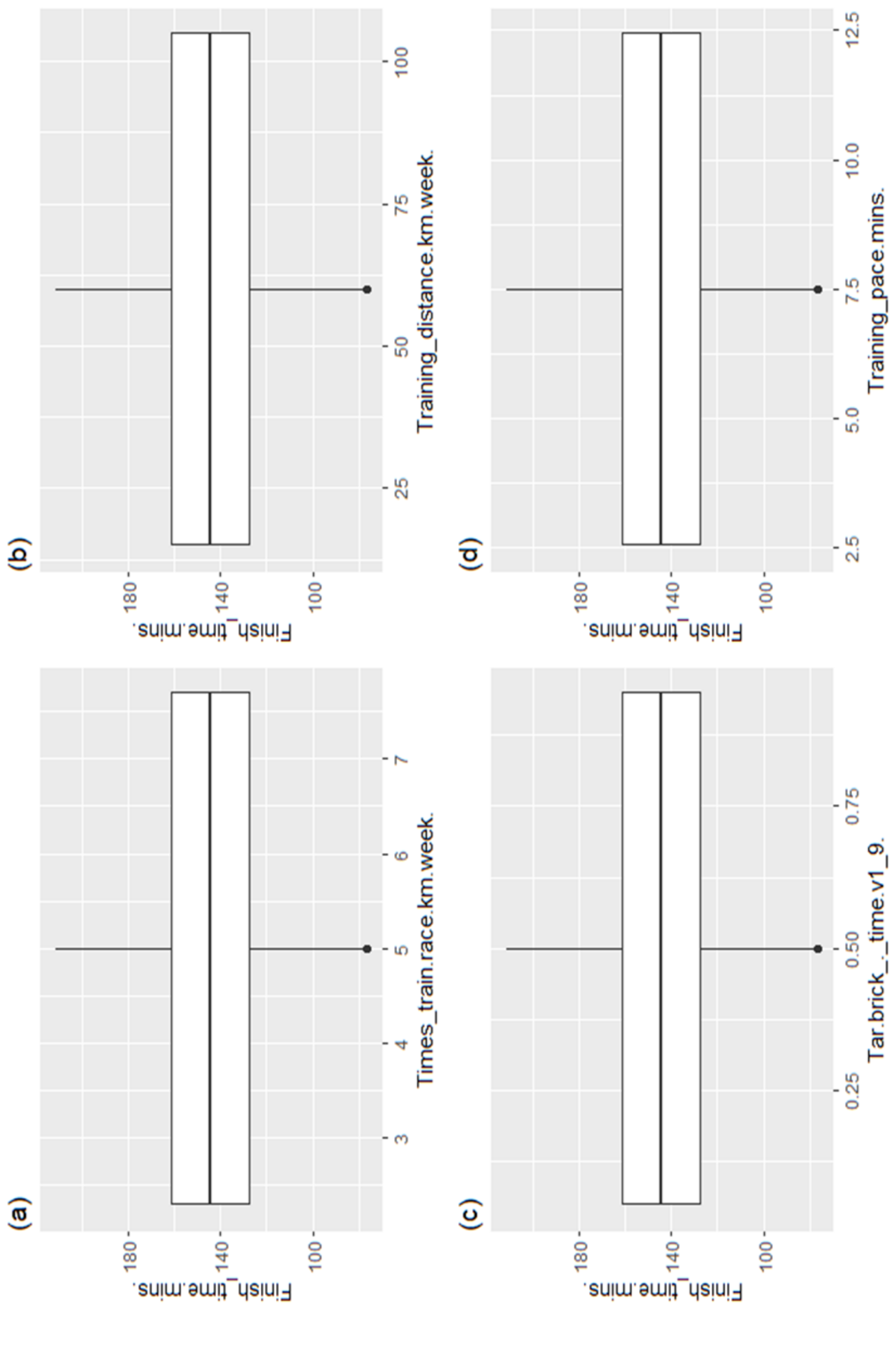


Figure 2.3: Boxplots for half marathon race type. Panel (a) is a boxplot of finish time and times train race per week, panel (b) is a boxplot of finish time and training distance per week, panel (c) is a boxplot of finish time and tar/brick percentage training time and panel (d) is a boxplot of finish time

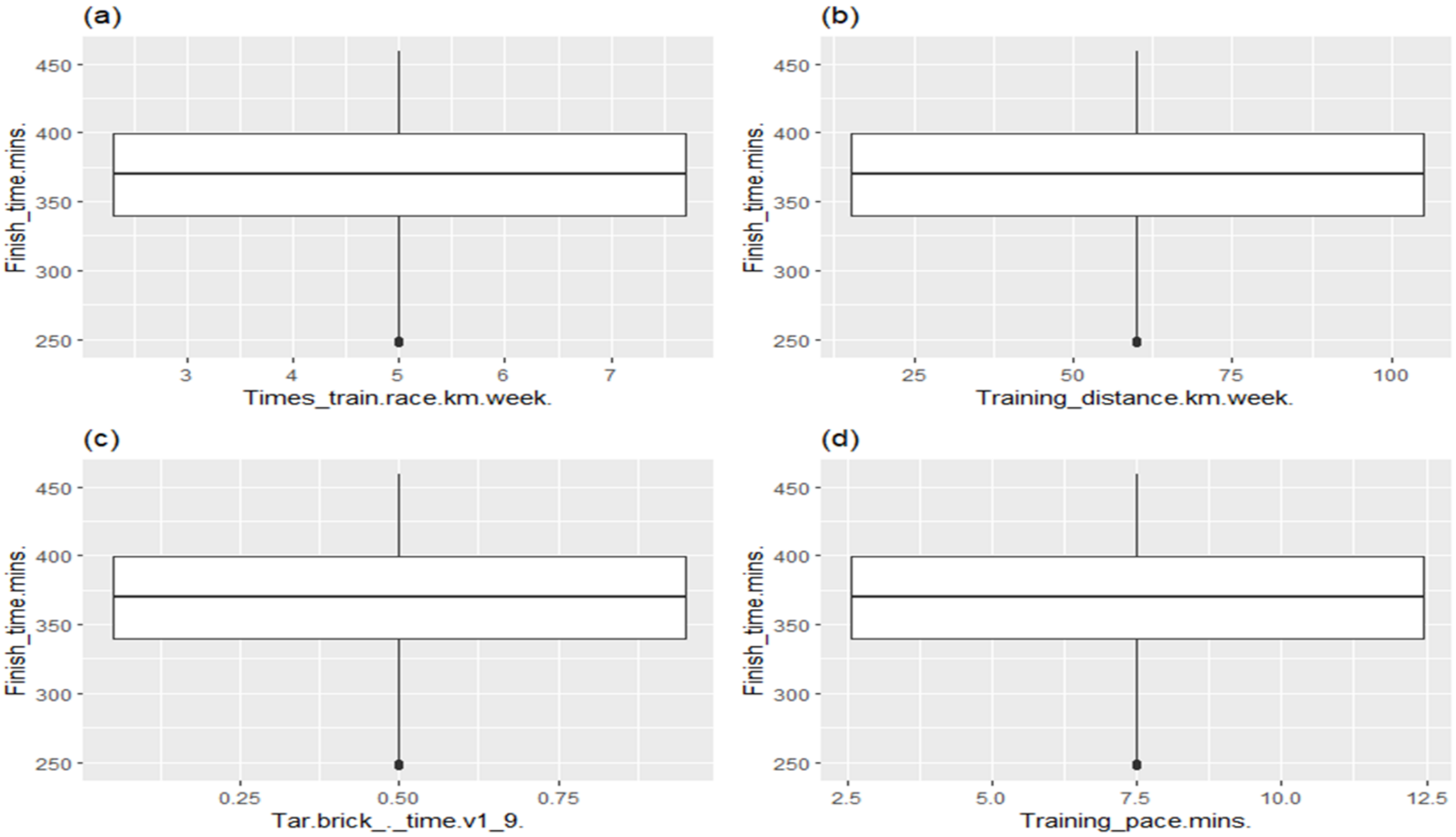


Figure 2.4: Boxplots for ultra-marathon race type. Panel (a) is a boxplot of finish time and times train race per week, panel (b) is a boxplot of finish time and training distance per week, panel (c) is a boxplot of finish time and tar/brick percentage training time and panel (d) is a boxplot of finish time and training pace

2.5.1 Finish time

Half and ultra-marathon runners

According to Figure 2.3 and Figure 2.4 on average, most runners in both race types train five times per week while runners in the half marathon race have an average finish time of about 150 minutes with a minimum finish time of 50 minutes and maximum finish time of 200 minutes. More runners in the ultra-marathon take 370 minutes to finish the race. On average, runners in both race types cover 60 km of training per week while about 50% of their training time is dedicated on training on tar and brick tracks. It is also evident that runners have an average training pace of 7.5 minutes.

Figure A.3 illustrate that there is evidence that the more runners dedicate much of their time to training per week, the better their finish time performance for runners, this is true in both race types, moreover, male runners seem to have a superior finish time performance than female runners. This behavior of a negative relationship between training times per week and finish time is persistent in both half and ultra-marathon race types however more evident in the ultra-marathon race type. Age of athletes seem to have an impact on the finish time performance of athletes. Younger athletes seem to perform better than older athletes throughout both race types. The four age categories have evident differences when number of years in recreational running is used as a training load variable, see Figure A.7. Such age differences are seen in both race types more evident in the half marathon race type where younger runners perform better than older runners. There is evidence also that runners with more years in recreational running have better finish times than runners with fewer years in recreational running, this is true especially for female runners. The abovementioned relationship of finish time and recreational running is quite similar to that of distance running and finish time as seen as seen on Figure A.6.

Half marathon runners who train for more kilometers per week seem to have superior finish times than those who train for shorter distances as seen on Figure A.4, this relationship holds for ultra-marathon runners also. Younger athletes still seem to perform better than older athletes, male runners still have better finish times than female runners. The more runners spend time training on tar and brick tracks does not seem to have a positive association with finish time performance for runners in both race types this is seen in Figure A.8, In fact, a slight positive relationship is visible in the two variables therefore runners who dedicate more time training on tar and brick tracks have worse performance than runners who spend lesser time training on tar and brick tracks.

Figure A.5 is a visualization of training pace and finish time performance measure, it can be deduced that male runners still outperform female runners in terms of finish times in both race types. Male and female runners who have lower training paces have superior finish times than those who have higher training paces however, there are diminishing performance differences between male and female runners as training pace increases. When it comes to the four age categories, there are minor differences between the age groups however there is slight evidence that the older runners get, the younger the runners are, the better they perform.

2.5.2 Race Pace

Half and ultra-Marathon runners

Most runners in both race types have race paces of about 6.5, 7.2 see Figure 2.5. Male runners who train more per week have slight superior racing pace performances than female athletes who train more per week and in general terms. It is also evident that athletes who train more per week have better racing paces than those who train lesser per week even though the differences are not major see Figure A.9. Runners who cover longer training distances per week still outperform those who cover shorter training distances per week in training when performance is measured in terms of race pace. The four age categories do not seem to have significant differences of race paces for both training times per week and training distance per week while the male runners seem to outperform female runners, but the differences are not major.

According to Figure A.10, spending more time training on tar and brick tracks does not seem to be associated with racing pace performance for runners in both race types. Age does not seem to have an evident association with race pace performance when training load is measured by time spent training on tar/ brick tracks while male runners still perform better than female runners. A similar relationship of training pace for runner and finish time is evident once again in Figure A.10. where the relationship of race pace and training pace is investigated, runners who have better training paces have superior race pace performances than those with poor training paces. Major differences in the age categories get more evident when number of years in recreational running increase, with younger runners having a better race pace than older runners, this is true for distance running too. see Figure A.11. There is also slight evidence that more years in recreational and distance running especially in females is associated with better race pace performances runners.

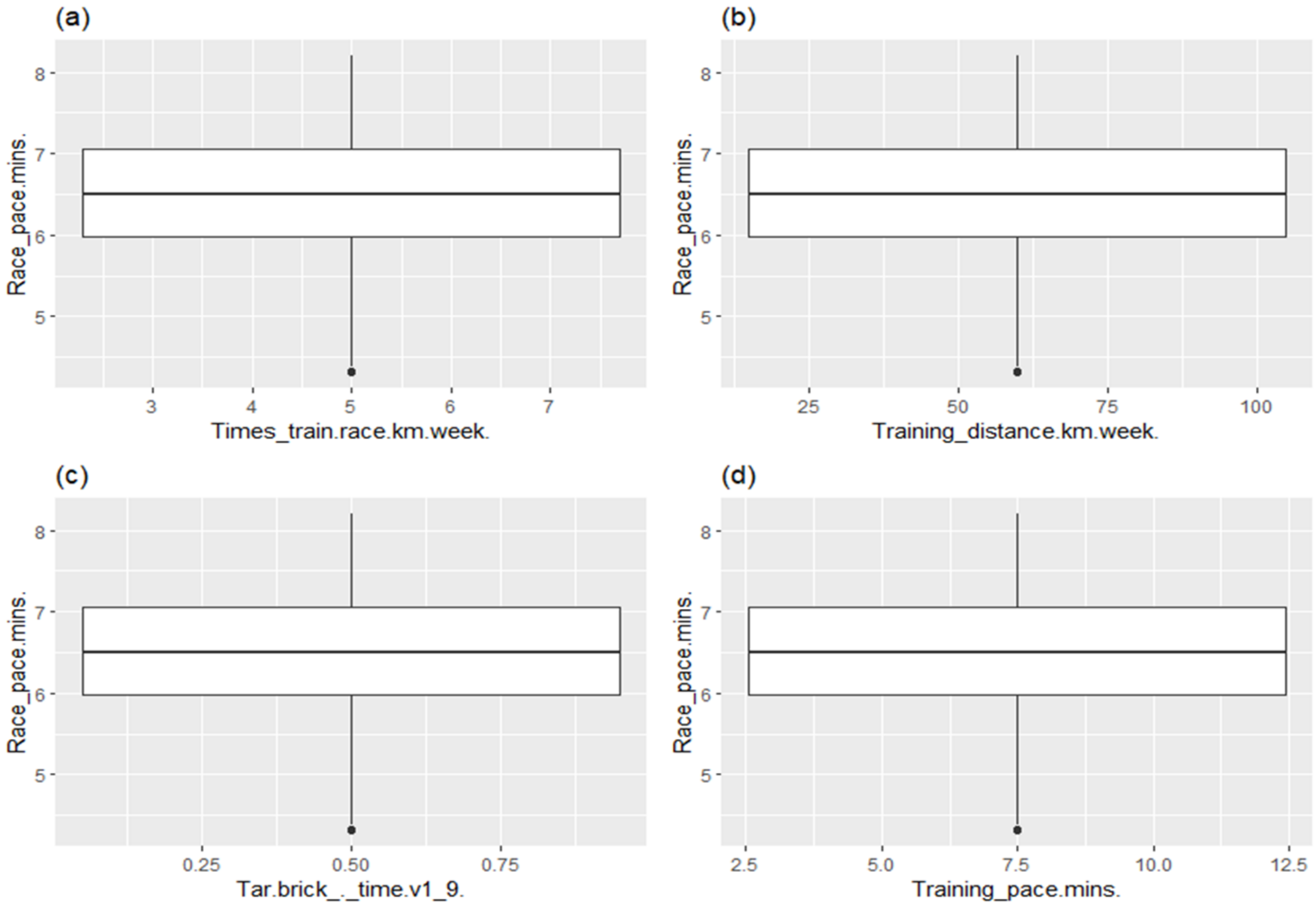


Figure 2.5: Panel (a) is a **boxplot of race pace and times train race per week, panel (b) is a boxplot for race pace and training distance per week, panel (c) is a scatterplot of race pace and tar/brick percentage training time and panel (d) is a boxplot of race pace and training distance.**

2.6 DISCUSSION OF EXPLORATORY DATA ANALYSIS

From graphical representations given above, there seems to be more evidence that male runners generally have better finish times and racing paces than female runners in both race types. There is also evidence that there are minor to major differences of finish time and race pace performances for both race types between the four age categories with younger runners performing better than older runners, this could be explained by stamina and strength of young bodies. Runners who train for longer distances and more per week seem to have superior finish times and racing paces while training on tar and brick tracks does not have an association with performance. Runners who have superior training paces have better finish time and racing pace performances for both race types while runners who have been participating in distance and recreational running especially female runners, have better finish times and race paces than runner with fewer years of participation in distance and recreational running. The aim of exploratory data analysis as stated is to assess the distribution of the data, to be more precise, to attain knowledge of the behavior of the predictor variables in respect to the response variable. The abovementioned differences that seem to be visible on graphical visualizations will further be tested in the following chapter using Analysis of variance (ANOVA). This will be a rather formal and detailed approach of testing for significant differences between the age categories, gender categories and race types.

CHAPTER 3
Multivariate Analysis of variance (MANOVA)

3.1 INTRODUCTION

The analysis of variance (ANOVA) involves one predictor variable which has a number of different levels which correspond to the different groups (Pallant, 2010). In the case of this study, the data was split by race-type into the half and ultra-marathon datasets and then each of those datasets further split to extract datasets that have one response variable i.e. finish time and race pace. The predictor variables in question for this analysis of variance are Gender and Age Category.

In this section, one-way MANOVA tests is used to see if there are any variations in finish time and race pace for the different age and gender categories. Following the MANOVA, one-way ANOVA tests is done to investigate whether there are significant differences in race paces and finish times per age and gender category. Finally, in each part of predictor variables, a multiple comparison test is performed to discover which groups of the predictor variable are different.

The purpose of doing the MANOVA, ANOVA, and multiple comparison tests at the same time is to answer the following three important questions: (1) Are there any disparities in finish time and race speed between the means of (age categories, gender categories, and race type)? (2) Which of the response variables (finish time and race speed) contributed to the overall difference if there were differences in finish time and race pace between (age, gender, and race categories)? and (3) What are the distinctions between the categories? Once these questions are answered, it will be obvious whether the two response variables (finish time and race pace) behave differently or similarly, allowing for either a holistic or predictor analysis (finish time and race pace). The above-mentioned questions will be answered at the end of this section, but first, let us make sure the following MANOVA assumptions are not violated by performing relevant tests. This analysis is used to gain insight on the variability within the groups of the predictor variable (Pallant, 2010) so as to determine whether the data should be split further into the predictor variable groups before modelling in Chapter 5.

3.2 ANOVA Assumptions

3.2.1 Testing for Normality

For the purpose of this research, histograms and Q-Q plots will be used to examine the normality of the distribution of the two response variables Race Pace and Finish Time. Accompanying these graphical representations of the distribution, will be two distribution metrics, kurtosis and skewness. The skewness and kurtosis metrics of the Gender and Age Category variables will give an idea about the distribution.

George & Mallery (2003) suggest that a value for kurtosis that is in between the range -2 and +2 is considered acceptable to consider the data as univariate normal and (Curran *et al.*,1996) propose a similar threshold of about 2.0 for skewness. Figures A.13 to A.16 are histograms annotated with kurtosis and skewness values, it is clear that none of them breach the threshold proposed (George & Mallery, 2003b) and (Curran *et al.*, 1996). Therefore, the null hypothesis is rejected and the distribution of the data can be accepted as normal. Furthermore, Tabachnick & Fidell (2007) suggest that The Central Limit Theorem reassures us that, with sufficiently large samples, the sampling distributions of means will be normally distributed regardless of the distributions of variables.

3.2.2 Testing for Homoscedasticity

The Levene's test is commonly used to compare the variability of two or more groups. We are comparing the null hypothesis of equal variances against the alternative hypothesis of different variability. Levene's test should not be significant for the null hypothesis to be retained. Levene’s test showed that the variances for Race pace per Age Category were not equal, F = 21.87, p <0.0001. Figures A.17 and A.18 are results for the Levene’s Test of Homoscedasticity for the half and ultra-marathons respectively, all of which have a p-value that is significant. Therefore, we reject the null hypothesis and assume that the variances are not equal.

3.2.3 Testing for Independence

The Chi-Square test of Independence is used to determine if there is a significant relationship between two categorical predictor variables. Figures A.19 to A.22 are Chi-square test results that deals with testing independence. The p-values are less than the chosen significance level $\alpha = 0.05$, we can reject the null hypothesis, and conclude that there is an association between Gender and Age categories.

3.3 ONE-WAY MANOVA

When there are multiple response variables, multivariate analysis of variance (MANOVA) is a variation of analysis of variance. These response variables must be related in some way, or there must be a conceptual rationale for grouping them together. This section uses a combination of the 2 response variables in the form of a composite variable. MANOVA examines the groups and determines if the mean differences in the response variables between the groups are likely to have arisen by chance. MANOVA accomplishes this by generating a new summary response variable that is a linear combination of all of the original response variables. The new combined response variable is then used in an analysis of variance (Pallant, 2010).

To evaluate age category variations in race performance, a one-way between-groups multivariate analysis of variance was used. Age category was used as the predictor variable, and a separate but equivalent one-way MANOVA was conducted with Gender replacing Age category as the predictor variable. Figure A.23 is a summary of the one-way MANOVA results and shows that there was a statistically significant difference in marathon performance based on age category, p <.05 for Wilk's Lambda. The results validate the need for further ANOVA testing in order to attain knowledge of the variables that are statistically significant.

Figure A.24 shows between-subjects effects and shows that age categories have a statistically significant effect on both Race Pace (p <.05) and Finish Time (p <.05). The results validate the need for further ANOVA testing in order to attain knowledge of the variables that are statistically significant. Figure A.25 shows the multiple-comparisons post-hoc tests using Tukey’s HSD and shows that there is a difference within the Age categories and illustrates that although there exist differences within the group, not all of them are statistically significant.

3.4 CONCLUSION

Using the F-tests for significance together with the difference between mean tables the output shows that athletes perform significantly differently in both race types (Half-Marathon and Ultra-Marathon race types) in terms of finish time and race speed. Individual variable analysis reveals that athletes from the two race types perform considerably differently under each performance metric (finish time and race pace). Only the runners in age category 4(oldest) have performances that are significantly different from the other groups. The number of runners in this age category are so small compared to the other age categories that the difference will be redundant moving forward into the modelling procedures and variable selection.

CHAPTER 4
REGRESSION MODELS AND PERFORMANCE METRICS

4.1 INTRODUCTION

After a thorough investigation of existing differences between the categorical predictor variables, it is crucial now to have a deeper understanding of which predictor variables are important in predicting (explaining) the response variables. Regression is a stepping stone in achieving the ultimate goal of achieving however it is equally important to understand how these techniques work before diving in the variable selection section which will also be discussed later in this Chapter. Detailed step by step discussion of each regression technique employed in this study and how these techniques doe variable selection and regression is the focal area of this chapter. To assess the performance of each model employed and know which model perform best, performance metrics for each model will be discussed also in this Chapter. Imputed data models are also subject to assessment through performance metrics thus, a method of data imputation utilized is given in detail in this Chapter.

4.2 ORDINARY LEAST SQUARES

Statistical learning always aims on building models for given problems and the aim is to ultimately build a model that will have a higher prediction accuracy. The first step to achieving an accurate model is to minimize the errors produced by the estimated linear model.

$$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 X_1 + \hat{\beta}_2 X_2 + \ldots + \hat{\beta}_P X_P \ . \qquad (4.1)$$

Equation 4.1 is the estimated regression line where $\hat{\beta}_0$ is the estimated intercept and $\hat{\beta}_i$ are the least square estimates while X_i are the predictor variables. Let RSS be denoted by the following Equation:

$$RSS = \sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_{\ j}\, x_{ij}\right)^2. \qquad (4.2)$$

According to (James *et al.*, 2013) Equation 4.2 given above is derived from the error terms Equation given by $e_i = y_i - \hat{y}_i$. This Equation denotes the sum of squared residuals for the given linear Equation which is the sum of squared differences between the estimated response values and the estimated response values. To find the minimum RSS, the coefficients β_j that will minimize Equation 4.1 are the priority. The LASSO follows the same procedure however with some important differences.

4.3 LASSO

The LASSO introduced by (Tibshirani, 2011) is a supervised machine learning shrinkage algorithm that is used to avoid overfitting and produces sparse matrices of variables by adding a penalty to the magnitude of coefficients (Breiman, 1995). The LASSO uses the l_1 penalty by finding optimal β_j coefficients that minimize the following:

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{J=1}^{P}\left|\beta_j\right|.$$

$$= RSS + \lambda\sum_{J=1}^{P}\left|\beta_j\right|. \qquad (4.3)$$

Equation 4.3 has a turning parameter $\lambda \geq 0$ and $\lambda\sum_{J=1}^{P}\left|\beta_j\right|$ is the shrinkage penalty and the l_1 norm is given by $\sum_{J=1}^{P}\left|\beta_j\right|$ (Frank and Friedman, 1993), this also helps minimze erros between actual and predicted values. Equation 4.3 finds the optimal coefficients by shrinking some of them to zero because of the nature of the shrinkage penalty. An optimal turning parameter λ is obtained through cross validation and it is critical to obtain an optimal one because it plays and important role in the estimated β_j coefficients. When λ is large, some β_j will shrink to zero and when λ is small, the penalty term converges to zero hence the LASSO regression is simply the ordinary least squares. The abovementioned ability of the LASSO to shrink some coefficients to zero qualifies the LASSO as a regression method and also variable selection method.

It can be shown that Equation 4.3 is the same as minimizing the RSS with $\sum_{J=1}^{P}\left|\beta_j\right| \leq s$ as the constraint as shown in Equation 4.4 below.

$$minimize\left\{\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2\right\} \quad subject\ to\ \ \sum_{J=1}^{P}\left|\beta_j\right| \leq s\,. \qquad (4.4)$$

Similarly, Equation 4.4 has a constant value s that determines the estimated β_j. When s is small, the LASSO regression coefficients β_j will have a stricter constraint: $\sum_{J=1}^{P}\left|\beta_j\right|$ thus forcing some of them equal to zero while a large s value will lead to a large constraint hence being similar to the ordinary least squares.

Constant RSS values are shown by the red elliptic region in Figure 4.1 left panel, large RSS values form elliptic regions that are far from the center which is the usual Ordinary least squares (OLS). The constraint of the LASSO $\sum_{J=1}^{P}\left|\beta_j\right| \leq s$ produces a square green constraint region.

Equation 4.4 shows that when the RSS coefficientsβ_j are minimized in respect to the constraint; $\sum_{J=1}^{P}|\beta_j| \leq s$, the coefficients β_j that produced that RSS ellipsis is the estimated LASSO coefficients $\sum_{J=1}^{P}|\beta_j| \leq s$. The LASSO coefficients are produced when the constraint meets the RSS ellipsis.

4.4 RIDGE REGRESSION AND ELASTIC NET

The elastic net is part of the shrinkage methods family with a different shrinkage penalty. Before discussing the elastic net, let us first review ridge regression because it plays an important role in the build-up of the elastic net. According to (James, 2013), The LASSO follows the same procedure for β_j coefficients minimization but uses the l_2 norm given by $\sum_{J=1}^{P}\beta_j^2$ as seen in Equation 4.5 below:

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{J=1}^{P}\beta_j^2. \tag{4.5}$$

The shrinkage penalty of ridge regression minimizes Equation 5 by shrinking the coefficients towards zero however the coefficients are never equal to zero unlike the l_1 norm used by the LASSO regression. This is where the major difference lies between the LASSO and the ridge regression, the LASSO regression automatically selects a sparse model by selecting $p \leq n$ variables while all p variables are chosen by the ridge regression.

It can be shown that Equation 4.5 is equivalent to the following:

$$minimize\left\{\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2\right\} \quad subject\ to\ \lambda\sum_{J=1}^{P}\beta_j^2 \leq s. \tag{4.6}$$

The minimization procedure of the LASSO is works in such a way that the l_2 norm creates a circular blue constraint region as seen in Figure 4.1 right panel. The elliptic region represents constant RSS values for different coefficients. Since the RSS is optimized in respect to the circular constraint region: $\sum_{J=1}^{P}\beta_j^2 \leq s$, the optimal coefficients are obtained where the RSS elliptic region contacts the circular ridge constraint area. This shows that the ridge regression coefficients will not be exactly equal to zero because the two regions meet where coefficient $\beta_j \geq 0$.

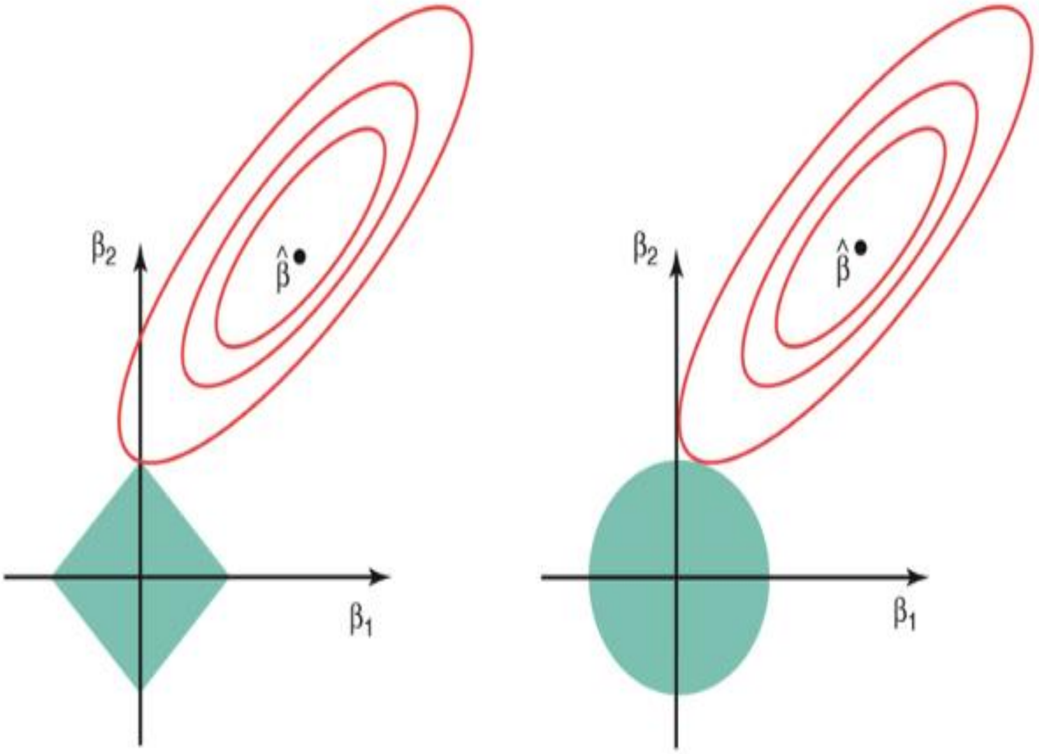


Figure 4.1: The left panel is the LASSO and the right panel is the ridge regression method.

Source: James et al, 2013:222

The major drawback of the l_1 norm is that it picks one variable from a group of collinear predictors for explaining the response. This drawback of the LASSO is neutralized by the l_2 norm by ridge regression which includes all predictor variables in the final model even though some variables may not be important to include. The elastic net regression comes in between these two shrinkage methods by utilizing both the l_1 and l_2 norm in optimization (Park & Konishi , 2015). Equation 4.7 below shows the how the minimum β coefficients that optimize the following are obtained.

$$\hat{\beta} = \left(1 + \frac{\lambda_2}{n}\right)\left\{\text{arg min } \beta \left\| y - \sum_{j=1}^{p} x_i \beta_j \right\|^2 + \lambda_1 \|\beta\|_1 + \lambda_2 \|\beta\|_2^2 \right\}. \qquad (4.7)$$

It can be seen from Equation 4.7 that the combination of l_1 and l_2 norms are used with the turning parameters λ_1 and λ_2 as penalty terms to determine optimal coefficients. When an optimal turning parameter are selected, the elastic net works well in selecting a sparse model while taking account of grouped correlated variables (ter Braak, 2009). It can also be shown that Equation 4.7 is equivalent to the Equation 4.8 below

$$\hat{\beta} = \arg\min|y - X\beta|^2 \ subject\ to \ \ (1-\alpha)|\beta|_1 + \ \alpha|\beta|^2 \ \leq s. \qquad (4.8)$$

The elastic net penalty $(1-\alpha)|\beta|_1 + \ \alpha|\beta|^2 \ \leq s$ term is a halfway penalty that resulted from the LASSO and the ridge penalties respectively. The α value plays an important role in the penalty term of the elastic net optimization because $\alpha = 0$ results to the LASSO, $\alpha = 0.5$ is the elastic-net while $\alpha = 1$ is simply the Ridge

4.5 TREE BASED METHODS

4.5.1 Regression trees

Regression trees are part of decision tress applied to regression problems (James et al, 2013). Figure 4.2 is a simple regression tree. There are two predictor variables $X_1 and\ X_2$. The first split that results in the first two branches is for $X_1 \leq \ \ t_1$ where t_1 is an integer value, the other split is for $X_1 > \ \ t_1$. The top two branches are referred to as internal nodes and the bottom three branches that have values displayed are referred to as terminal nodes. The values displayed on the terminal nodes are the region means. The second internal node for $X_1 > \ \ t_1$ is further split into two nodes $X_2 \leq \ \ t_2$ and $X_1 < \ \ t_1$. The regions that resulted from the splitting are shown in Figure A.12. A step by step detailed discussion of how to build a regression tree is discussed later in the Section.

Similarly, to when predictions are made, variables $X_1 and\ X_2$ are used for the prediction. Each and every region has a mean response, predictions for observations are determined by the mean response of the region where such an observation falls into as seen on Figure 4.2 terminal nodes.

To build a regression tree, the construction procedure can be broken down into two steps;

Step 1

- The predictor space is divided into M non-overlapping regions: $R_1, R_2, \ldots, R_M$

Step 2

- Each region has a mean response and for every observation, if an observation falls in region M, the mean of region M will be the prediction.

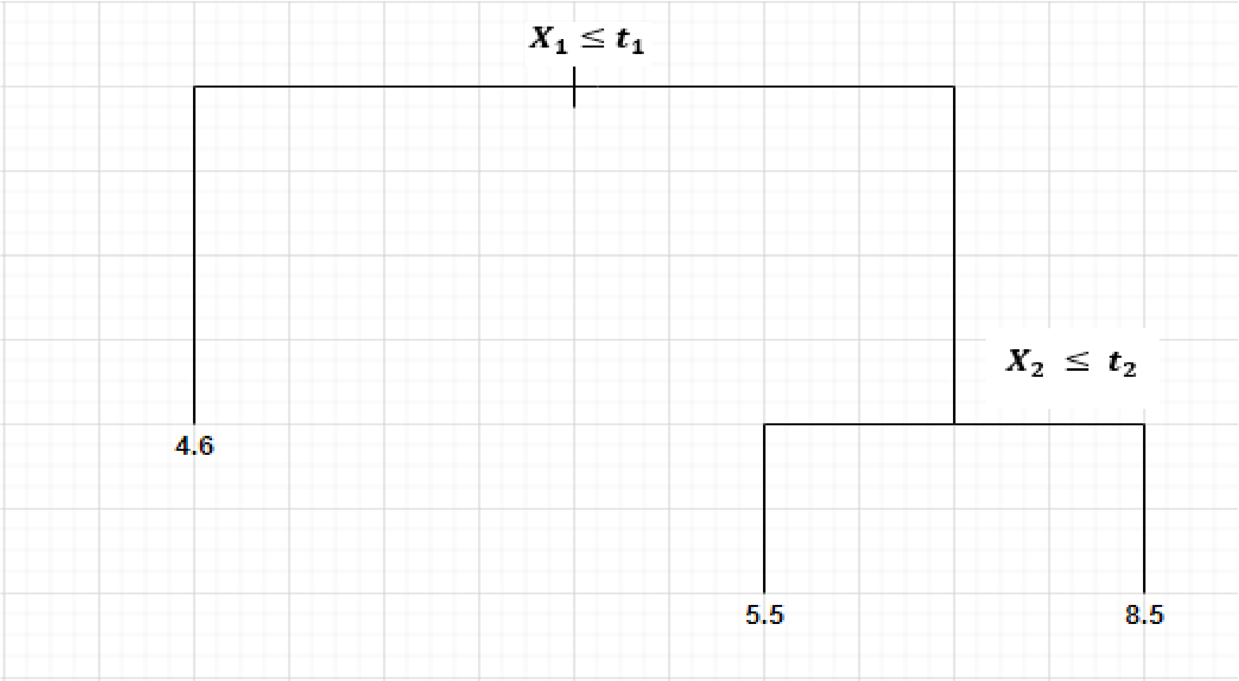


Figure 4.2: Illustration of a regression tree

To future elaborate how a full decision tree is built, we first have to understand how the predictor space is split into M regions. To obtain optimal splits, the RSS is minimized and the RSS is as follows

$$\sum_{m=1}^{M} \sum_{i \in R_m} (y_i - \widehat{y_{R_m}})^2. \quad (4.9)$$

Where y_i, is the training observations, $\widehat{y_R}$, represents the mean response for the M_{th} region. A cutoff value s is chosen when the splitting procedure takes place, an optimal s is the one that the minimum RSS is produced for

$$R_1(j, s) = \{X | X_m < s\}.$$

and

$$R_2(j, s) = \{X | X_m \geq s\}. \quad (4.10)$$

So that values of s and j that will result in the minimum of 4.11 are obtained.

$$\sum_{i:\, x_i \in R_1(m,s)}(y_i - \widehat{y_{R_1}})^2 \; + \sum_{i:\, x_i \in R_2(m,s)}(y_i - \widehat{y_{R_2}})^2. \tag{4.11}$$

The above splitting process is repeated for the next best predictor and cutoff value that will result to the greatest reduction of the RSS. The splitting is not done on the entire predictor space but on previously obtained regions. The process is reiterated until a specific stopping criterion is met.

4.5.2 Bootstrapping and Bagging

Full Unpruned regression trees build models that suffer from high variance and very low bias, this impairs the prediction accuracy of the predictive model. To overcome this shortfall, an average of multiple predictive models is taken to build a better predictive model, this will reduce the variance of the predictive model and increase its predictive accuracy. To obtain this, we first obtain B estimated bootstrapped statistical learning models $\widehat{f^1}(x), \widehat{f^2}(x), \ldots, \widehat{f^B}(x)$ Where $\widehat{f^b}(x),$ is the estimated statistical learning method (James et al, 2013). The average of the obtained estimated models is given by

$$\widehat{f_{avg}}(x) = \frac{1}{B}\sum_{b=1}^{B}\widehat{f^b}(x). \tag{4.12}$$

A practical way of obtaining B estimated regression trees is to obtain multiple samples of training sets from a single training set since we do not have access to multiple training sets. After obtaining the bootstrapped training samples, full unpruned trees are trained on the samples to obtain $\widehat{f^{*1}}(x), \widehat{f^{*2}}(x), \ldots, \widehat{f^{*B}}(x)$. To obtain a low variance and better predictive decision tree, the average of the decision trees is taken as shown in Equation 4.12 This extended procedure of bootstrapping described here is called bagging.

$$\widehat{f_{bag}}(x) = \frac{1}{B}\sum_{b=1}^{B}\widehat{f^{*b}}(x). \tag{4.13}$$

4.5.3 Random forest

Random forest proposed by (Breiman, 2001) is an ensemble method with some similarities to bagging. This regression technique like the other implemented shrinkage methods is also not affected by collinearity of variables. Equation 4.14 below shows how random forest works.

$$\hat{f}_{rf}(x_0) = \frac{1}{B}\sum_{1}^{B}\hat{f}^{*b}\,(x_0). \tag{4.14}$$

Consider Equation 4.14. Random forest regression, similar to bagging has a set of B trees. The average of the B trees is the final output of the procedure. Let us take a step by step explanation of the given procedure.

Step 1. Consider a given dataset with X-input vector X and response vector Y.

- Draw bootstrap samples from the given dataset. The samples are randomly selected with replacement.

Step 2. Construct an unpruned tree from each bootstrap sample and do the following at each node:

- Randomly select a subset of predictors for the split, this subset is called a M_{try} predictors where $M_{try} < p$. $M_{try} = p$ is just bagging, an ensemble method with a similar procedure to random forest.

Step 3. Repeat step 1 and 2 until a full tree is constructed and no further splits are possible.

Step 4. Obtain the average of the trees and use it for regression.

4.5.4 Imputation

missForest

Imputing missing data can be cumbersome for mixed typed data however, missForest missing data imputation method works well with mixed -type data. (Bühlmann and Stekhoven, 2012) This is a non-parametric imputation method because it uses the Random Forest technique in the imputation procedure, moreover, it does not make any assumptions of the distribution of the data. (Breiman, 2001).

Procedure:

I. Impute missing values with mean or mode for every variable with missing values

II. Order columns with missing values in ascending order.

III. Fit a Random Forest model with all but missing values of the first variable in the order.

IV. Predict the missing values that are missing in the first variable in the order.

V. Repeat the above procedure for all other variables with missing values individually in the order of number missing values.

VI. Repeat the training and predicting steps above and stop when a stopping criterion specified by the user is reached (could be the target number of iterations achieved).

The more iterations are made, the better the prediction accuracy of the trees hence why the missForest procedure being more advantageous.

4.6 PERFORMANCE METRICS

The two performance measures that will be used in the analysis to evaluate the implemented machine learning algorithms are the root mean square error (RMSE) and the mean absolute error (MAE).

Root mean square error

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{y_i})^2}. \tag{4.15}$$

Mean absolute error

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|. \tag{4.16}$$

Equations 4.15 and 5.16 have n as the number of observations. y_i, are the true values while $\widehat{y_i}$, are the predicted values. In Equation 4.15, the RMSE measure takes the average of squared errors while Equation 4.16, the MAE takes the sum of the absolute values of the errors and divides by the number of observations. These two measures determine the average magnitude of the error (James, 2013).

4.7 CONCLUSION

Regression techniques discussed in this chapter give an insight of how the modeling and variable selection procedures are works for each model and also, understanding these procedures help in understanding the results obtained the next Chapter. Depending on the nature of the data and how each regression technique works, results obtained by these models will not be similar thus leading to different performances when assessed through performance metrics. Results obtained by employed models are discussed in the in the next Chapter.

CHAPTER 5
VARIABLE SELECTION AND MODEL EVALUATION

5.1 INTRODUCTION

The variable selection process performed by the three fitted models (LASSO, elastic net and Random forest) is an important stepping stone in the analysis. The purpose of variable selection is to return a subset of important predictor variables in the models. Subsets of predictor variables obtained by the three models will be discussed individually and a through comparison of variables selected will be performed later in the chapter.

Since there are two response variables: Finish time and Race pace while the analysis is focused on two race types: Half marathon and ultra-marathon runners. Our first focus of variable selection will be the LASSO model output coefficients.

5.2 VARIABLE SELECTION

5.2.1 Lasso model

Figure 5.1 gives evidence that training pace generally carries the highest level of importance in predicting finish time and race pace for runners in the two race types. Age and gender are also highly important non-training load variables. Treadmill training and training distance are the two important training load variables while number of years in distance running, training distance and number of years in recreational running have the least levels of importance.

5.2.2 Elastic-net model

The Elastic net model variable selection output is given in Figure 5.2 below for the two race types and each of the two response variables. The results obtained Figure 5.2 Have more similarities to that in Figure 5.1 of the LASSO models. Training pace remains the most important variable in predicting finish time and race pace for both half and ultra-marathon runners. The Age and gender also appear amongst variables of high importance while Important training load variables are training distance, treadmill training and number of years in recreational running. Number of years in distance running and tar/brick training are the least important variables. Gravel training is discarded in predicting finish times and race paces for only half marathon runners.

5.2.3 Random forest model

The random forest models have a slightly different way of displaying important variables. Important variables chosen by the four fitted random forest models are displayed on graphs. Figure 5.3 above shows the plots of important variables and each of the four plots are variable importance plots representing each random forest model fitted. The small square brackets in the graphs indicate the extent of variable's importance. It can be deduced from all four graphs that training pace is extremely important in explaining both finish time and race pace under both race types while training distance and number of years in recreational running follow. Gravel time, age, time taken on training on a treadmill and gender are the least important in explaining training pace for ultra-marathon runners. The level of importance for training load variables in the half marathon race explaining finish time is practically equal for all but training pace variable however this does not hold for training load variables explaining finish time in the ultra-race type. Adding more training data points does bring about any change in the existing variable importance.

5.2.4 Discussion of variable selection

The three regression models used for variable selection show that the two shrinkage methods have quite similar lists of selected variables while output obtained from the random forest models has significant differences from the output obtained from the shrinkage methods. Training pace is significantly important in predicting both finish time and race pace for half and ultra-marathon runners. Training distance and number of years recreational running are also highly important in predicting finish time and race pace for marathon runners in both race types. Gravel training is generally the least important training load variable while age and gender have mixed results. Distance running, treadmill training, tar/brick training and training times per week also have mixed results.

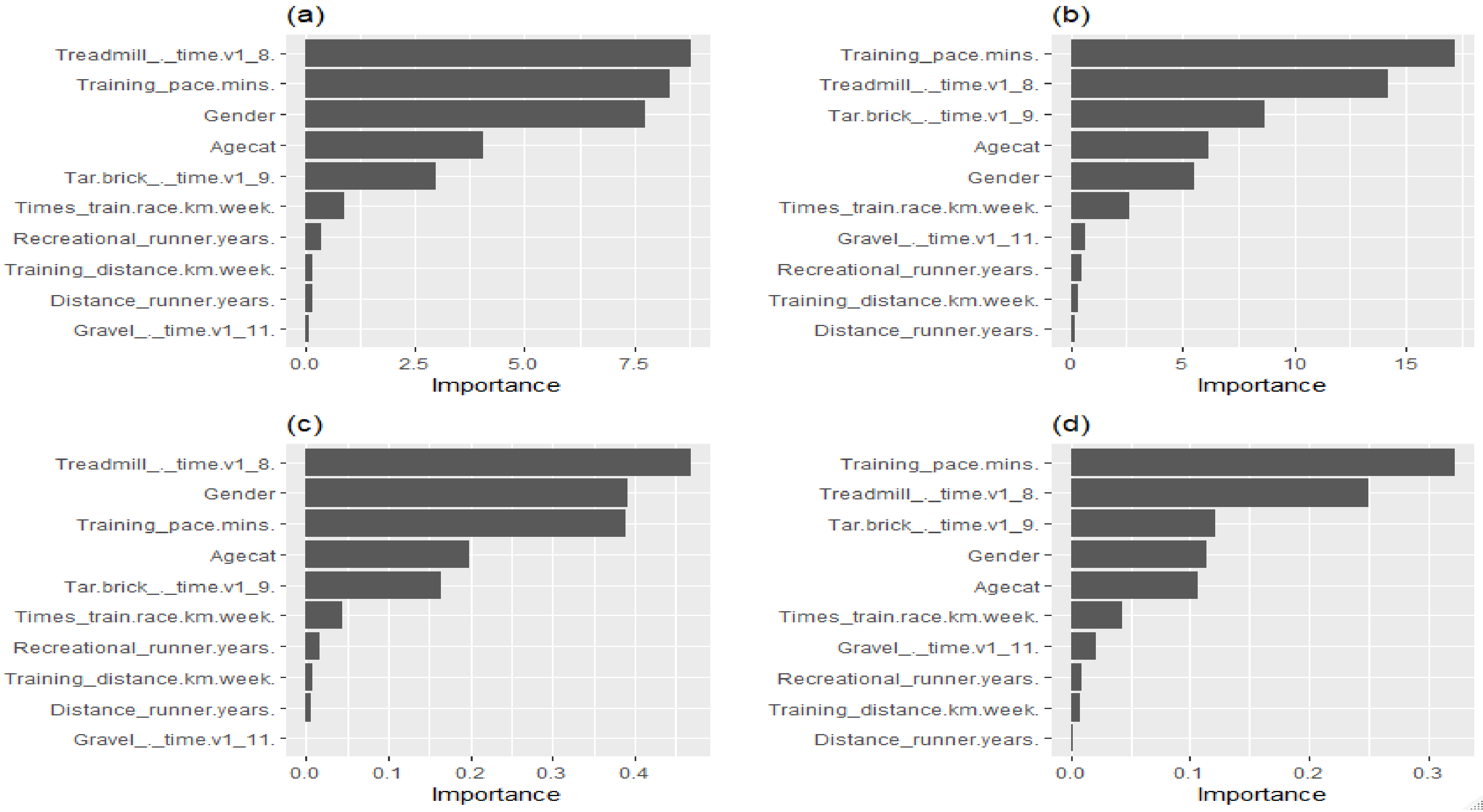


Figure 5.1: Variable importance plots for LASSO model. The length of the columns represents the level of importance of the variables. The variables are displayed starting with the most important variable at the top and the least important at the bottom.

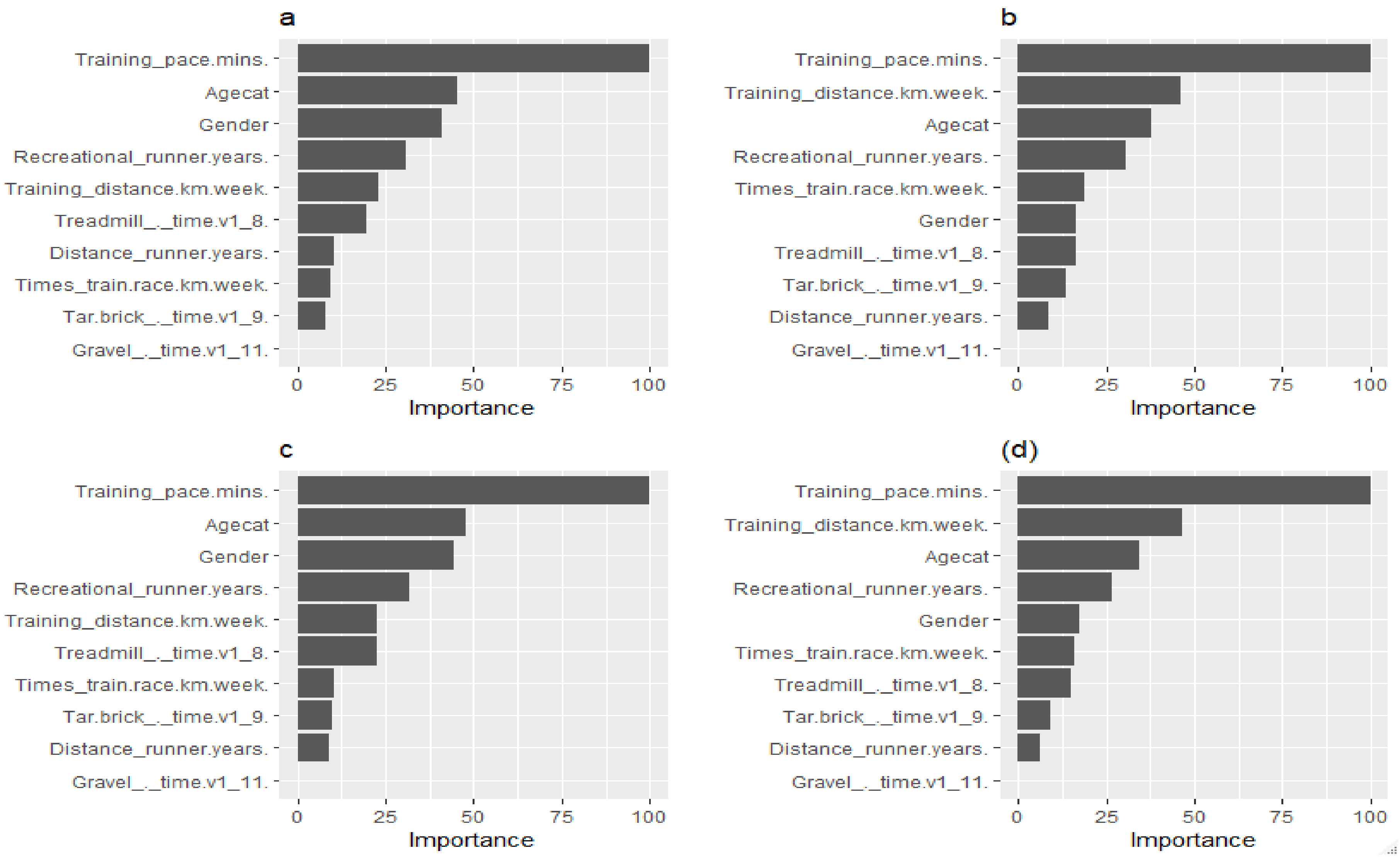


Figure 5.2: The elastic-net regress plot of selected variables.

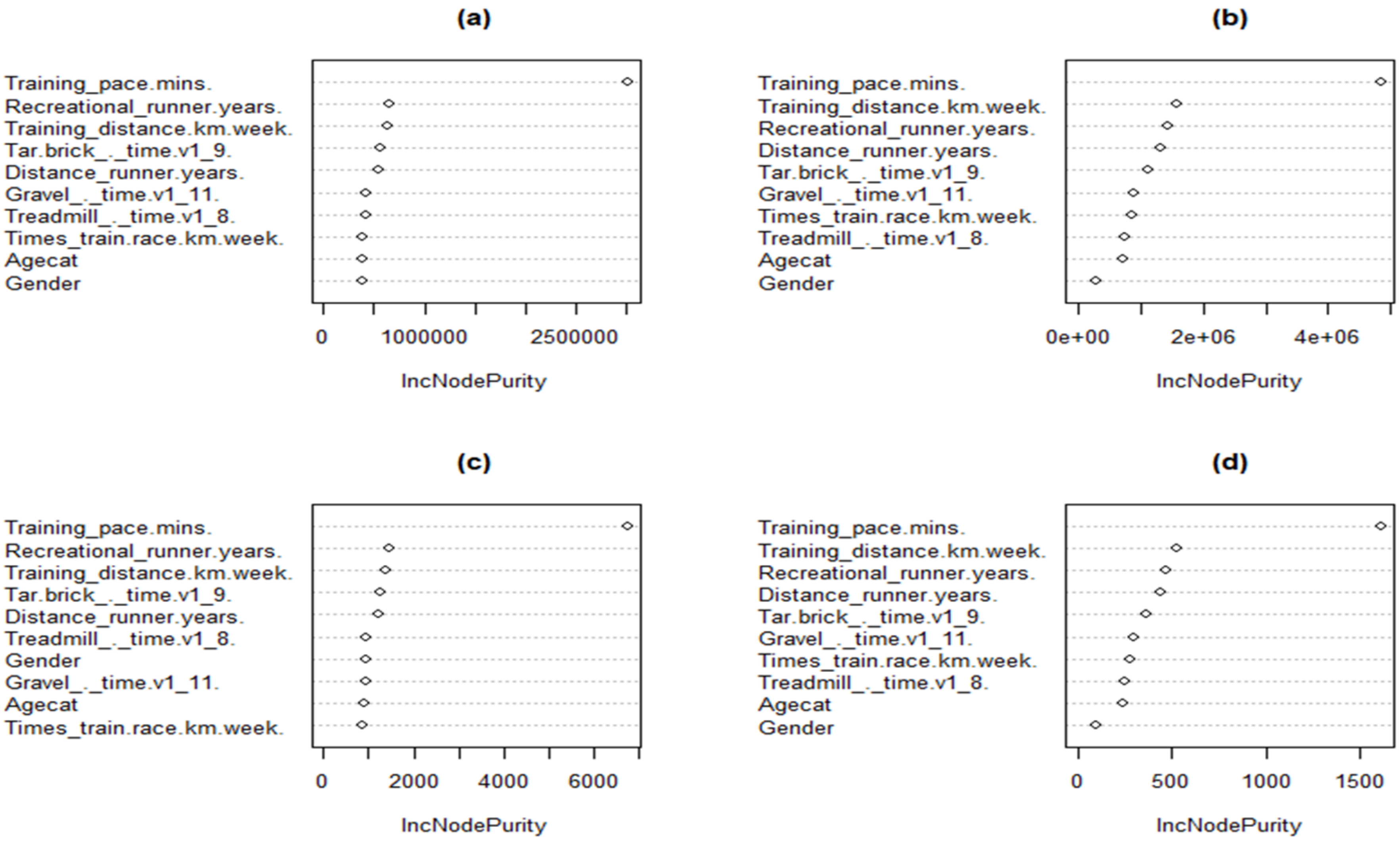


Figure 5.3: Random forest variable importance plot

5.3 MODEL EVALUATION

This main focus in this section is to evaluate the performance of the models separately under each race type for both performance measures. The Root Mean Square Error (RMSE) and the mean absolute error are the performance metrics that will ultimately lead to selection of the best performing model. For every model fitted, both test and train errors were obtained. Since the two error measures specifically measure the average magnitude of the error, the model that produces the lowest error is preferred. It is safe to use the training error as the reference point for comparisons. Percentage increase of training error to test error is measured in order to determine which model has best learned the data and developed an accurate predictive model therefore, the model with higher predictive accuracy will have the least percentage error increase. The results are given in Table 5.1 and 5.2

5.3.1 Finish time

Half-race type

The three fitted models; LASSO, elastic-net and random forest used for predicting finish time for the half-marathon runners have their respective test root mean square errors (RMSE) (19.41, 19.37 and 18.02). The Random forest clearly outperformed the three models with the least test RMSE (18.02) while the LASSO model is the worst performing model with the highest test RMSE (19.41). The fitted models also have test MAEs (15.49, 15.45 and 14.38). The model which evidently outperformed the three models with the least test MAE is the random forest (14.38) and the LASSO model has the worst test MAE (15.49). The RMSE percentage increase of the three models (2.79, 2.71 and 91.84) respectively verify that the elastic net model has the least percentage increase of the RMSE (2.71).The random forest model has the highest percentage increases of RMSE (91.84).The percentage increase of the MAEs of the three models (2.73, 2.62 and 93.40) verify that the elastic net model has the least percentage increase (2.62) while the random forest has the highest percentage increase of MAE (93.40).

Table 5.1: Summary of the RMSE performance metrics for the LASSO, elastic-net and random forest models. All three fitted models are displayed with the test and train RMSEs. The last column is the percentage increase of the RMSE from train to test RMSE.

Race type	Response variable	Model	Test RMSE	Train RMSE	% increase
Half-marathon	**Finish time**	**LASSO**	**19.41**	**18.88**	**2.8**
		Elastic-Net	**19.37**	**18.86**	**2.71**
		Random-Forest	**18.02**	**9.4**	**91.84**
Ultra-Marathon		**LASSO**	**22.05**	**21.76**	**1.32**
		Elastic-Net	**22.03**	**21.75**	**1.29**
		Random-Forest	**21.9**	**11.03**	**98.55**
Half-marathon	**Race pace**	**LASSO**	**0.88**	**0.9**	**-1.7**
		Elastic-Net	**0.88**	**0.9**	**-1.74**
		Random-Forest	**0.83**	**0.44**	**87.47**
Ultra-Marathon		**LASSO**	**0.37**	**0.36**	**1.86**
		Elastic-Net	**0.37**	**0.36**	**1.84**
		Random Forest	**0.36**	**0.18**	**97.65**

Table 5.2: Summary of the MAE performance metrics for the LASSO, elastic-net and random forest models. All three fitted models are displayed with the test and train MAEs. The last column is the percentage increase of the MAE from train to test MAE.

Race type	Response variable	Model	Test MAE	Train MAE	% increase
Half-marathon	**Finish time**	**LASSO**	**15.49**	**15.08**	**2.73**
		Elastic-Net	**15.45**	**15.06**	**2.62**
		Random-Forest	**14.38**	**7.44**	**93.4**
Ultra-Marathon		**LASSO**	**18.75**	**18.52**	**1.23**
		Elastic-Net	**18.72**	**18.52**	**1.1**
		Random-Forest	**18.34**	**9.17**	**100.04**
Half-marathon	**Race pace**	**LASSO**	**0.71**	**0.72**	**-0.66**
		Elastic-Net	**0.71**	**0.72**	**-0.78**
		Random-Forest	**0.67**	**0.35**	**90.46**
Ultra-Marathon		**LASSO**	**0.32**	**0.31**	**0.74**
		Elastic-Net	**0.32**	**0.31**	**0.62**
		Random Forest	**0.31**	**0.15**	**99.67**

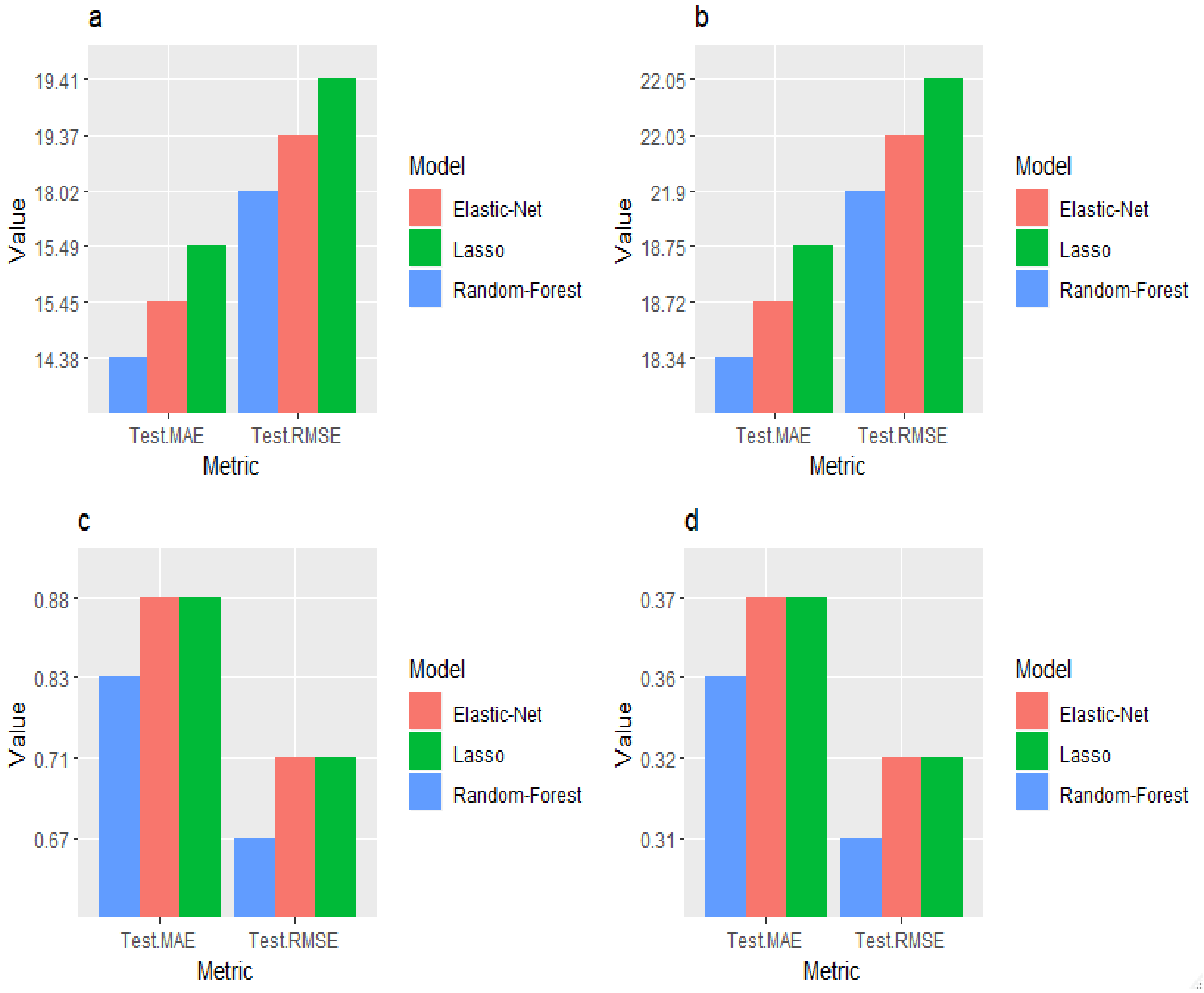


Figure 5.4: Bar plots of the MSE and MAE for the tow race types and performance measures

Ultra-race type

The three models predicting finish time for the runners who participated in the ultra-marathon race type have the three test RMSEs (22.05, 22.03 and 21.90). The Random forest once again shows outstanding performance with the least RMSE (21.90) amongst the three models while the LASSO model has the worst performing RMSE (22.05). The test MAEs for the three models (18.75, 18.72 and 18.34) used in predicting finish time for the ultra-marathon runners have the random forest model having the lowest test MAE (18.34) relatively to the other models hence being the best model in terms of MAE. The LASSO model is again the worst performing model with the highest test MAE (18.75). According to the percentage increase of the RMSEs (1.32, 1.29 and 98.55), the elastic net model has the lowest percentage increase between the training and test RMSE (1.29) while the random forest has the highest percentage increase of RMSE (98.55). The MAE percentage increases of the three models (1.23, 1.10 and 100.04) denote that the elastic net model is the best performing model (1.10) while the random forest has the most increase of the rate (100.04).

5.3.2 Race pace

Half-race type

The fitted models used for predicting race pace for runners in the half-marathon race type give RMSE output (0.88, 0.88 and 0.83), the random forest model has the lowest RMSE (0.83) while the LASSO fitted model is the worst performing model with the highest RMSE (0.88). The test MAEs for the three fitted models (0.71, 0.71 and 0.67). The random forest model is the best performing model with the least test MAE (0.67) while the LASSO model has the highest test MAE (0.71). The comparisons of percentage changes of the models RMSEs (-1.70, -1.74 and 87.47) show that the random forest is the only model that has an increase in RMSE (87.47) while the LASSO and elastic net models had a decrease in the RMSEs. The MAE percentage increase for the three models (-0.66, -0.78 and 90.46) also denote that the random forest is the only model with an increase in MAE (90.46).

Ultra-race type

The ultra-marathon runner's race pace performance was predicted using the three models with test RMSEs (0.37, 0.37 and 0.36). The random forest model has the lowest test RMSE (0.36) and the LASSO model has the highest RMSE (0.37). When it comes to the test MAE for each model (0.32, 0.32 and 0.31), the random forest model once again outperforms the rest of the fitted models with the lowest test MAE (0.31) while the LASSO model has the highest test RMSE (0.32). The percentage RMSE increases (1.86, 1.84 and 97.65) verify that the elastic net model with the lowest RMSE percentage increase (1.84) while the random forest model has the worst performance with the highest percentage increase (97.65). The MAE percentage increase for the three fitted models (0.74, 0.62 and 99.67) give evidence that the lowest percentage increase of the MAE from test to train is produced by the elastic net model (0.62) while the random forest is the worst performing model with the highest percentage increase (99.67).

5.3.3 Imputed data

The imputed missing values have a significant impact on the test errors see Figure 5.2.3, the significant increase in both the test and training RMSE and MAE may indicate a model failing to accurately learn and predict the imputed data. Even though some models built on imputed data may produce a small difference between the test and training error, the overall models for imputed data perform generally worse than models built with data that consists of complete cases. This could mean that due to a significantly large portion of the dataset being missing values, the techniques used to impute these missing values could not accurately impute values that resemble the overall pattern of the data accurately hence possibly altering the overall structure of the dataset. This shortfall may the lead to fitted models failing to accurately learn the data and make better predictions thus imputing values that are close to describing the overall behaviour of the data. See Table A.5 and A.6 for numerical values of the abovementioned performance metrics.

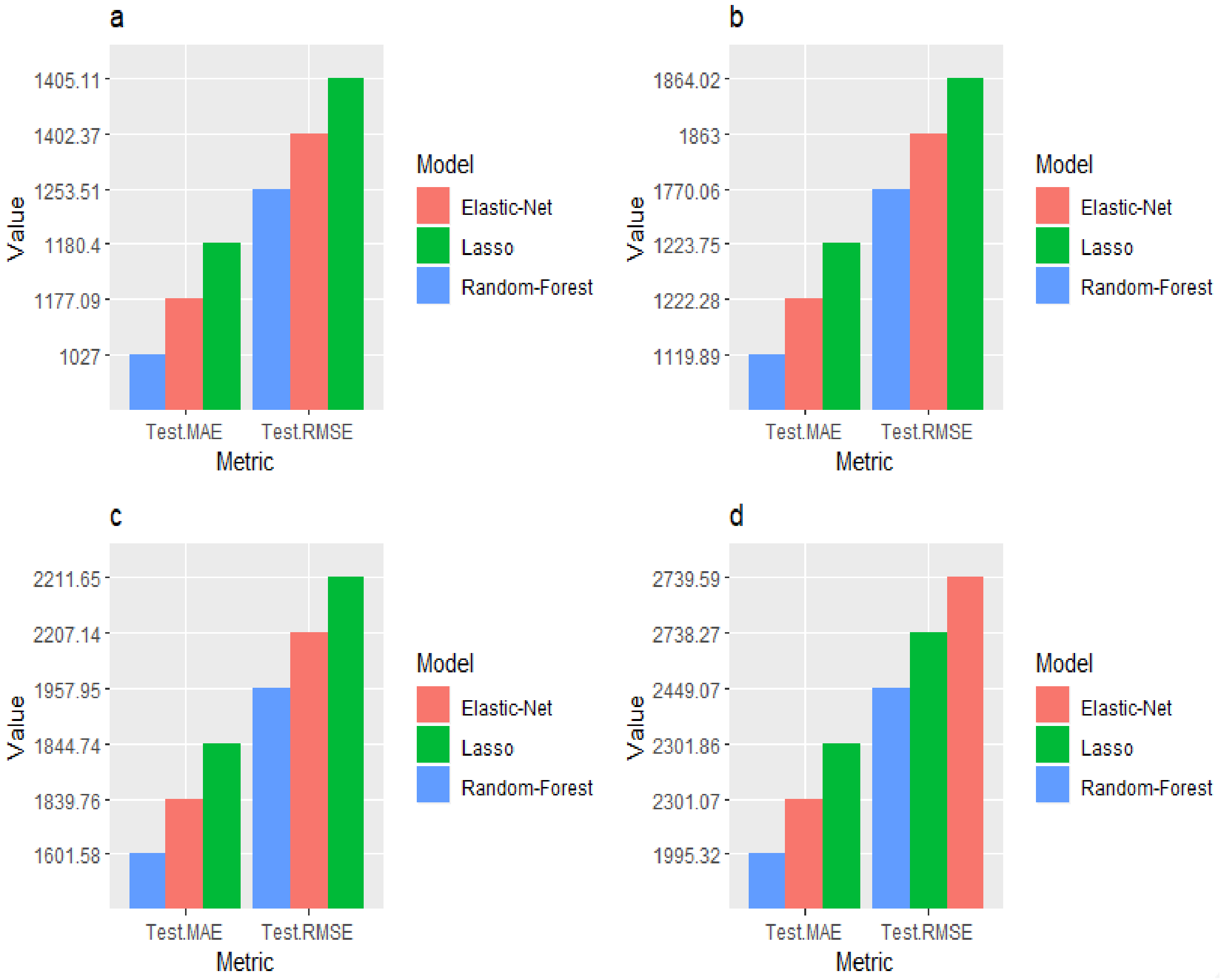


Figure 5.5: Bar plots of the MSE and MAE for the tow race types and performance measures

5.4 DISCUSSION OF PERFORMANCE METRICS

The prediction accuracy of the random forest outlined by the RMSE and the MAE of the random forest outperforms that of the LASSO and elastic net model as seen on Figure 5.2.1 the latter is dominated by the elastic net models while the LASSO models are the worst performing models in predicting both finish times and race paces of half and ultra-marathon runners. The three fitted models seem to predict race pace better than finish time, this is depicted by lower RMSEs and MAEs for such models. While the three fitted models seem to be performing better in predicting finish time for the half-marathon runners, this is not true when predicting race pace. the models can predict race paces for runners in the ultra-marathon better than those for runners in the half marathon race. this is observed when comparing the performance measures. the MSE and MAE differences between the elastic net model and the random forest is quite larger than the differences between the two shrinkage methods even though in general terms, the ranges of these metrics are not large. Even though the random forest models have the least test errors, they produce the biggest differences between the test and training errors relatively to the LASSO and elastic net models while the elastic net models produce the least differences.

CHAPTER 6
CONCLUSION AND RECOMMENDATIONS

The first question addressed in the study was to see if there are significant difference in the response variables per age and gender respectively. In both race types, there appears to be greater evidence that male runners had superior finish times and racing paces than female runners. There is also evidence in the exploratory data analysis indicating that for both race types, there are modest to large disparities in finish time and race pace performances between the four age groups. The general trend however, is that younger runners perform better than older runners, which is usually expected. There's also some evidence that more years of recreational and distance running, particularly among women, is linked to faster race times. A recommendation would be to separate the age groups into more groups with smaller age ranges and then to further split the genders. The goal here would be to attempt to group together those athletes that have similar physiological characteristics. This will allow the coaches to see how these groups react to different training load factors without the worry that maybe results are influenced by differences in age or gender.

There is also proof that runners who have spent more time in recreational running have faster finish times than runners who have spent less time in recreational running, especially for female runners. This may seem counter-intuitive considering that age and experience are move in the same direction but when looking at athletes that have the same age but have different experience, those with more experience tend to perform better. A recommendation to athletes who plan to pursue distance running would be to start training seriously as early in their lives as possible and also to participate in as many events as they can in order to gain experience. Special care, however, must be taken to ensure that athletes do not train or perform outside of their physical capabilities or to the extent where by it becomes detrimental to their development.

Evidence in chapter 2 suggests that athletes who run for greater distances and more times per week appear to have better finish times and racing speeds, but training on tar and brick tracks has no link to performance. An important consideration is the ‘nature vs nurture’ debate. Put simply, while it is hard to estimate the magnitude of the impact of nature (genes) and nurture (environment) on sports performance, it is irrational to disregard either as a fundamental element. The transition of a gifted athlete into a superstar is commonly thought to be the outcome of a dynamic mixture of several factors. These components are heredity, practice, and preparation in their simplest form. Each of those characteristics is critical to the development of a top performer(Georgiades *et al.*, 2017). There seems to be an accentuation on the counting of practice hours for example, and there has rarely been an effort for a synthesis of ideas which are characterized in Figure 6.1. As a result, going forward, emphasis should place on enhancing the training value of specific types of training by better understanding how athletes perform and identifying training principles and approaches that help accelerate an athlete's development(Baker *et al.*, 2017). Data on the athletes' performance, training regimens, diet and genetics should be accurately tracked and collected in order to do meaningful statistical inference.

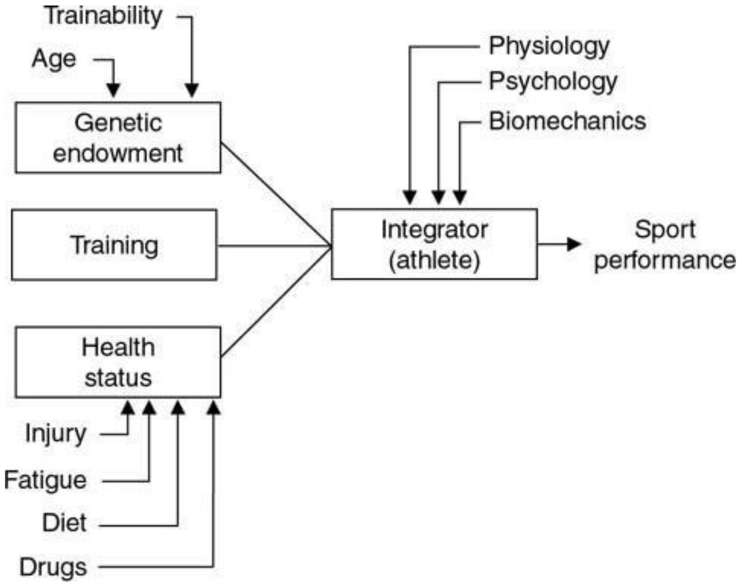


Figure 6.1: Factors that influence athletic performance

Source: MacDougall, Wenger & Green (1991)

Recommendations

i. Shorter and precise questionnaires rather than general long ones to avoid loss of much information as people get exhausted halfway and some prefer not to give consent at all.

ii. Coaching needed for runners during training due to the following reasons:

- Separation of individuals into groups based on their performance abilities so that such groups can be assed separately.
- Some individuals may seem like they have a high training load but due to being inexperienced in proper ways of training, their training efforts tend to be less beneficial.

REFERENCES

1. Bacon, A.P., Carter, R.E., Ogle, E.A. & Joyner, M.J. 2013. VO2max Trainability and High Intensity Interval Training in Humans: A Meta-Analysis. *PLOS ONE*. 8(9):e73182. [Online], Available: https://doi.org/10.1371/journal.pone.0073182.
2. Baker, J., Cobley, S., Schorer, J. & Wattie, N. 2017. *Routledge handbook of talent identification and development in sport*.
3. Bompa, T.O. & Buzzichelli, C. 1973-. 2019. *Periodization : theory and methodology of training LK - https://sun.on.worldcat.org/oclc/1019739245*. Sixth edit ed. Champaign, IL SE - ix, 381 pages : illustrations ; 29 cm: Human Kinetics.
4. Bouchard, C., Leon, A.S., Rao, D.C., Skinner, J.S., Wilmore, J.H. & Gagnon, J. 1995. The HERITAGE family study. Aims, design, and measurement protocol. *Medicine and science in sports and exercise*. 27(5):721–729.
5. Budgett, R. 1998. Fatigue and underperformance in athletes: the overtraining syndrome. *British journal of sports medicine*. 32(2):107–110.
6. Curran, P.J., West, S.G. & Finch, J.F. 1996.
7. Foster, C., Daines, E., Hector, L., Snyder, A.C. & Welsh, R. 1996. Athletic performance in relation to training load. *Wisconsin medical journal*. 95(6):370–374.
8. George, D. & Mallery, P. 2003a. SPSS for Windows Step by Step A Simple Guide. 386. [Online], Available: http://wps.ablongman.com/wps/media/objects/1577/1615609/george5answers.pdf.
9. George, D. & Mallery, P. 2003b. SPSS for Windows Step by Step A Simple Guide. 386.
10. Georgiades, E., Klissouras, V., Baulch, J., Wang, G. & Pitsiladis, Y. 2017. Why nature prevails over nurture in the making of the elite athlete. *BMC Genomics*. 18(8):835.
11. Hopkins, W.G. 2001. Genes and training for athletic performance. *Sportscience*. 5(1):1–7. [Online], Available: https://sportsci.org/jour/0101/wghgene.pdf.
12. Kallus, K.W. 1992. *Beanspruchung und Ausgangszustand*. Psychologie-Verlag-Union.
13. MacDougall, J.D., Wenger, H.A. & Green, H.J. 1991. The purpose of physiological testing. *Physiological testing of the highperformance athlete (2nd edn., pp. 1–5). Champaign, IL: Human Kinetics*.

14. Martin, D.E., Vroon, D.H., May, D.F. & Pilbeam, S.P. 1986. Physiological Changes in Elite Male Distance Runners Training for Olympic Competition. *The Physician and Sportsmedicine*. 14(1):152–206.
15. Mikesell, K.A. & Dudley, G.A. 1984. Influence of intense endurance training on aerobic power of competitive distance runners. *Medicine and science in sports and exercise*. 16(4):371–375.
16. Pallant, J. 2010. *SPSS survival manual : a step by step guide to data analysis using SPSS*. Fourth edition. Maidenhead : Open University Press/McGraw-Hill, 2010.
17. Smith, D.J. 2003. A Framework for Understanding the Training Process Leading to Elite Performance. *Sports Medicine*. 33(15):1103–1126.
18. Tabachnick, B.G. 1936- & Fidell, L.S. 2007. *Using multivariate statistics LK - https://sun.on.worldcat.org/oclc/62766132*. 5th ed. ed. Boston SE - xxviii, 980 pages : illustrations ; 25 cm: Pearson/Allyn & Bacon. [Online], Available: http://bvbr.bib-bvb.de:8991/F?func=service&doc_library=BVB01&doc_number=016738385&line_number=0001&func_code=DB_RECORDS&service_type=MEDIA.
19. Tibshirani, R., 2011. Regression shrinkage and selection via the lasso: a retrospective. *Journal of the Royal Statistical Society ,* 73(3), pp. 273-282.

APPENDIX A:
TABLES AND FIGURES

A.1 RAW DATA, DATA CLEANING AND SUMMARY STATISTICS

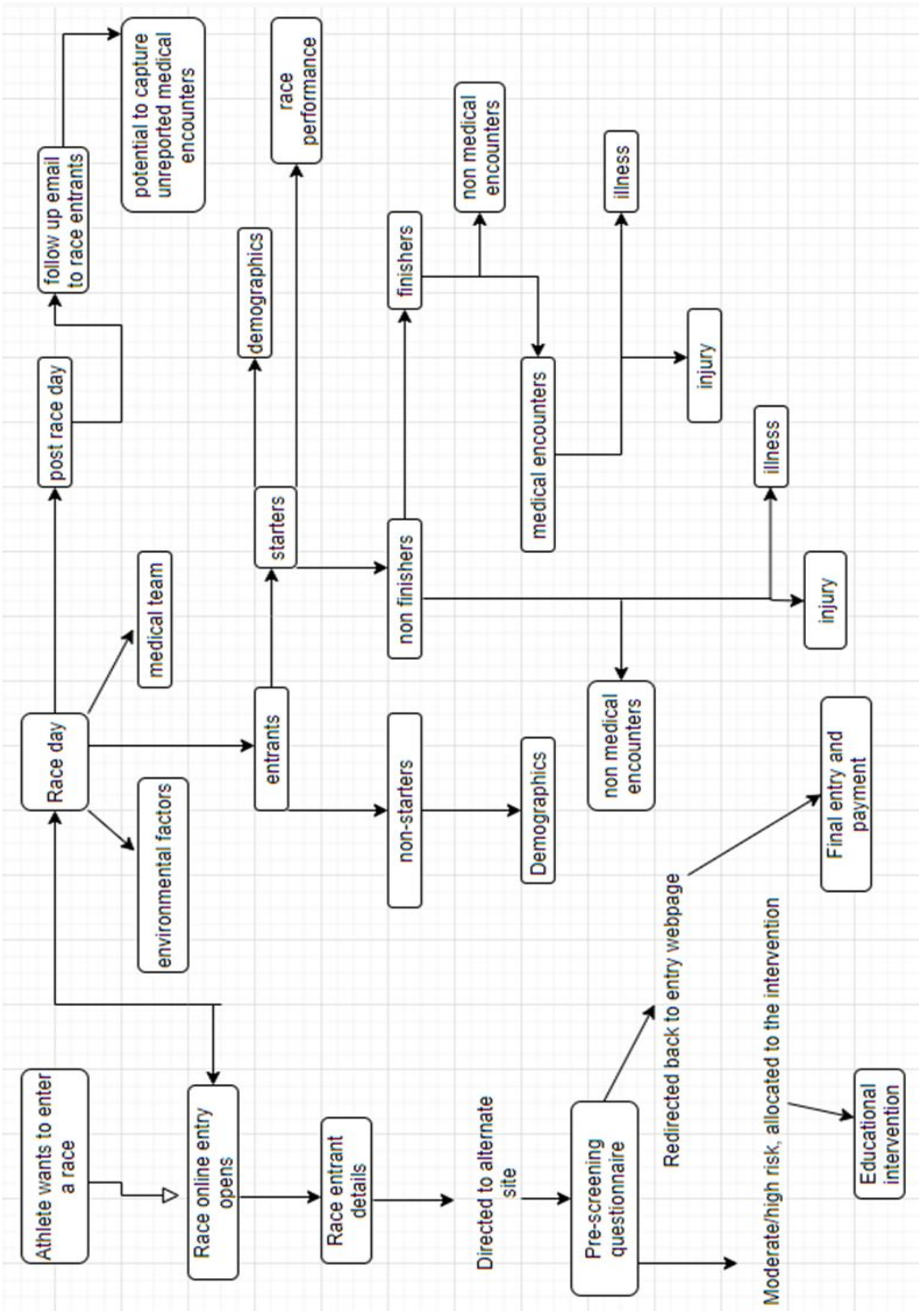


Figure A.1: Flow diagram of research methodology

Table A.1: Summary statistics for the training load variables

Statistic	Recreational runner(years)	Distance runner (Years)	Times train race (Years)	Training distance (Years)	Training pace(mins)	Treadmill (% time)	Tar/brick (% time)	Gravel (% time)	Race pace(mins)	Finish time(mins)
Minimum	0.500000	0.500000	2.000000	10.00000	2.000000	-1.0000000	-1.0000000	-1.0000000	3.056872	64.5000
Maximum	70.000000	58.000000	8.000000	110.00000	13.000000	1.0000000	1.0000000	1.0000000	22.534755	684.0000
Mean	9.240819	6.654200	3.706246	36.39029	5.885336	-0.1811341	0.1973464	-0.1717236	6.604883	226.8348
Median	6.000000	4.000000	4.000000	30.00000	6.000000	0.0000000	0.5000000	0.0000000	6.627083	165.7833
Lower Quartile	3.000000	2.000000	3.000000	20.00000	5.250000	-1.0000000	-1.0000000	-1.0000000	5.917857	138.1500
Upper Quartile	13.000000	10.000000	4.000000	50.00000	6.500000	0.1000000	0.8000000	0.2000000	7.290774	347.5333
Inter Quartile Range	10.000000	8.000000	1.000000	30.00000	1.250000	1.1000000	1.8000000	1.2000000	1.372917	209.3833
Standard Deviation	8.438109	7.195082	1.137360	19.46293	1.062317	0.5412242	0.7818899	0.5406904	1.008494	110.4777

Table A.2: Summary of all participants who started the race and those that did not start the race.

race type	Did not start the race		Total
	no	yes	
Ultra	34438	7565	42003
Half	52905	11835	64740
Total	87343	19400	106743
Percentage	81.83	18.17	100

Table A.3: Summary of study participants who finished the race and those that did not finish the race.

Table of study participants			
race type	Finish the race		Total
	no	yes	
Ultra	22964	1032	23996
Half	38843	13	38856
Total	61807	1045	62852
Percentage	98.34	1.66	100

Table A.4: Summary of all participants who finished the race and those that did not finish the race.

race type	Finished the race		Total
	no	yes	
Ultra	32874	1564	34438
Half	52883	22	52905
Total	85757	1586	87343
Percentage	98.18	1.82	100

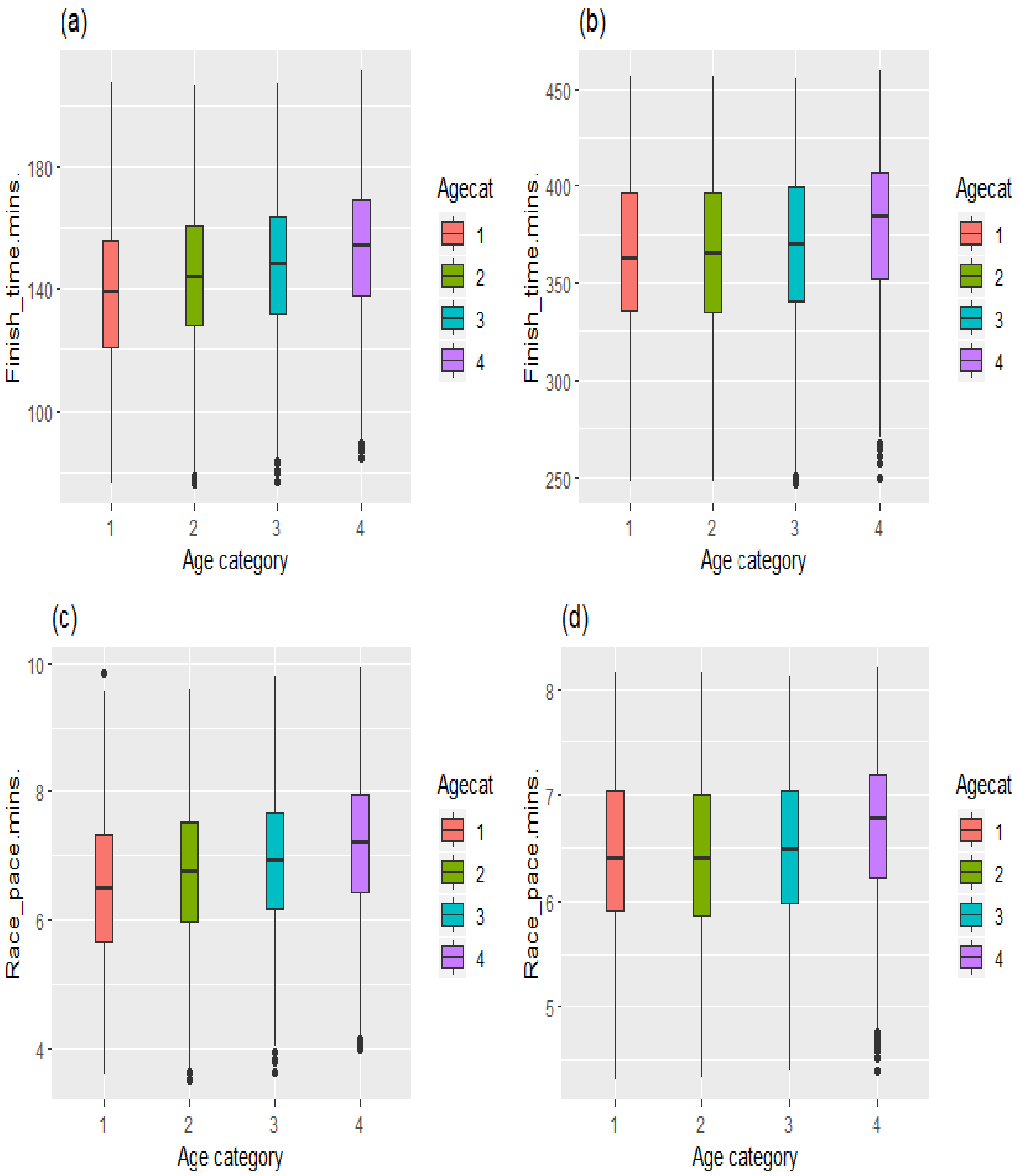


Figure A.2: Boxplots of age categories for finish time and race pace post removal of outliers.

A.2 EXPLORATORY DATA ANALYSIS PLOTS

Finish time

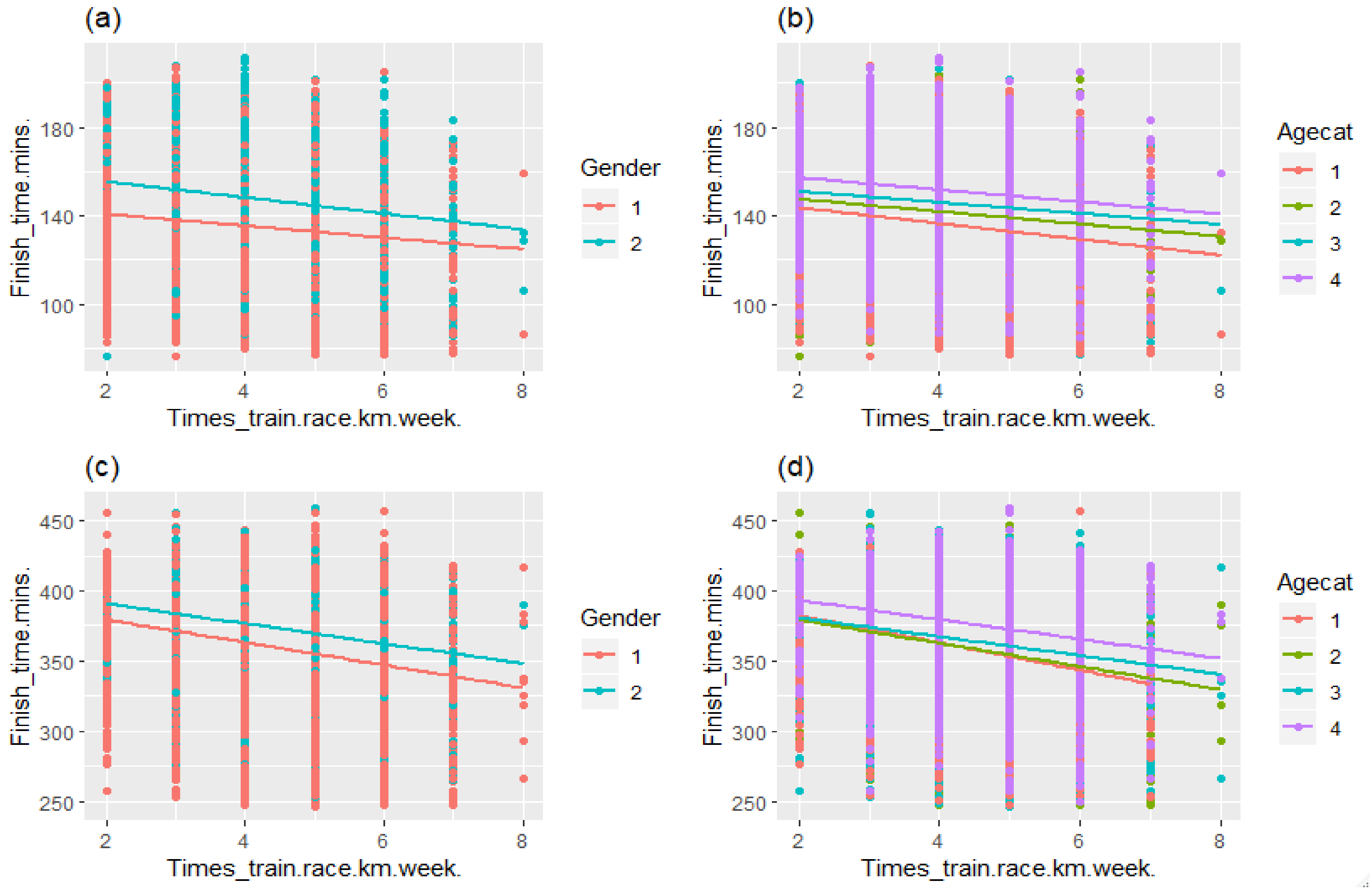


Figure A.3: Scatterplots for race training times per week and finish time. There are two gender categories, the orange scatter points represent male runners while the blue scatter points represent female runners. The right panel represents the four age categories 1,2,3 and 4 with orange, green, blue and purple scatters respectively plotted against finish time in minutes.

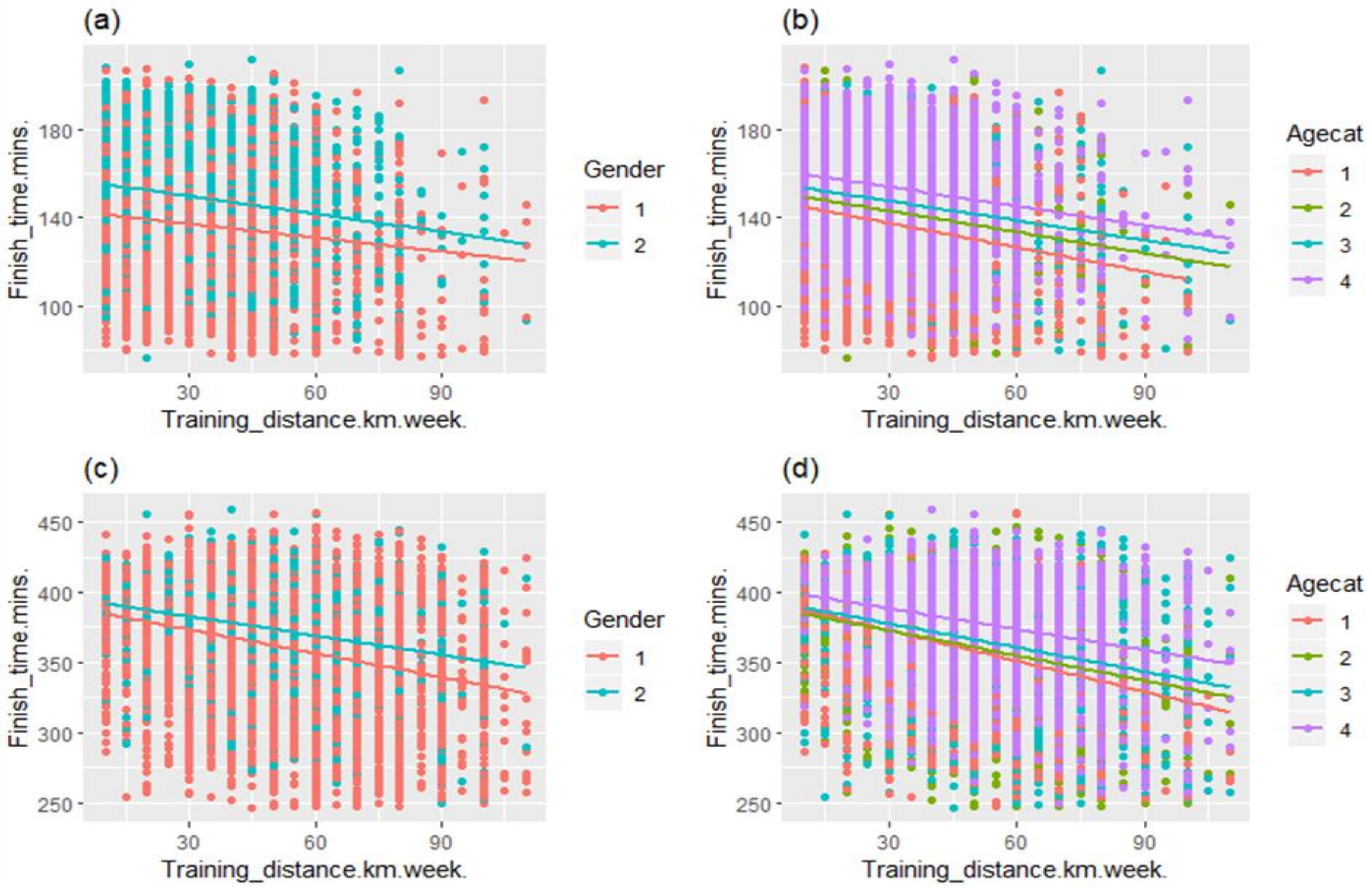


Figure A.4: Scatterplots for training distance per week and finish.

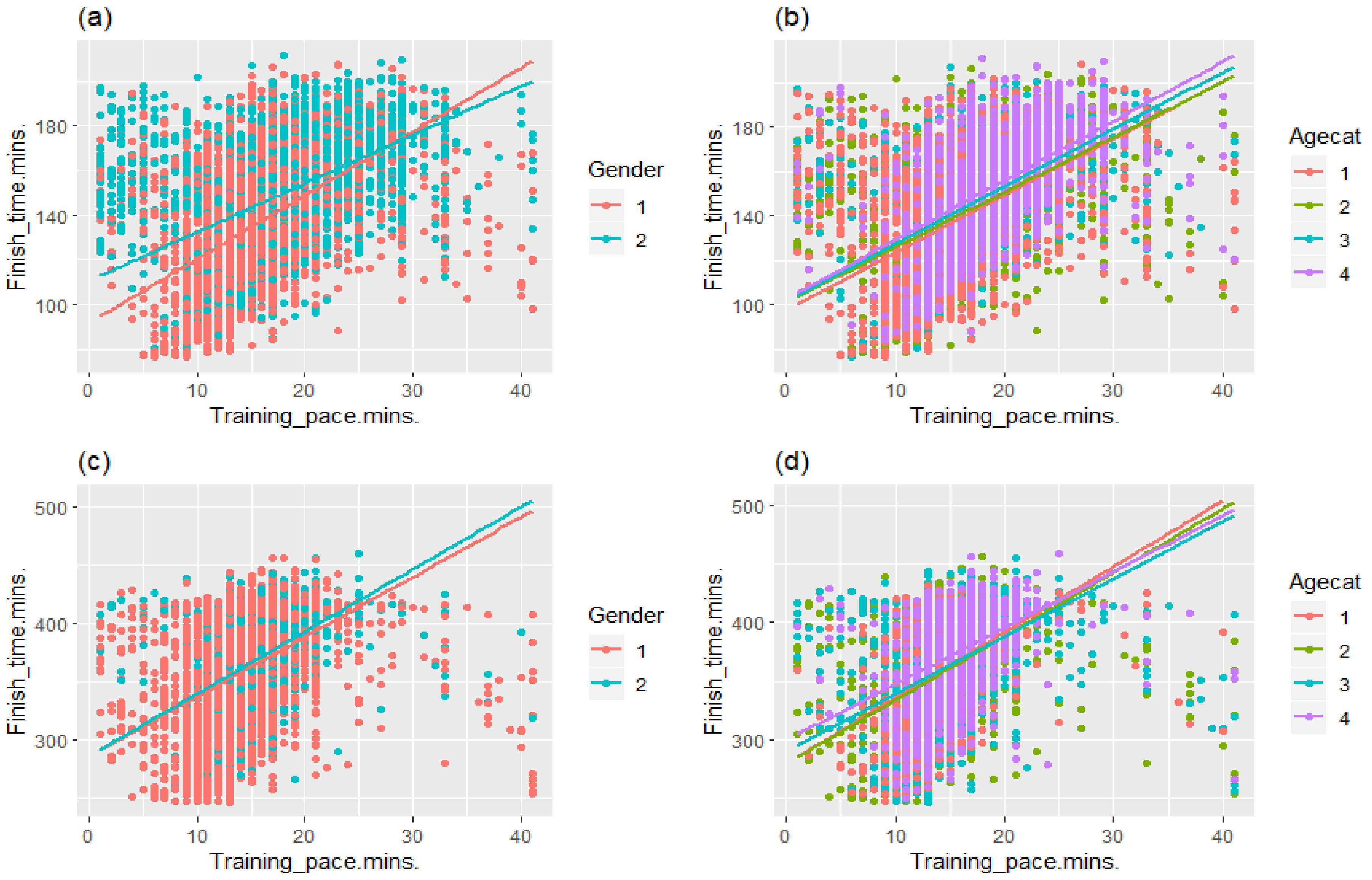


Figure A.5: Scatterplot of finish time and training race

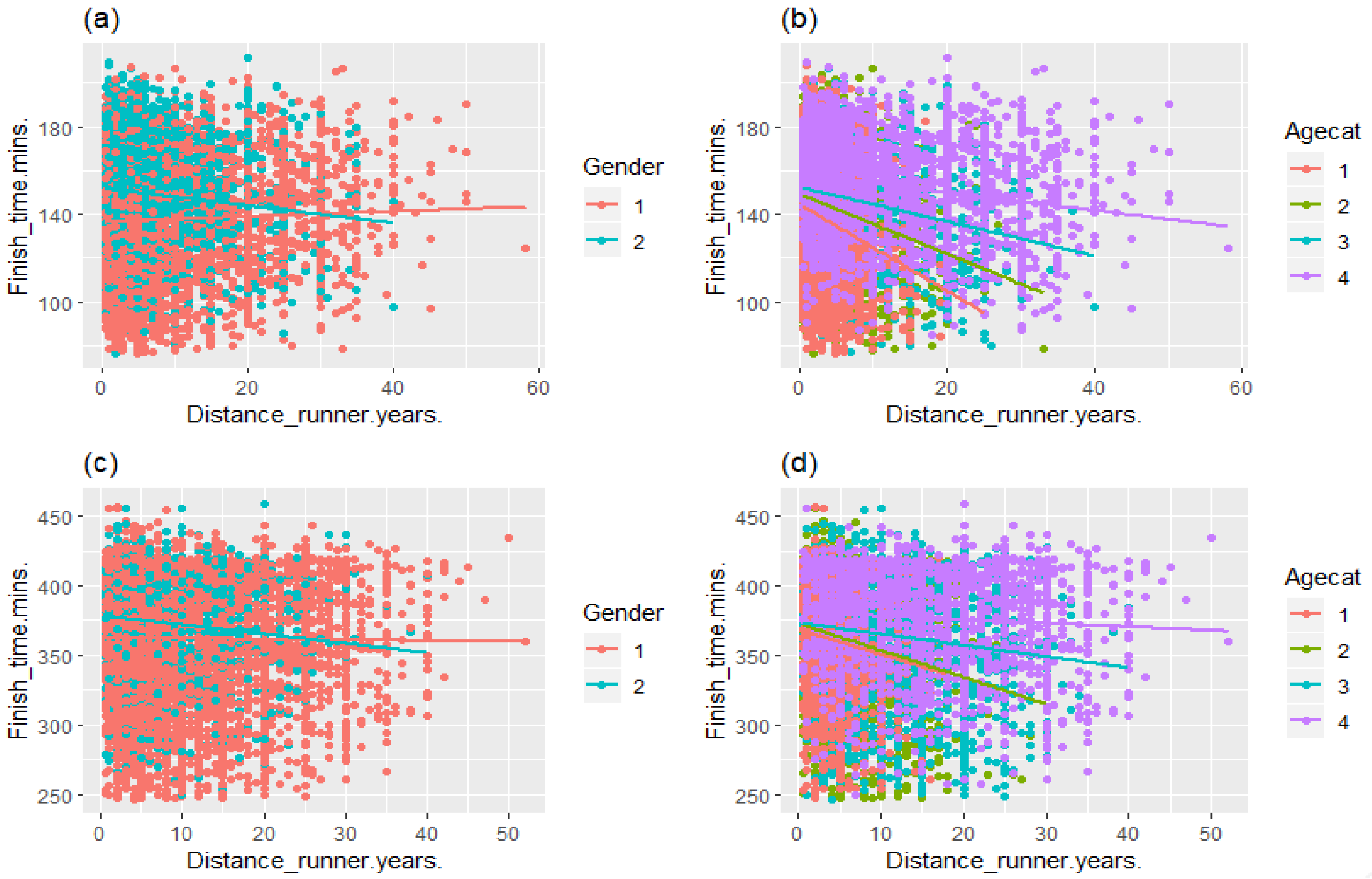


Figure A.6: Scatterplot of Finish time and Distance running years

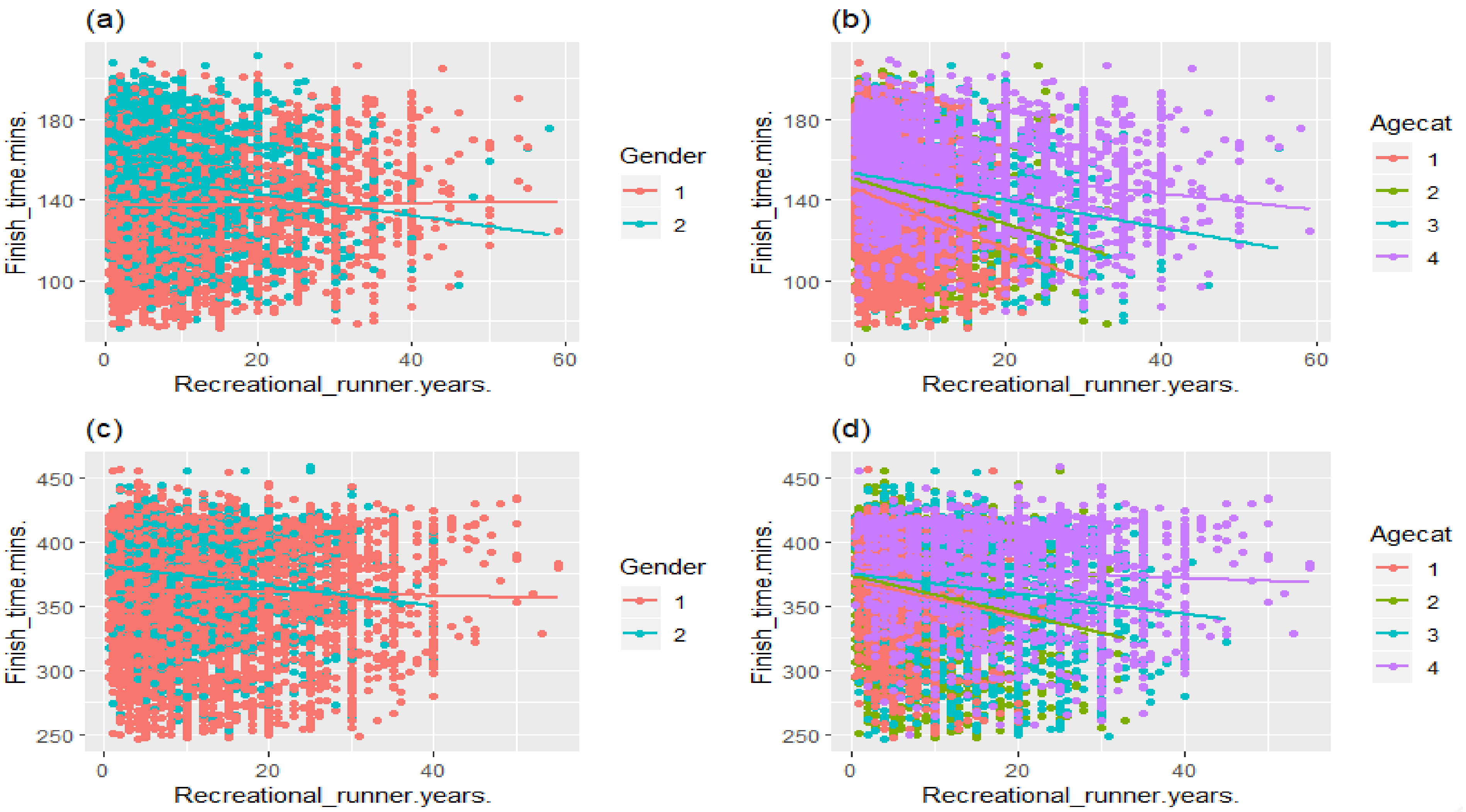


Figure A.7: Scatterplot of Finish time and recreational running years

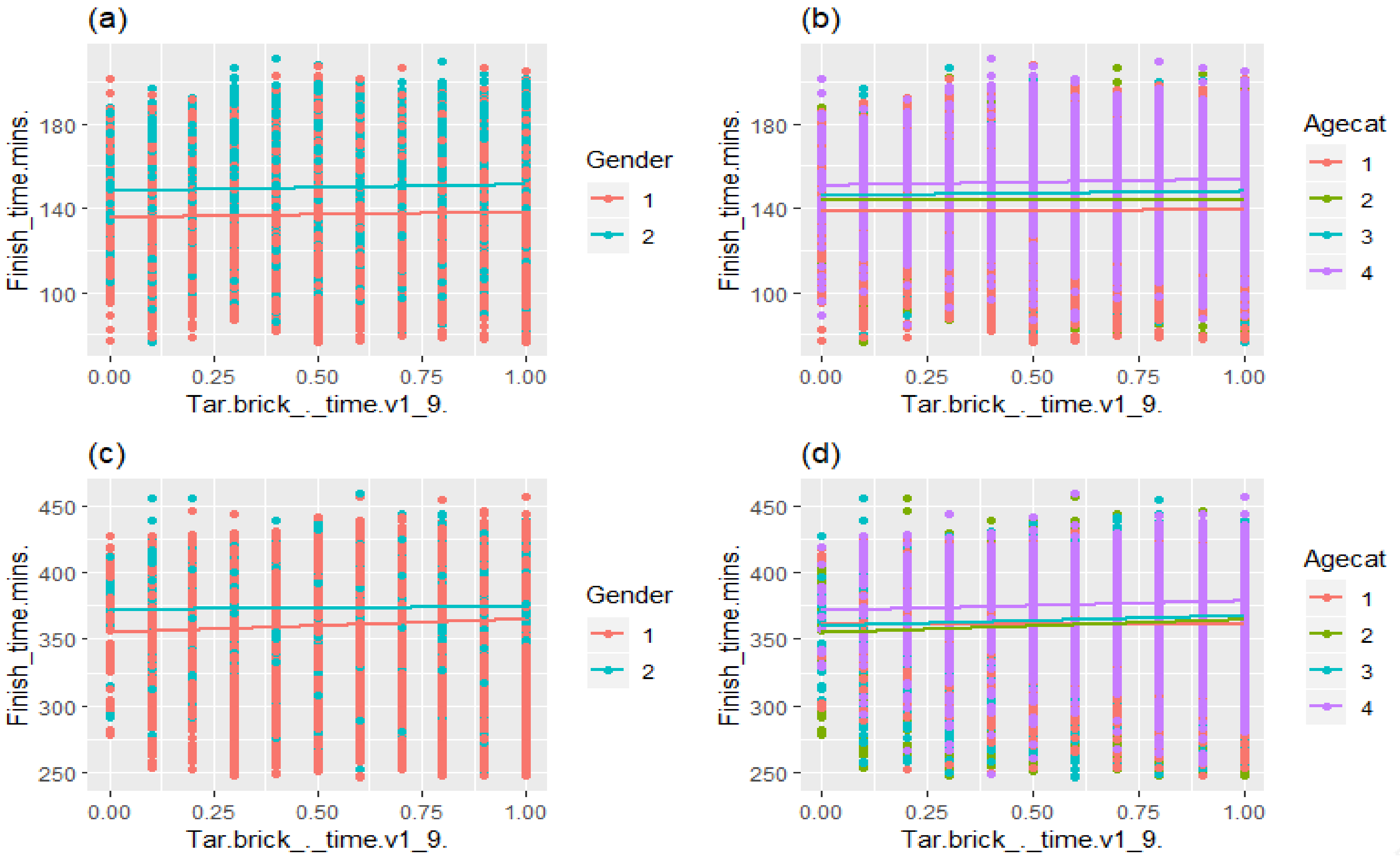


Figure A.8: Scatterplot of Finish time and tar/brick racing percentage time

Race pace

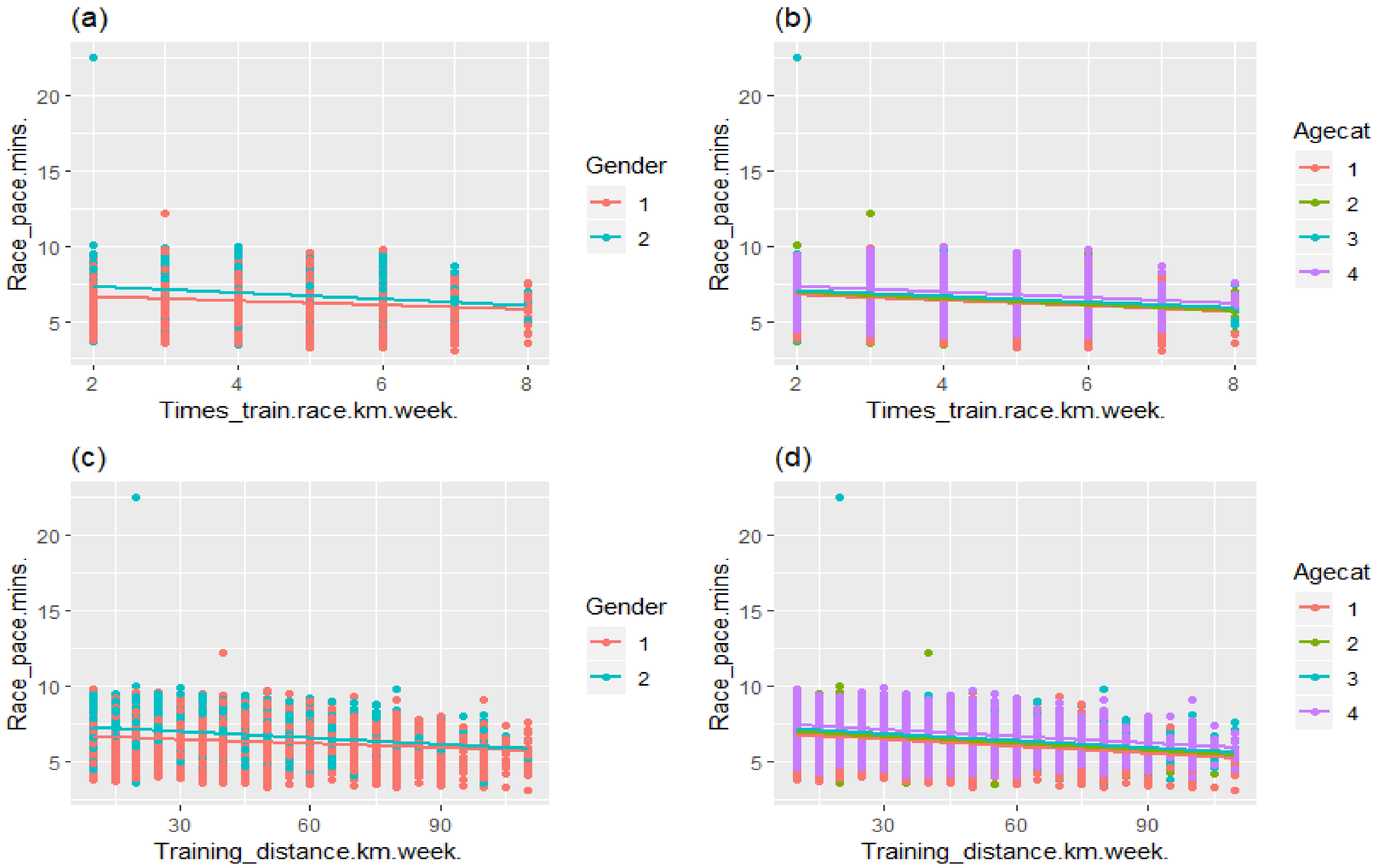


Figure A.9: Panel (a) is a scatterplot of race pace in minutes and times train race per week per gender category, panel (b) is a scatterplot of race pace and times train race per week per age category, panel (c) shows the scatterplot of race pace and training distance per week per gender category while panel (d) is a scatterplot of race pace and training distance per week per gender category.

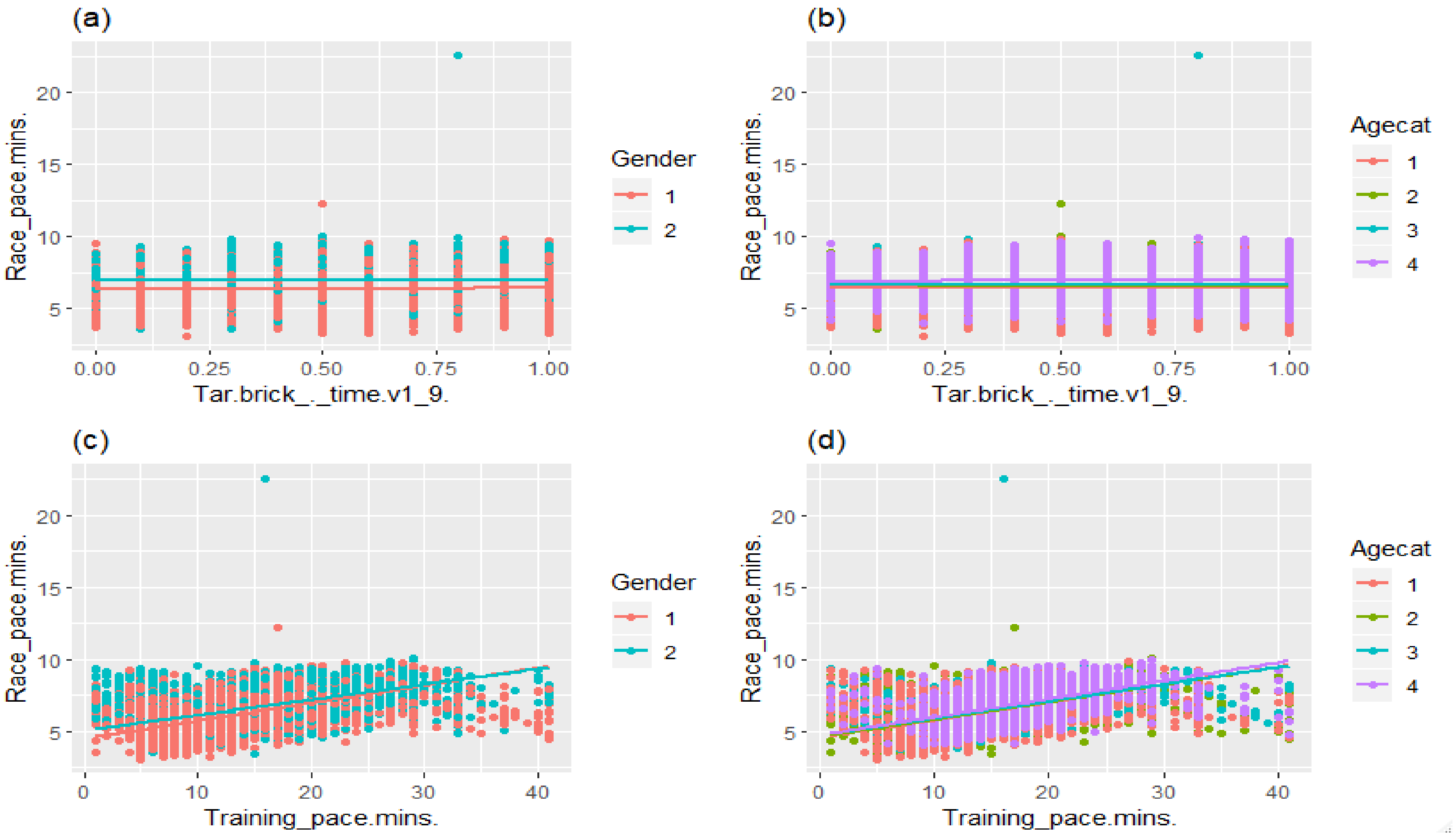


Figure A.10: Panel (a) is a scatterplot for race pace and tar/brick percentage racing times per gender category. Panel (b) is a scatterplot for race pace and tar/brick percentage racing times per age category, panel (c) is a scatterplot for race pace and training pace per gender category while panel (d) is a scatterplot for race pace and training pace per age category.

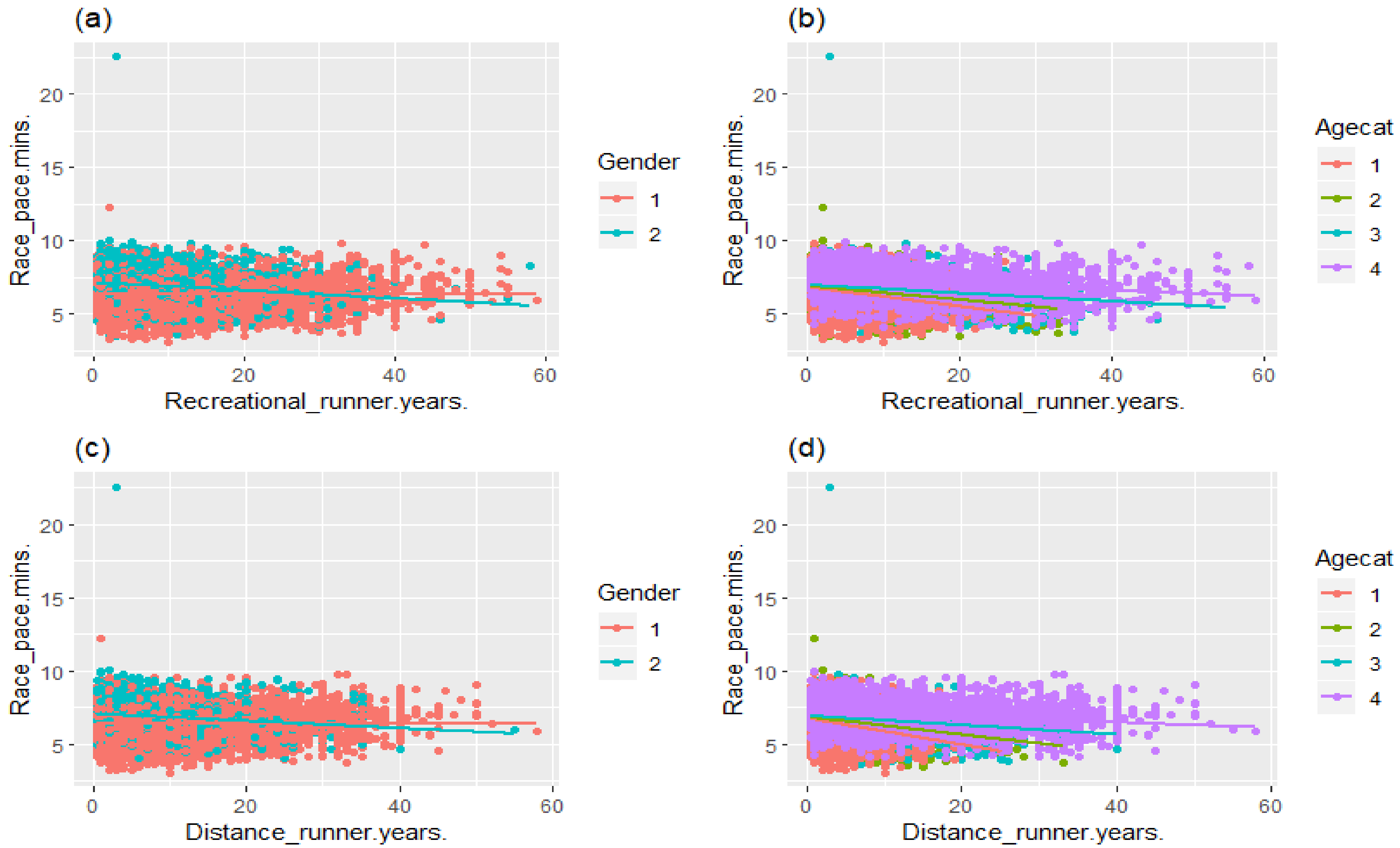


Figure A.11: Panel (a) is a scatterplot for race pace and Recreational running years per gender category. Panel (b) is a scatterplot for race pace and Recreational running per age category, panel (c) is a scatterplot for race pace and training pace per gender category while panel (d) is a scatterplot for race pace and training pace per age category.

A.3 MODELLING

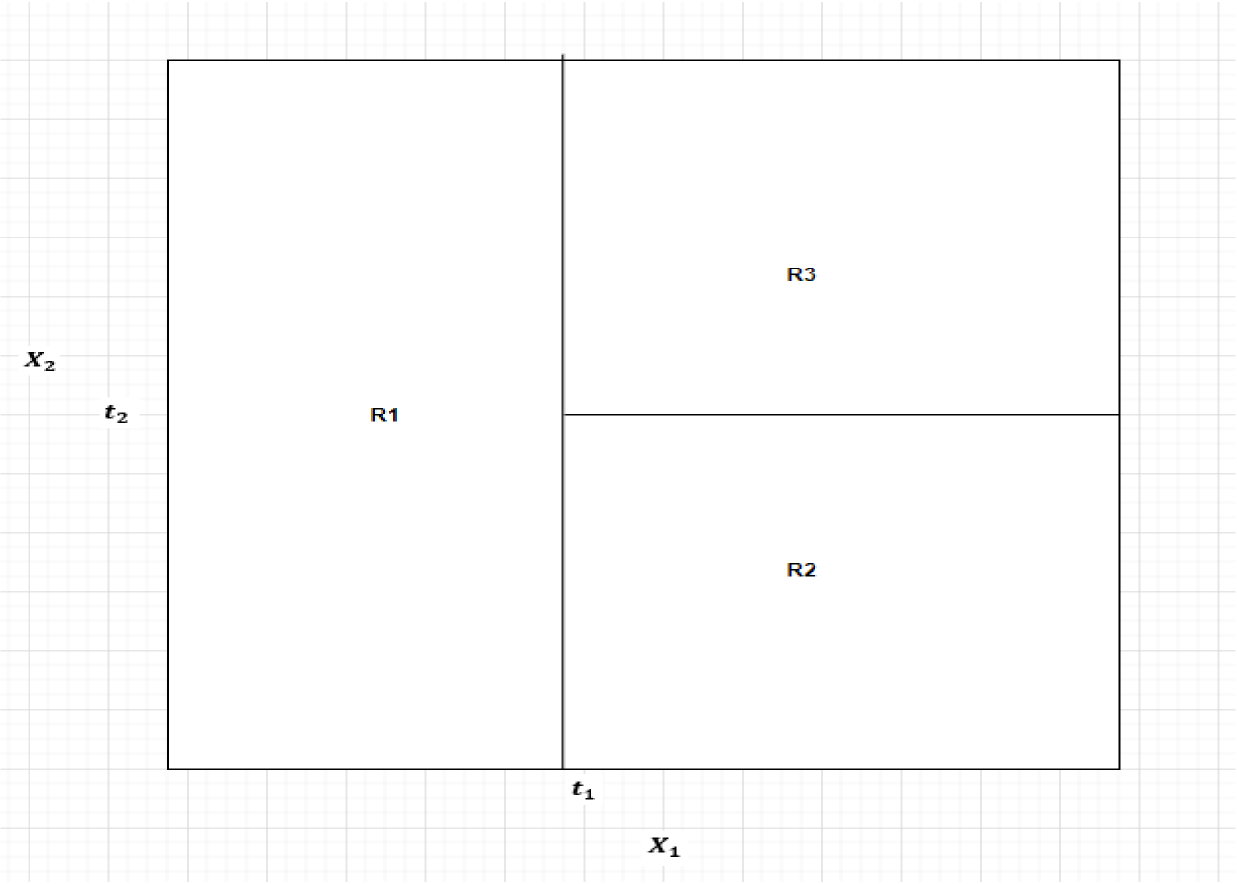


Figure A.12: Regions formed by a regression tree

A.4 MISSING DATA PERFORMANCE METRICS (MISS FOREST IMPUTED DATA 80%/20% TRAIN-TEST DATA PARTITION)

Table A.5: Training and test RMSEs for the three fitted models under the half and ultra-marathon race types with the percentage increase of train-test RMSE.

Race type	Response variable	Model	Test RMSE	Train RMSE	% increase
Half-marathon	**Finish time**	**LASSO**	**1405.11**	**1405.4**	**-0.02**
		Elastic-Net	**1402.37**	**1401.93**	**0.03**
		Random-Forest	**1253.51**	**642.03**	**95.24**
Ultra-Marathon		**LASSO**	**1864.02**	**1897.91**	**-1.79**
		Elastic-Net	**1863.0**	**1897.57**	**-1.82**
		Random-Forest	**1770.06**	**920.21**	**92.35**
Half-marathon	**Race pace**	**LASSO**	**2211.65**	**2212.31**	**-0.03**
		Elastic-Net	**2207.14**	**2206.91**	**0.01**
		Random-Forest	**1957.95**	**1005.04**	**94.81**
Ultra-Marathon		**LASSO**	**2738.27**	**2740.41**	**-0.08**
		Elastic-Net	**2739.59**	**2738.68**	**0.03**
		Random Forest	**2449.07**	**1250.70**	**95.81**

Table A.6: Training and test MAEs for the three fitted models under the half and ultra-marathon race types with the percentage increase of train-test MAE.

Race type	Response variable	Model	Test MAE	Train MAE	% increase
Half-marathon	**Finish time**	**LASSO**	**1180.4**	**1172.22**	**0.7**
		Elastic-Net	**1177.09**	**1169.1**	**0.68**
		Random-Forest	**1027.0**	**519.25**	**97.78**
Ultra-Marathon		**LASSO**	**1223.75**	**1240.25**	**-1.33**
		Elastic-Net	**1222.28**	**1239.2**	**-1.37**
		Random-Forest	**1119.89**	**570.88**	**96.17**
Half-marathon	**Race pace**	**LASSO**	**1844.74**	**1831.52**	**0.73**
		Elastic-Net	**1839.76**	**1826.85**	**0.71**
		Random-Forest	**1601.58**	**809.86**	**97.76**
Ultra-Marathon		**LASSO**	**2301.86**	**2305.0**	**-0.14**
		Elastic-Net	**2301.07**	**2302.67**	**-0.07**
		Random Forest	**1995.32**	**1016.23**	**96.35**

A.5 ANOVA ASSUMPTIONS

Testing for normality

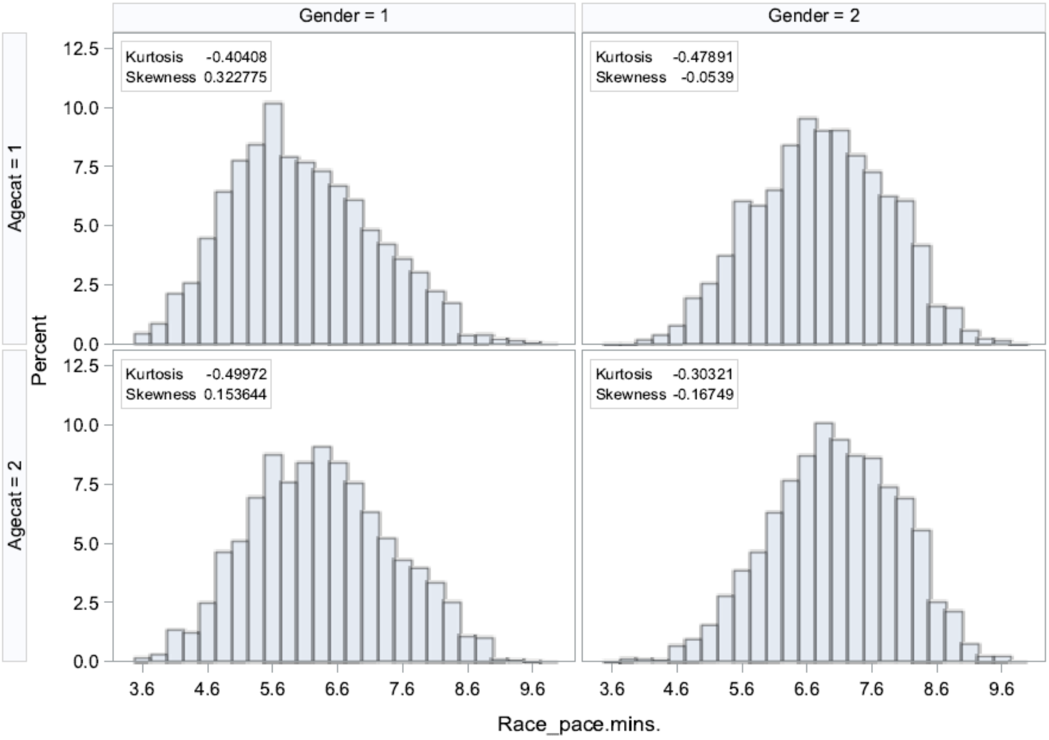


Figure A.13: Race pace histograms comparing gender and age categories 1 and 2

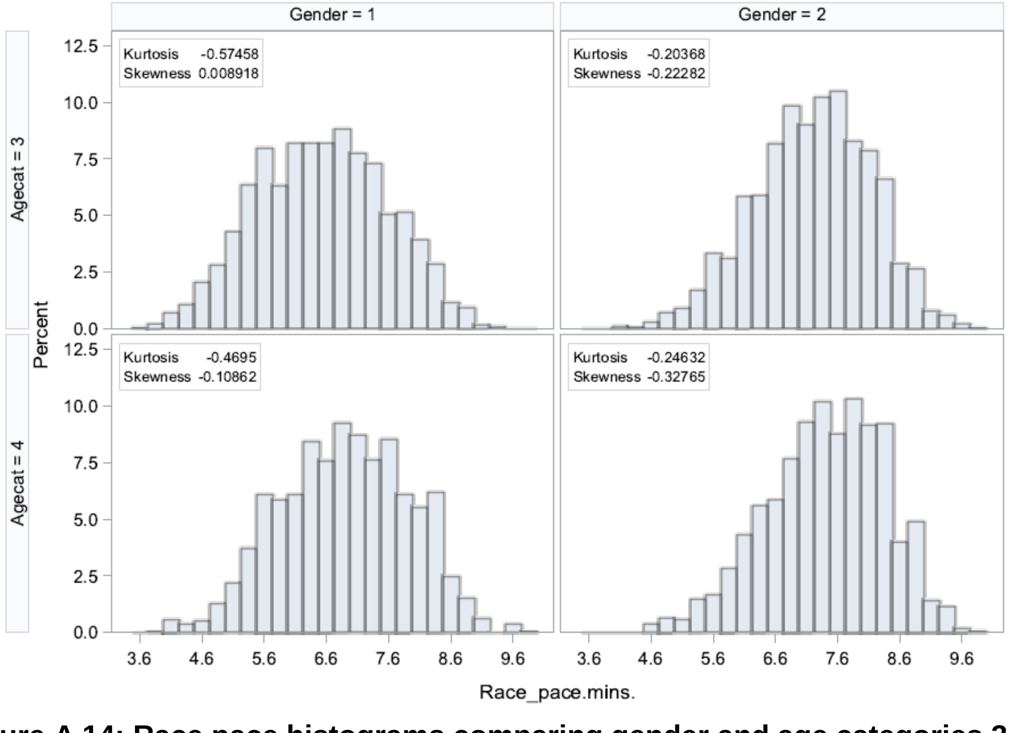


Figure A.14: Race pace histograms comparing gender and age categories 3 and 4

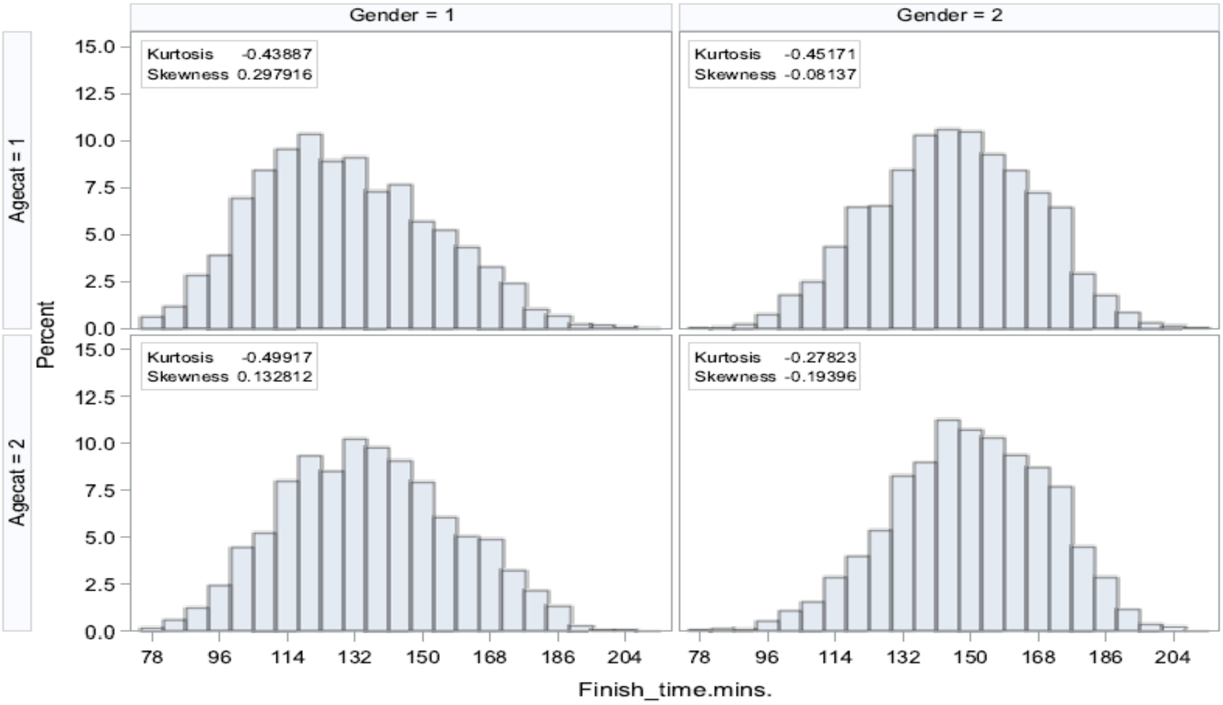


Figure A.15: Finish time histograms comparing gender and age categories 1 and 2

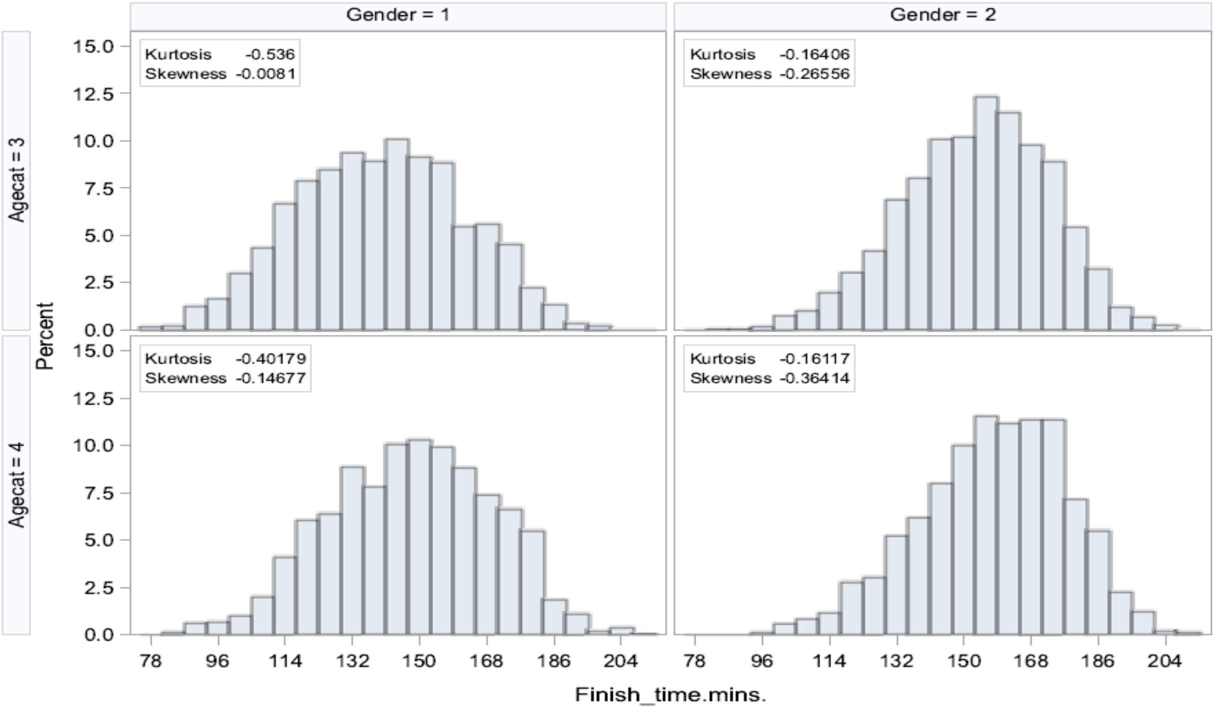
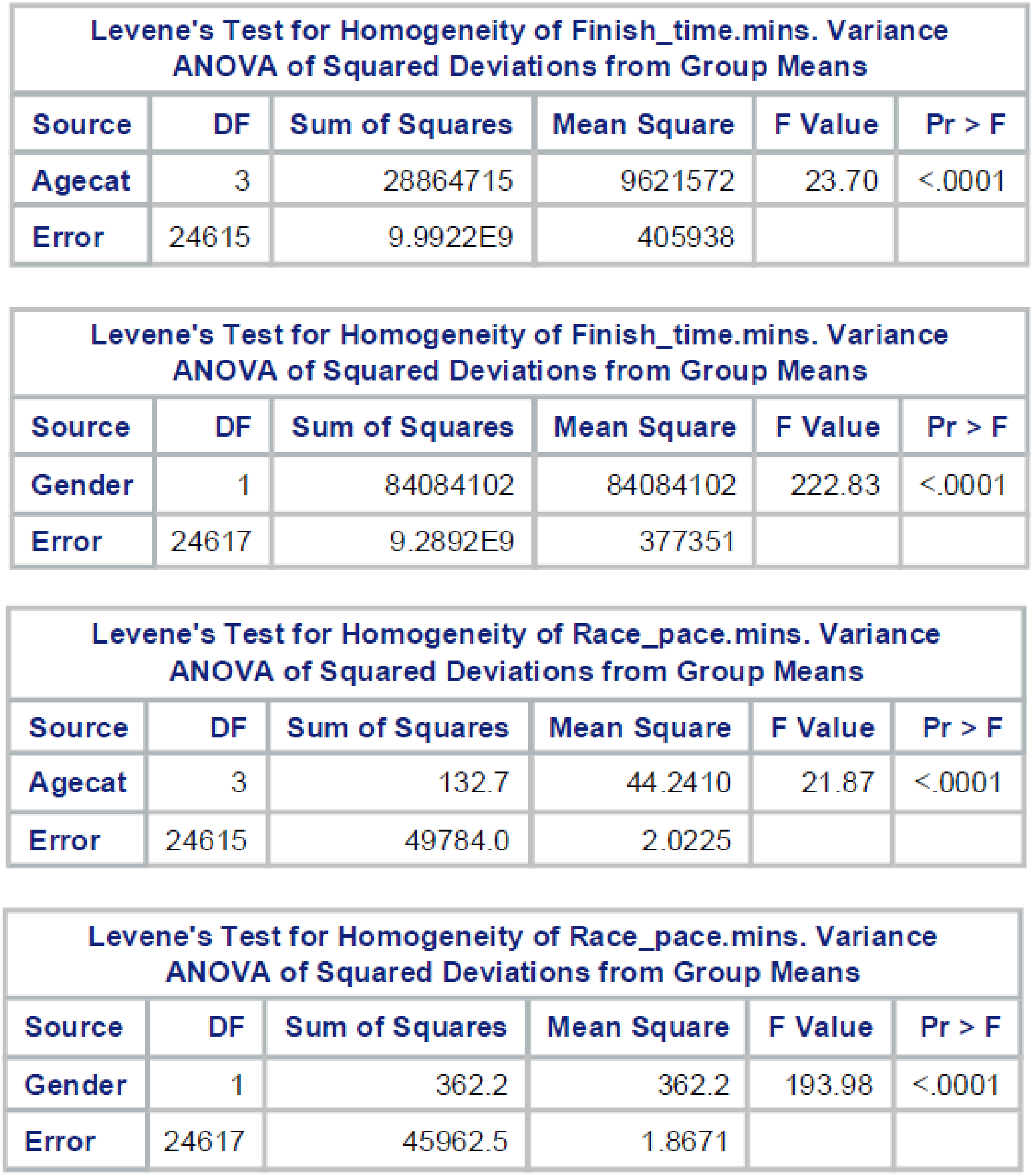


Figure A.16: Finish time histograms comparing gender and age categories 3 and 4

Testing for homoscedasticity



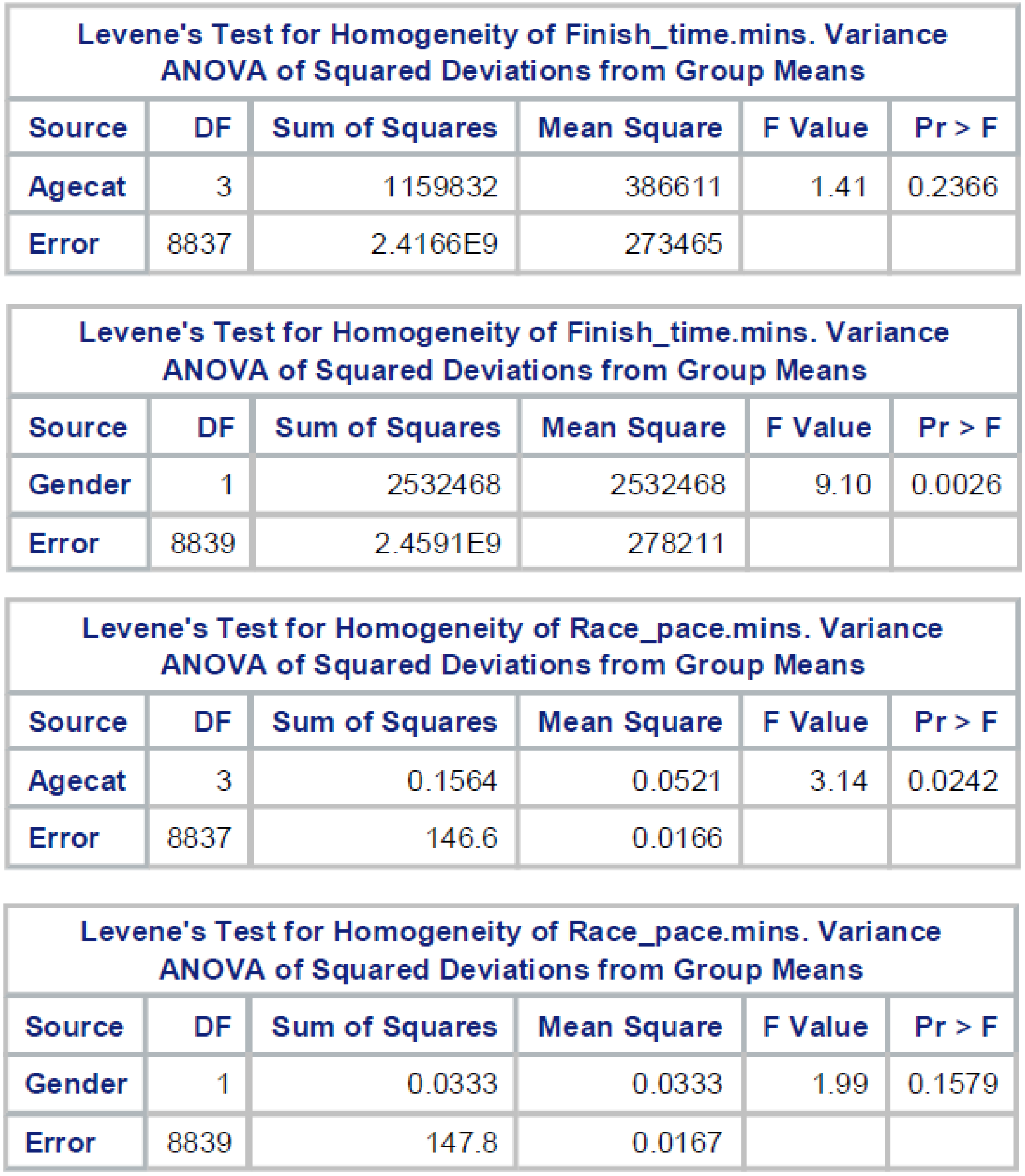
Levene's Test for Homogeneity of Finish_time.mins. Variance ANOVA of Squared Deviations from Group Means					
Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Agecat	3	28864715	9621572	23.70	<.0001
Error	24615	9.9922E9	405938		

Levene's Test for Homogeneity of Finish_time.mins. Variance ANOVA of Squared Deviations from Group Means					
Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Gender	1	84084102	84084102	222.83	<.0001
Error	24617	9.2892E9	377351		

Levene's Test for Homogeneity of Race_pace.mins. Variance ANOVA of Squared Deviations from Group Means					
Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Agecat	3	132.7	44.2410	21.87	<.0001
Error	24615	49784.0	2.0225		

Levene's Test for Homogeneity of Race_pace.mins. Variance ANOVA of Squared Deviations from Group Means					
Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Gender	1	362.2	362.2	193.98	<.0001
Error	24617	45962.5	1.8671		

Figure A.17: Half Marathon Levene's test for Homogeneity



Levene's Test for Homogeneity of Finish_time.mins. Variance
ANOVA of Squared Deviations from Group Means

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Agecat	3	1159832	386611	1.41	0.2366
Error	8837	2.4166E9	273465		

Levene's Test for Homogeneity of Finish_time.mins. Variance
ANOVA of Squared Deviations from Group Means

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Gender	1	2532468	2532468	9.10	0.0026
Error	8839	2.4591E9	278211		

Levene's Test for Homogeneity of Race_pace.mins. Variance
ANOVA of Squared Deviations from Group Means

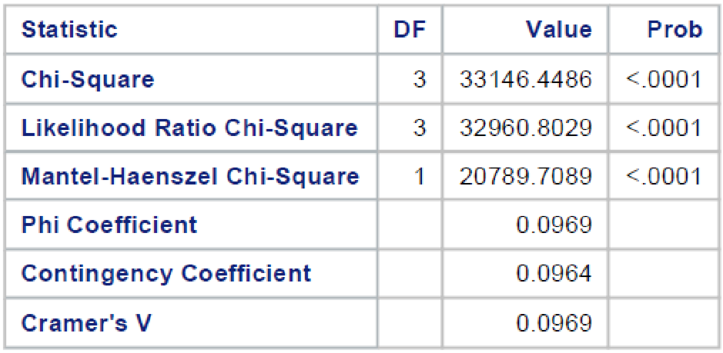
Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Agecat	3	0.1564	0.0521	3.14	0.0242
Error	8837	146.6	0.0166		

Levene's Test for Homogeneity of Race_pace.mins. Variance
ANOVA of Squared Deviations from Group Means

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Gender	1	0.0333	0.0333	1.99	0.1579
Error	8839	147.8	0.0167		

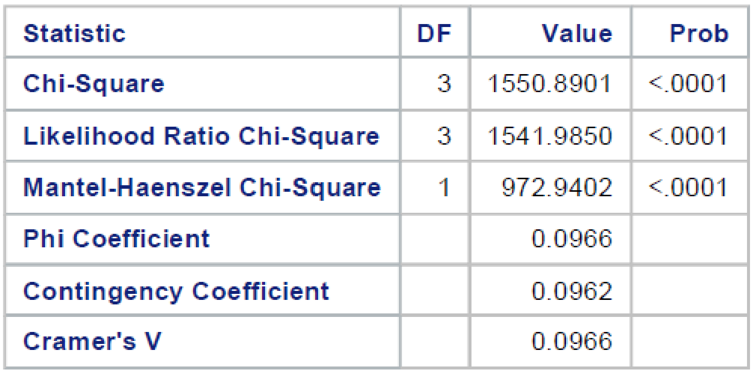
Figure A.18: Ultra Marathon Levene’s test for Homogeneity

Testing for independence



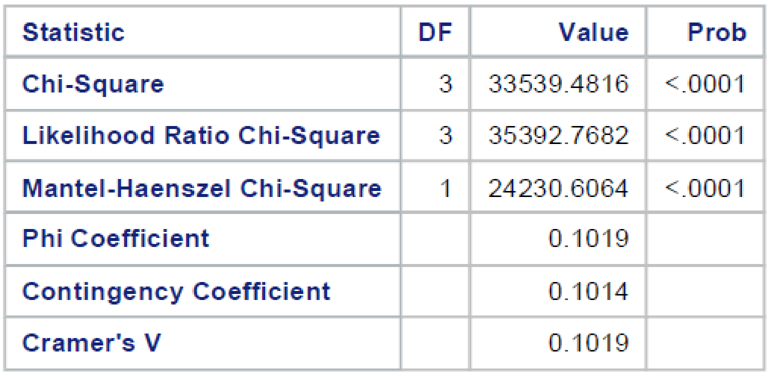
Statistic	DF	Value	Prob
Chi-Square	3	33146.4486	<.0001
Likelihood Ratio Chi-Square	3	32960.8029	<.0001
Mantel-Haenszel Chi-Square	1	20789.7089	<.0001
Phi Coefficient		0.0969	
Contingency Coefficient		0.0964	
Cramer's V		0.0969	

Figure A.19: Half marathon finish time Chi-square test



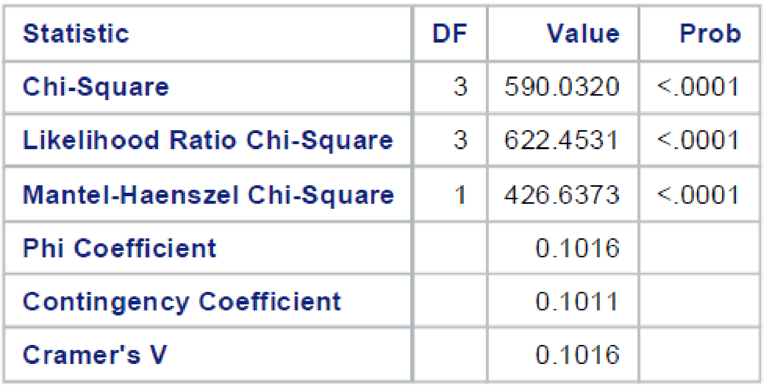
Statistic	DF	Value	Prob
Chi-Square	3	1550.8901	<.0001
Likelihood Ratio Chi-Square	3	1541.9850	<.0001
Mantel-Haenszel Chi-Square	1	972.9402	<.0001
Phi Coefficient		0.0966	
Contingency Coefficient		0.0962	
Cramer's V		0.0966	

Figure A.20: Half marathon race pace Chi-square test



Statistic	DF	Value	Prob
Chi-Square	3	33539.4816	<.0001
Likelihood Ratio Chi-Square	3	35392.7682	<.0001
Mantel-Haenszel Chi-Square	1	24230.6064	<.0001
Phi Coefficient		0.1019	
Contingency Coefficient		0.1014	
Cramer's V		0.1019	

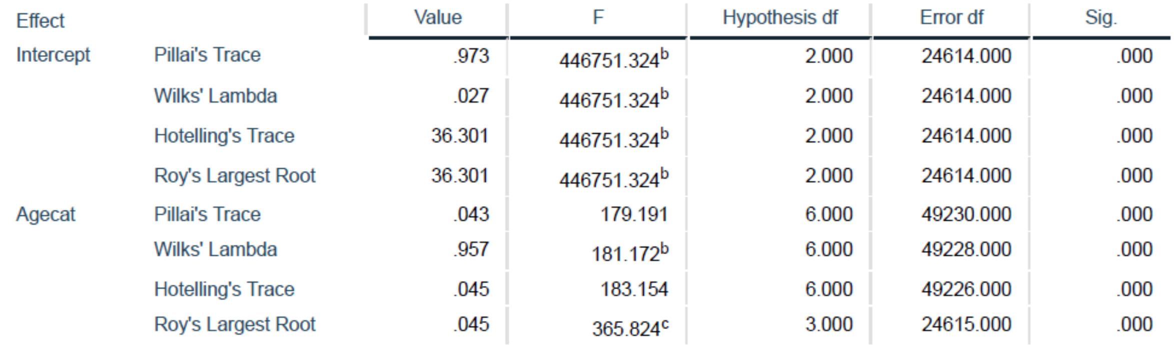
Figure A.21: Ultra marathon finish time Chi-square test



Statistic	DF	Value	Prob
Chi-Square	3	590.0320	<.0001
Likelihood Ratio Chi-Square	3	622.4531	<.0001
Mantel-Haenszel Chi-Square	1	426.6373	<.0001
Phi Coefficient		0.1016	
Contingency Coefficient		0.1011	
Cramer's V		0.1016	

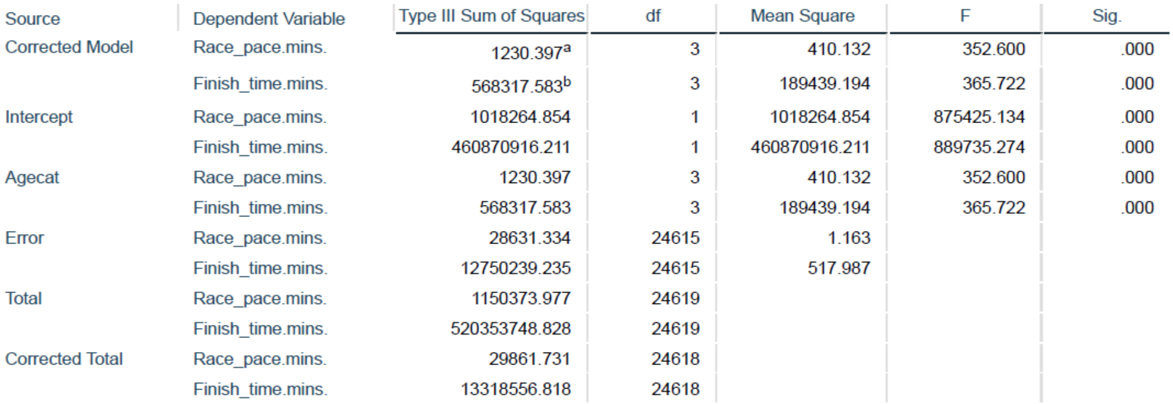
Figure A.22: Ultra marathon race pace Chi-square test

A.6 MANOVA



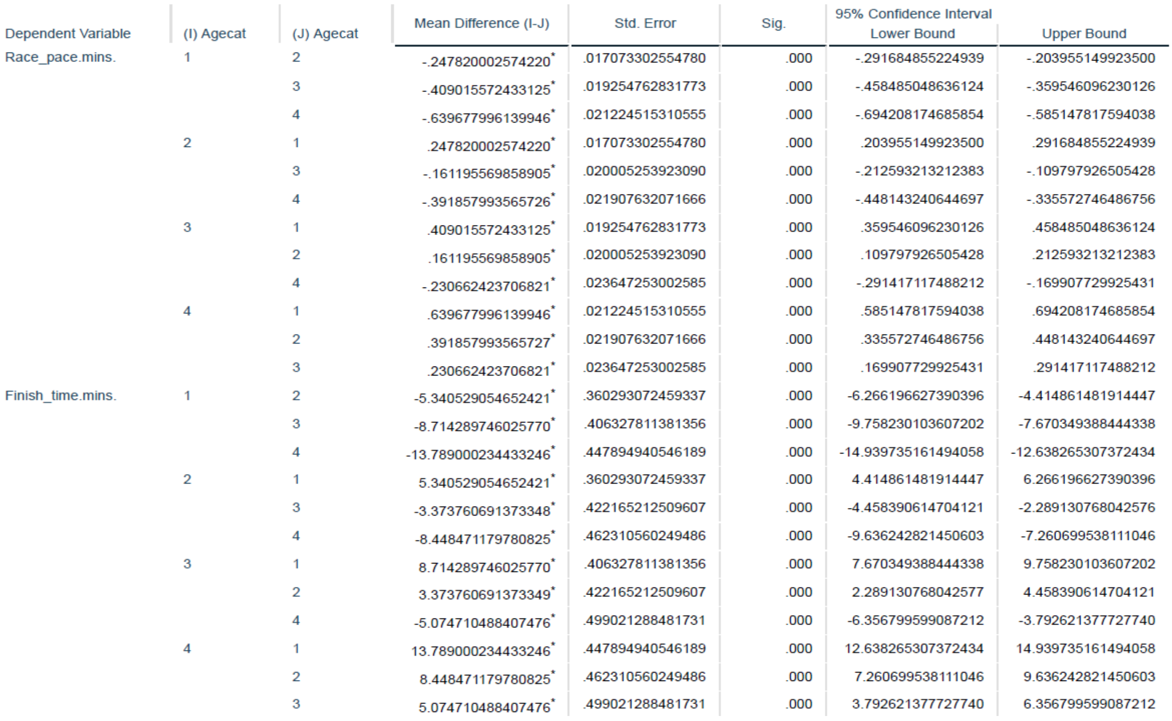
Effect		Value	F	Hypothesis df	Error df	Sig.
Intercept	Pillai's Trace	.973	446751.324[b]	2.000	24614.000	.000
	Wilks' Lambda	.027	446751.324[b]	2.000	24614.000	.000
	Hotelling's Trace	36.301	446751.324[b]	2.000	24614.000	.000
	Roy's Largest Root	36.301	446751.324[b]	2.000	24614.000	.000
Agecat	Pillai's Trace	.043	179.191	6.000	49230.000	.000
	Wilks' Lambda	.957	181.172[b]	6.000	49228.000	.000
	Hotelling's Trace	.045	183.154	6.000	49226.000	.000
	Roy's Largest Root	.045	365.824[c]	3.000	24615.000	.000

Figure A.23: One-way MANOVA



Source	Dependent Variable	Type III Sum of Squares	df	Mean Square	F	Sig.
Corrected Model	Race_pace.mins.	1230.397[a]	3	410.132	352.600	.000
	Finish_time.mins.	568317.583[b]	3	189439.194	365.722	.000
Intercept	Race_pace.mins.	1018264.854	1	1018264.854	875425.134	.000
	Finish_time.mins.	460870916.211	1	460870916.211	889735.274	.000
Agecat	Race_pace.mins.	1230.397	3	410.132	352.600	.000
	Finish_time.mins.	568317.583	3	189439.194	365.722	.000
Error	Race_pace.mins.	28631.334	24615	1.163		
	Finish_time.mins.	12750239.235	24615	517.987		
Total	Race_pace.mins.	1150373.977	24619			
	Finish_time.mins.	520353748.828	24619			
Corrected Total	Race_pace.mins.	29861.731	24618			
	Finish_time.mins.	13318556.818	24618			

Figure A.24: Test of between-subjects effects



Dependent Variable	(I) Agecat	(J) Agecat	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval Lower Bound	Upper Bound
Race_pace.mins.	1	2	-.247820002574220[*]	.017073302554780	.000	-.291684855224939	-.203955149923500
		3	-.409015572433125[*]	.019254762831773	.000	-.458485048636124	-.359546096230126
		4	-.639677996139946[*]	.021224515310555	.000	-.694208174685854	-.585147817594038
	2	1	.247820002574220[*]	.017073302554780	.000	.203955149923500	.291684855224939
		3	-.161195569858905[*]	.020005253923090	.000	-.212593213212383	-.109797926505428
		4	-.391857993565726[*]	.021907632071666	.000	-.448143240644697	-.335572746486756
	3	1	.409015572433125[*]	.019254762831773	.000	.359546096230126	.458485048636124
		2	.161195569858905[*]	.020005253923090	.000	.109797926505428	.212593213212383
		4	-.230662423706821[*]	.023647253002585	.000	-.291417117488212	-.169907729925431
	4	1	.639677996139946[*]	.021224515310555	.000	.585147817594038	.694208174685854
		2	.391857993565727[*]	.021907632071666	.000	.335572746486756	.448143240644697
		3	.230662423706821[*]	.023647253002585	.000	.169907729925431	.291417117488212
Finish_time.mins.	1	2	-5.340529054652421[*]	.360293072459337	.000	-6.266196627390396	-4.414861481914447
		3	-8.714289746025770[*]	.406327811381356	.000	-9.758230103607202	-7.670349388444338
		4	-13.789000234433246[*]	.447894940546189	.000	-14.939735161494058	-12.638265307372434
	2	1	5.340529054652421[*]	.360293072459337	.000	4.414861481914447	6.266196627390396
		3	-3.373760691373348[*]	.422165212509607	.000	-4.458390614704121	-2.289130768042576
		4	-8.448471179780825[*]	.462310560249486	.000	-9.636242821450603	-7.260699538111046
	3	1	8.714289746025770[*]	.406327811381356	.000	7.670349388444338	9.758230103607202
		2	3.373760691373349[*]	.422165212509607	.000	2.289130768042577	4.458390614704121
		4	-5.074710488407476[*]	.499021288481731	.000	-6.356799599087212	-3.792621377727740
	4	1	13.789000234433246[*]	.447894940546189	.000	12.638265307372434	14.939735161494058
		2	8.448471179780825[*]	.462310560249486	.000	7.260699538111046	9.636242821450603
		3	5.074710488407476[*]	.499021288481731	.000	3.792621377727740	6.356799599087212

Figure A.25: Multiple-Comparisons Post-Hoc(Tukey) tests

APPENDIX B:
R CODE

B.1 DATA CLEANING

```
#data with missing values#

data_raw<-read.csv('clipboard',sep="\t",header=T)

#Random forest imputation#



data_raw[,14]<-as.numeric(data_raw[,14])

data_raw[,5]<-as.numeric(data_raw[,5])

data_raw[,6]<-as.numeric(data_raw[,6])

data_raw[,7]<-as.numeric(data_raw[,7])

data_raw[,8]<-as.numeric(data_raw[,8])

data_raw[,9]<-as.numeric(data_raw[,9])

data_raw[,10]<-as.numeric(data_raw[,10])

data_raw[,11]<-as.numeric(data_raw[,11])

data_raw[,12]<-as.numeric(data_raw[,12])

data_raw[,13]<-as.numeric(data_raw[,13])




data_raw[,2]<-as.factor(data_raw[,2])

data_raw[,3]<-as.factor(data_raw[,3])

data_raw[,4]<-as.factor(data_raw[,4])
```

```
#this imputation part is only done when the analysis is redone for imputed data on the second round#

#random forest imputation#

require("missForest")

mrc.imp<-missForest(data_raw)

complete.mrc_data_forest <-mrc.imp$ximp

#taking maximum of finish time for duplicate entries based on runners code#

#finish time#

nodupkey<-function(dat){

  r<-1

  finish_time<-matrix(0,ncol=1,nrow=nrow(dat))

  dat[,15]<-finish_time

  t<-nrow(dat)

  while (r<=t){

    i<-dat[r,1]

    j<-dat[dat[,1]==i,]

    k<-nrow(j)

    dat[r,15]<-max(j[,5])

    r<-r+k

  }

  dat<-dat[dat[,15]!=0,]

  dat<-data.frame(dat)

  write_xlsx(dat,"H:/MRC DATA/noduplicates_forest.xlsx")

}

nodupkey(complete.mrc_data_forest)
```

```
#taking maximum of duplicate race pace entries based on runners code#
nodupkey<-function(dat){
  r<-1
  Race_pace<-matrix(0,ncol=1,nrow=nrow(dat))
  dat[,16]<-Race_pace
  t<-nrow(dat)
  while (r<=t){
    i<-dat[r,1]
    j<-dat[dat[,1]==i,]
    k<-nrow(j)
    dat[r,16]<-max(j[,14])
    r<-r+k
  }
  dat<-dat[dat[,16]!=0,]
  dat<-data.frame(dat)
  write_xlsx(dat,"H:/MRC DATA/noduplicates_forest.xlsx")
}
nodupkey(complete.mrc_data_forest)


#mice imputation#
imputed_Data <- mice(data_raw, m=5, maxit = 50, method = 'cart', seed = 500)
complete.mrc_data_mice <- complete(imputed_Data,1)
```

```
#taking maximum of duplicate entries for imputed data#

#finish time#


nodupkey<-function(dat){

  r<-1

  finish_time<-matrix(0,ncol=1,nrow=nrow(dat))

  dat[,15]<-finish_time

  t<-nrow(dat)

  while (r<=t){

    i<-dat[r,1]

    j<-dat[dat[,1]==i,]

    k<-nrow(j)

    dat[r,15]<-max(j[,5])

    r<-r+k

  }

  dat<-dat[dat[,15]!=0,]

  dat<-data.frame(dat)

  write_xlsx(dat,"H:/MRC DATA/noduplicates_mice.xlsx")

}

nodupkey(complete.mrc_data_mice)


# taking the maximum of duplicate race pace entries#
```

```
nodupkey<-function(dat){
  r<-1
  Race_pace<-matrix(0,ncol=1,nrow=nrow(dat))
  dat[,16]<-Race_pace
  t<-nrow(dat)
  while (r<=t){
    i<-dat[r,1]
    j<-dat[dat[,1]==i,]
    k<-nrow(j)
    dat[r,16]<-max(j[,14])
    r<-r+k
  }
  dat<-dat[dat[,16]!=0,]
  dat<-data.frame(dat)
  write_xlsx(dat,"H:/MRC DATA/noduplicates_mice.xlsx")
}
nodupkey(complete.mrc_data_mice)
#complete cases data#
mrc.complete<-read.csv("clipboard",header=T,sep = "\t")
#complete cases#
#taking maximum of duplicate values for complete cases data#
#finish time#
```

```
nodupkey<-function(dat){

  r<-1

  finish_time<-matrix(0,ncol=1,nrow=nrow(dat))

  dat[,15]<-finish_time

  t<-nrow(dat)

  while (r<=t){

    i<-dat[r,1]

    j<-dat[dat[,1]==i,]

    k<-nrow(j)

    dat[r,15]<-max(j[,5])

    r<-r+k

  }

  dat<-dat[dat[,15]!=0,]

  dat<-data.frame(dat)

  write_xlsx(dat,"H:/MRC DATA/noduplicates.xlsx")

}

nodupkey(mrc.complete)
```

```
#Race pace#
#taking maximum of duplicate values for complete cases data#
nodupkey<-function(dat){
  r<-1
  Race_pace<-matrix(0,ncol=1,nrow=nrow(dat))
  dat[,16]<-Race_pace
  t<-nrow(dat)
  while (r<=t){
    i<-dat[r,1]
    j<-dat[dat[,1]==i,]
    k<-nrow(j)
    dat[r,16]<-max(j[,14])
    r<-r+k
  }
  dat<-dat[dat[,16]!=0,]
  dat<-data.frame(dat)
  write_xlsx(dat,"H:/MRC DATA/noduplicates_1.xlsx")
}
nodupkey(mrc.complete)
```

```
#importing and preparing data#
Mrc.nodup<-read.csv("clipboard",header=T,sep = "\t")
DAT<-mrc.nodup
DAT<-na.omit(DAT)


DAT[,5]<-as.numeric(DAT[,5])
DAT[,4]<-as.numeric(DAT[,4])
DAT[,3]<-as.numeric(DAT[,3])
DAT[,2]<-as.numeric(DAT[,2])
DAT[,1]<-as.numeric(DAT[,1])
DAT[,10]<-as.numeric(DAT[,10])
DAT[,9]<-as.numeric(DAT[,9])
DAT[,11]<-as.numeric(DAT[,11])
DAT[,12]<-as.numeric(DAT[,12])
DAT[,13]<-as.numeric(DAT[,13])


DAT[,6]<-as.factor(DAT[,6])
DAT[,7]<-as.factor(DAT[,7])
DAT[,8]<-as.factor(DAT[,8])
mrc<-DAT


#splitting data in the race types#
half<-mrc[mrc["Racetype"]==2,]
ultra<-mrc[mrc["Racetype"]==1,]
```

```
#removing outliers#

Q <- quantile(half$Finish_time.mins., probs=c(.25, .75), na.rm = FALSE)

iqr <- IQR(half$Finish_time.mins.)

Q1 <- quantile(ultra$Finish_time.mins., probs=c(.25, .75), na.rm = FALSE)

Iqr <- IQR(ultra$Finish_time.mins.)

q<-quantile(half$Race_pace.mins., probs=c(.25, .75), na.rm = FALSE)

iqr1<-IQR(half$Race_pace.mins.)

q1<-quantile(ultra$Race_pace.mins., probs=c(.25, .75), na.rm = FALSE)

Iqr1<-IQR(ultra$Race_pace.mins.)

half<- subset(half, half$Finish_time.mins. > (Q[1] - 1.5*iqr) & half$Finish_time.mins. < (Q[2]+1.5*iqr))

ultra<- subset(ultra, ultra$Finish_time.mins. > (Q1[1] - 1.5*Iqr) & ultra$Finish_time.mins. < (Q1[2]+1.5*Iqr))

half<- subset(half, half$Race_pace.mins. > (q[1] - 1.5*iqr1) & half$Race_pace.mins. < (q[2]+1.5*iqr1))

ultra<- subset(ultra, ultra$Race_pace.mins. > (q1[1] - 1.5*Iqr1) & ultra$Race_pace.mins. < (q1[2]+1.5*Iqr1))

#correlation plot#

library(corrplot)

cor1<-mrc[,-6:-8]

cor1<-data.matrix(cor1)

corrplot(cor1,method = "number")
```

B.2 EXPLORATORY DATA ANALYSIS

```
#exploratory data analysis#

#scatterplots#

#finish time#

library(ggplot2)

library(ggpubr)

p1<-ggplot(half,aes(x=Times_train.race.km.week.,y=Finish_time.mins.,colour=
Gender))+geom_point()+geom_smooth(method = lm,se=F) + labs(title="(a)")

p2<-ggplot(half,aes(x=Times_train.race.km.week.,y=Finish_time.mins.,colour =
Agecat))+geom_point()+geom_smooth(method = lm,se=F) + labs(title="(b)")

p3<-ggplot(ultra,aes(x=Times_train.race.km.week.,y=Finish_time.mins.,colour=
Gender))+geom_point()+geom_smooth(method = lm,se=F) + labs(title="(c)")

p4<-ggplot(ultra,aes(x=Times_train.race.km.week.,y=Finish_time.mins.,colour =
Agecat))+geom_point()+geom_smooth(method = lm,se=F) + labs(title="(d)")

ggarrange(p1, p2,p3, p4,ncol = 2, nrow = 2)
```

```
p5<-ggplot(half,aes(x=Training_distance.km.week.,y=Finish_time.mins.,colour=
Gender))+geom_point()+geom_smooth(method = lm,se=F)+labs(title="(a)")

p6<-ggplot(half,aes(x=Training_distance.km.week.,y=Finish_time.mins.,colour =
Agecat))+geom_point()+geom_smooth(method = lm,se=F) + labs(title="(b)")

p7<-ggplot(ultra,aes(x=Training_distance.km.week.,y=Finish_time.mins.,colour=
Gender))+geom_point()+geom_smooth(method = lm,se=F) + labs(title="(c)")

p8<-ggplot(ultra,aes(x=Training_distance.km.week.,y=Finish_time.mins.,colour =
Agecat))+geom_point()+geom_smooth(method = lm,se=F) + labs(title="(d)")

ggarrange(p5,p6,p7,p8,ncol = 2, nrow = 2)

p9<-ggplot(half,aes(x=Tar.brick_._time.v1_9.,y=Finish_time.mins.,colour=
Gender))+geom_point()+geom_smooth(method = lm,se=F) + labs(title="(a)")

p10<-ggplot(half,aes(x=Tar.brick_._time.v1_9.,y=Finish_time.mins.,colour =
Agecat))+geom_point()+geom_smooth(method = lm,se=F) + labs(title="(b)")

p11<-ggplot(ultra,aes(x=Tar.brick_._time.v1_9.,y=Finish_time.mins.,colour=
Gender))+geom_point()+geom_smooth(method = lm,se=F) + labs(title="(c)")

p12<-ggplot(ultra,aes(x=Tar.brick_._time.v1_9.,y=Finish_time.mins.,colour =
Agecat))+geom_point()+geom_smooth(method = lm,se=F) + labs(title="(d)")

ggarrange(p9,p10,p11,p12, ncol = 2, nrow = 2)
```

```
p13<-ggplot(half,aes(x=Training_pace.mins.,y=Finish_time.mins.,colour=
Gender))+geom_point()+geom_smooth(method = lm,se=F) + labs(title="(a)")

p14<-ggplot(half,aes(x=Training_pace.mins.,y=Finish_time.mins.,colour =
Agecat))+geom_point()+geom_smooth(method = lm,se=F) + labs(title="(b)")

p15<-ggplot(ultra,aes(x=Training_pace.mins.,y=Finish_time.mins.,colour=
Gender))+geom_point()+geom_smooth(method = lm,se=F) + labs(title="(c)")

p16<-ggplot(ultra,aes(x=Training_pace.mins.,y=Finish_time.mins.,colour =
Agecat))+geom_point()+geom_smooth(method = lm,se=F) + labs(title="(d)")

ggarrange(p13,p14,p15,p16,ncol = 2, nrow = 2)



p133<-ggplot(half,aes(x=Recreational_runner.years.,y=Finish_time.mins.,colour=
Gender))+geom_point()+geom_smooth(method = lm,se=F) + labs(title="(a)")

p144<-ggplot(half,aes(x=Recreational_runner.years.,y=Finish_time.mins.,colour =
Agecat))+geom_point()+geom_smooth(method = lm,se=F) + labs(title="(b)")

p155<-ggplot(ultra,aes(x=Recreational_runner.years.,y=Finish_time.mins.,colour=
Gender))+geom_point()+geom_smooth(method = lm,se=F) + labs(title="(c)")

p166<-ggplot(ultra,aes(x=Recreational_runner.years.,y=Finish_time.mins.,colour =
Agecat))+geom_point()+geom_smooth(method = lm,se=F) + labs(title="(d)")

ggarrange(p133,p144,p155,p166,ncol = 2, nrow = 2)

p1333<-ggplot(half,aes(x=Distance_runner.years.,y=Finish_time.mins.,colour=
Gender))+geom_point()+geom_smooth(method = lm,se=F) + labs(title="(a)")

p1444<-ggplot(half,aes(x=Distance_runner.years.,y=Finish_time.mins.,colour =
Agecat))+geom_point()+geom_smooth(method = lm,se=F) + labs(title="(b)")

p1555<-ggplot(ultra,aes(x=Distance_runner.years.,y=Finish_time.mins.,colour=
Gender))+geom_point()+geom_smooth(method = lm,se=F) + labs(title="(c)")

p1666<-ggplot(ultra,aes(x=Distance_runner.years.,y=Finish_time.mins.,colour =
Agecat))+geom_point()+geom_smooth(method = lm,se=F) + labs(title="(d)")

ggarrange(p1333,p1444,p1555,p1666,ncol = 2, nrow = 2)
```

```
#race pace#

p17<-ggplot(mrc,aes(x=Times_train.race.km.week.,y=Race_pace.mins.,colour=
Gender))+geom_point()+geom_smooth(method = lm,se=F) + labs(title="(a)")

p18<-ggplot(mrc,aes(x=Times_train.race.km.week.,y=Race_pace.mins.,colour =
Agecat))+geom_point()+geom_smooth(method = lm,se=F) + labs(title="(b)")

p19<-ggplot(mrc,aes(x=Training_distance.km.week.,y=Race_pace.mins.,colour=
Gender))+geom_point()+geom_smooth(method = lm,se=F) + labs(title="(c)")

p20<-ggplot(mrc,aes(x=Training_distance.km.week.,y=Race_pace.mins.,colour =
Agecat))+geom_point()+geom_smooth(method = lm,se=F) + labs(title="(d)")

ggarrange(p17,p18,p19,p20,ncol = 2, nrow = 2)

p177<-ggplot(mrc,aes(x=Recreational_runner.years.,y=Race_pace.mins.,colour=
Gender))+geom_point()+geom_smooth(method = lm,se=F) + labs(title="(a)")

p188<-ggplot(mrc,aes(x=Recreational_runner.years.,y=Race_pace.mins.,colour =
Agecat))+geom_point()+geom_smooth(method = lm,se=F) + labs(title="(b)")

p199<-ggplot(mrc,aes(x=Distance_runner.years.,y=Race_pace.mins.,colour=
Gender))+geom_point()+geom_smooth(method = lm,se=F) + labs(title="(c)")

p200<-ggplot(mrc,aes(x=Distance_runner.years.,y=Race_pace.mins.,colour =
Agecat))+geom_point()+geom_smooth(method = lm,se=F) + labs(title="(d)")

ggarrange(p177,p188,p199,p200,ncol = 2, nrow = 2)
```

```
p21<-ggplot(mrc,aes(x=Tar.brick_._time.v1_9.,y=Race_pace.mins.,colour=
Gender))+geom_point()+geom_smooth(method = lm,se=F) + labs(title="(a)")

p22<-ggplot(mrc,aes(x=Tar.brick_._time.v1_9.,y=Race_pace.mins.,colour =
Agecat))+geom_point()+geom_smooth(method = lm,se=F) + labs(title="(b)")

p23<-ggplot(mrc,aes(x=Training_pace.mins.,y=Race_pace.mins.,colour=
Gender))+geom_point()+geom_smooth(method = lm,se=F) + labs(title="(c)")

p24<-ggplot(mrc,aes(x=Training_pace.mins.,y=Race_pace.mins.,colour =
Agecat))+geom_point()+geom_smooth(method = lm,se=F) + labs(title="(d)")

ggarrange(p21,p22,p23,p24,ncol = 2, nrow = 2)
```

```
#violin plots#

#finish time#

#half#

tr<-ggplot(half, aes(x=Times_train.race.km.week., y=Finish_time.mins.)) +

  geom_boxplot(width = .2) + labs(title="(a)")

dist<-ggplot(half,aes(x=Training_distance.km.week.,y=Finish_time.mins.)) +

  geom_boxplot(width = .2)  + labs(title="(b)")

Tar<-ggplot(half, aes(x=Tar.brick_._time.v1_9., y=Finish_time.mins.)) +

  geom_boxplot(width = .2)  + labs(title="(c)")

pace<-ggplot(half, aes(x=Training_pace.mins., y=Finish_time.mins.)) +

  geom_boxplot(width = .2)  + labs(title="(d)")

ggarrange(tr, dist,Tar, pace,ncol = 2, nrow = 2)

#ultra #

p26<-ggplot(ultra,aes(x=Times_train.race.km.week.,y=Finish_time.mins.)) +

  geom_boxplot(width = .2) + labs(title="(a)")

p27<-ggplot(ultra,aes(x=Training_distance.km.week.,y=Finish_time.mins.)) +

  geom_boxplot(width = .2)  + labs(title="(b)")

p28<-ggplot(ultra, aes(x=Tar.brick_._time.v1_9., y=Finish_time.mins.)) +

  geom_boxplot(width = .2)  + labs(title="(c)")

p29<-ggplot(ultra, aes(x=Training_pace.mins., y=Finish_time.mins.)) +

  geom_boxplot(width = .2)  + labs(title="(d)")

ggarrange(p26,p27,p28,p29, ncol = 2, nrow = 2)
```

```
p30<-ggplot(ultra, aes(x=Times_train.race.km.week., y=Race_pace.mins.))
+geom_boxplot(width = .2) + labs(title="(a)")

p31<-ggplot(ultra,aes(x=Training_distance.km.week., y=Race_pace.mins.)) +
geom_boxplot(width = .2)  + labs(title="(b)")

p32<-ggplot(ultra, aes(x=Tar.brick_._time.v1_9., y=Race_pace.mins.)) +

 geom_boxplot(width = .2) + labs(title="(c)")

p33<-ggplot(ultra, aes(x=Training_pace.mins., y=Race_pace.mins.)) +

  geom_boxplot(width = .2)  + labs(title="(d)")

ggarrange(p30,p31,p32,p33, ncol = 2, nrow = 2)

p34<-ggplot(half, aes(x=Agecat, y=Finish_time.mins.,fill=Agecat)) + geom_boxplot(width = .2) +
labs(title="(a)",x= "Age category")

p35<-ggplot(ultra, aes(x=Agecat, y=Finish_time.mins.,fill=Agecat)) + geom_boxplot(width = .2)
+ labs(title="(b)",x= "Age category")

p36<-ggplot(half, aes(x=Agecat, y=Race_pace.mins.,fill=Agecat)) + geom_boxplot(width = .2) +
labs(title="(c)", x="Age category")

p37<-ggplot(ultra, aes(x=Agecat, y=Race_pace.mins.,fill=Agecat))+ geom_boxplot(width = .2) +
labs(title="(d)", x="Age category")



ggarrange(p34,p35,p36,p37, ncol = 2, nrow = 2)
```

B.3 MODEL-FITTING

```
Model-fitting

library(caret)

library(dplyr)

library(tidyr)

library(glmnet)

library(randomForest)

#Finish time#

#LASSO#

#half#

#taking out some variables#

half_finish<-half[,-12]

half_finish<-half_finish[,-7]

#splitting the data#

set.seed(2021)

Ind_half_finish<-createDataPartition(half_finish[,11],p=0.8,list = F)

train_half_finish<-half_finish[Ind_half_finish,]

test_half_finish<-half_finish[-Ind_half_finish,]



#LASSO MODEL BUILDING#

Cv_model <- cv.glmnet( data.matrix(train_half_finish[ ,-11]),  data.matrix(train_half_finish[,11]),
alpha = 1)

Best_lambda <- Cv_model$lambda.min

Best_model <- glmnet(data.matrix(train_half_finish[ ,-11]),  data.matrix(train_half_finish[,11]),
alpha = 1, lambda = Best_lambda)
```

```
# Variable importance#

L1<-vip(Best_model) + ggtitle("(a)")

#coefficients#

c1<-coef(Best_model)

#predictions#

predictions_half_finish_LASSO_test<-Best_model%>%predict(data.matrix(test_half_finish[,-11]))

predictions_half_finish_LASSO_train<-Best_model%>%predict(data.matrix(train_half_finish[,-11]))

#performance#

#RMSE#

D1<-data.frame(Type="Half-Finish-time",Model="LASSO", "Test RMSE"=
RMSE(predictions_half_finish_LASSO_test,test_half_finish[,11]),"Train RMSE"=
RMSE(predictions_half_finish_LASSO_train,train_half_finish[,11]))

#MAE#

d1<-data.frame(Type="Half-Finish-time",Model="LASSO", "Test MAE"=
MAE(predictions_half_finish_LASSO_test,test_half_finish[,11]),"Train MAE"=
MAE(predictions_half_finish_LASSO_train,train_half_finish[,11]))

#elastic net model building#

set.seed(2021)

#model building#

Cont <- trainControl(method = "repeatedcv",

                     number = 10,

                     repeats = 5,

                     search = "random",

                     verboseIter = TRUE)
```

```
# Train the model
Elastic <- train(Finish_time.mins. ~ .,
               data = train_half_finish,
               method = "glmnet",
               preProcess = c("center", "scale"),
               tuneLength = 10,
               trControl = Cont)
E1<-vip(Elastic) + ggtitle("a")
#coefficients#
e1<-coef(Elastic$finalModel, Elastic$bestTune$lambda)
plot(varImp(e1))
#predictions#
predictions_half_finish_elastic_test<-Elastic%>%predict(test_half_finish[,-11])
predictions_half_finish_elastic_train<-Elastic%>%predict(train_half_finish[,-11])
#performance#
#RMSE#
D2<-data.frame(Type="Half-Finish-time",Model="Elastic-Net", "Test RMSE"=
RMSE(predictions_half_finish_elastic_test,test_half_finish[,11])
             ,"Train RMSE"= RMSE(predictions_half_finish_elastic_train,train_half_finish[,11]))


#MAE#
d2<-data.frame(Type="Half-Finish-time",Model="Elastic-Net", "Test MAE"=
MAE(predictions_half_finish_elastic_test,test_half_finish[,11])
             ,"Train MAE"= MAE(predictions_half_finish_elastic_train,train_half_finish[,11]))
```

```
#Random forest model building#

set.seed(2021)

half.time.rf <- randomForest(Finish_time.mins. ~ ., data = train_half_finish)

half.time.rf.pred_test <- half.time.rf %>% predict(test_half_finish)

half.time.rf.pred_train <- half.time.rf %>% predict(train_half_finish)

#performance#

#RMSE#

D3 <-data.frame(Type="Half-Finish-time",Model="Random-Forest", "Test RMSE"=
RMSE(half.time.rf.pred_test,test_half_finish[,11]),

            "Train RMSE"=RMSE(half.time.rf.pred_train,train_half_finish[,11]))

#MAE#

d3 <-data.frame(Type="Half-Finish-time",Model="Random-Forest", "Test MAE"=
MAE(half.time.rf.pred_test,test_half_finish[,11]),

            "Train MAE"=MAE(half.time.rf.pred_train,train_half_finish[,11]))



#variable importance plot#

par(mfrow=c(2,2))

varImpPlot(half.time.rf,main = "(a)")
```

```
#ultra#
#LASSO#
#Finish time#
#taking variables out#
ultra_finish<-ultra[,-12]
ultra_finish<-ultra_finish[,-7]
#splitting the data#
set.seed(2021)
Ind_ultra_finish<-createDataPartition(ultra_finish[,11],p=0.9,list = F)
train_ultra_finish<-ultra_finish[Ind_ultra_finish,]
test_ultra_finish<-ultra_finish[-Ind_ultra_finish,]

#LASSO model building#
cv_model <- cv.glmnet( data.matrix(train_ultra_finish[ ,-11]),  data.matrix(train_ultra_finish[,11]),
alpha = 1)
best_lambda <- cv_model$lambda.min
best_model <- glmnet(data.matrix(train_ultra_finish[ ,-11]),  data.matrix(train_ultra_finish[,11]),
alpha = 1, lambda = best_lambda)
L2<-vip(best_model) + ggtitle("(b)")
#coefficients#
c2<-coef(best_model)
predictions_ultra_finish_LASSO_test<-best_model%>%predict(data.matrix(test_ultra_finish[,-
11]))
predictions_ultra_finish_LASSO_train<-best_model%>%predict(data.matrix(train_ultra_finish[,-
11]))
```

```
#performance#

#RMSE#

D4<-data.frame(Type="Ultra-Finish-time",Model="LASSO", "Test RMSE"=
RMSE(predictions_ultra_finish_LASSO_test,test_ultra_finish[,11]),"Train RMSE"=
RMSE(predictions_ultra_finish_LASSO_train,train_ultra_finish[,11]))



#MAE#

d4<-data.frame(Type="Ultra-Finish-time",Model="LASSO", "Test MAE"=
MAE(predictions_ultra_finish_LASSO_test,test_ultra_finish[,11]),"Train MAE"=
MAE(predictions_ultra_finish_LASSO_train,train_ultra_finish[,11]))

#elastic net building#

set.seed(2021)

#model building#

cont <- trainControl(method = "repeatedcv",

                     number = 10,

                     repeats = 5,

                     search = "random",

                     verboseIter = TRUE)

# Train the model

elastic <- train(Finish_time.mins. ~ .,

                 data = train_ultra_finish,

                 method = "glmnet",

                 preProcess = c("center", "scale"),

                 tuneLength = 10,

                 trControl = cont)

E2<-vip(elastic) + ggtitle("b")
```

```
#coefficients#

e2<-coef(elastic$finalModel, elastic$bestTune$lambda)



#predictions#

predictions_ultra_finish_elastic_test<-elastic%>%predict(test_ultra_finish[,-11])

predictions_ultra_finish_elastic_train<-elastic%>%predict(train_ultra_finish[,-11])



#performance#

#RMSE#

D5<-data.frame(Type="Ultra-Finish-time",Model="Elastic-Net", "Test RMSE"=
RMSE(predictions_ultra_finish_elastic_test,test_ultra_finish[,11]),"Train RMSE"=
RMSE(predictions_ultra_finish_elastic_train,train_ultra_finish[,11]))



#MAE#

d5<-data.frame(Type="Ultra-Finish-time",Model="Elastic-Net", "Test MAE"=
MAE(predictions_ultra_finish_elastic_test,test_ultra_finish[,11])

,"Train MAE"= MAE(predictions_ultra_finish_elastic_train,train_ultra_finish[,11]))
```

```
#Random forest building#
set.seed(2021)
ultra.time.rf <- randomForest(Finish_time.mins. ~ ., data = train_ultra_finish)
ultra.time.rf.pred_test <- ultra.time.rf %>% predict(test_ultra_finish)
ultra.time.rf.pred_train <- ultra.time.rf %>% predict(train_ultra_finish)


#performance#
#RMSE#
D6 <-data.frame(Type="Ultra-Finish-t ime",Model="Random-Forest", "Test RMSE"=
RMSE(ultra.time.rf.pred_test,test_ultra_finish[,11]),"Train RMSE"=
RMSE(ultra.time.rf.pred_train,train_ultra_finish[,11]))
#MAE#
d6 <-data.frame(Type="Ultra-Finish-time",Model="Random-Forest", "Test MAE"=
MAE(ultra.time.rf.pred_test,test_ultra_finish[,11]),"Train MAE"=
MAE(ultra.time.rf.pred_train,train_ultra_finish[,11]))


#variable importance plot#
varImpPlot(ultra.time.rf,main = "(b)")


#Race Pace#
#LASSO#
#half#
#taking variables out#
half_pace<-half[,-13]
half_pace<-half_pace[,-7]
```

```
#splitting the data#
set.seed(2021)
Ind_half_pace<-createDataPartition(half_pace[,11],p=0.9,list = F)


train_half_pace<-half_pace[Ind_half_pace,]
test_half_pace<-half_pace[-Ind_half_pace,]
#LASSO model building#


cv_Model <- cv.glmnet( data.matrix(train_half_pace[ ,-11]),  data.matrix(train_half_pace[,11]), alpha = 1)
best_Lambda <- cv_Model$lambda.min
best_Model <- glmnet(data.matrix(train_half_pace[ ,-11]),  data.matrix(train_half_pace[,11]), alpha = 1, lambda = best_Lambda)
L3<-vip(best_Model) + ggtitle("(c)")


##coefficients
c3<-coef(best_Model)


#predictions#
predictions_half_pace_LASSO_test<-best_Model%>%predict(data.matrix(test_half_pace[,-11]))
predictions_half_pace_LASSO_train<-best_Model%>%predict(data.matrix(train_half_pace[,-11]))
```

```
#performance#

D7<-data.frame(Type="Half-Race-pace",Model="LASSO", "Test RMSE"=
RMSE(predictions_half_pace_LASSO_test,test_half_pace[,11]),"Train RMSE"=
RMSE(predictions_half_pace_LASSO_train,train_half_pace[,11]))


d7<-data.frame(Type="Half-Race-pace",Model="LASSO", "Test MAE"=
MAE(predictions_half_pace_LASSO_test,test_half_pace[,11]),"Train MAE"=
MAE(predictions_half_pace_LASSO_train,train_half_pace[,11]))


#elastic net building#

set.seed(2021)

#model building#

cnt <- trainControl(method = "repeatedcv",

                     number = 10,

                     repeats = 5,

                     search = "random",

                     verboseIter = TRUE)

# Train the model

elastic_N <- train(Race_pace.mins. ~ .,

                    data = train_half_pace,

                    method = "glmnet",

                    preProcess = c("center", "scale"),

                    tuneLength = 10,

                    trControl = cnt)


E3<-vip(elastic_N) + ggtitle("c")
```

```
#coefficients#

e3<-coef(elastic_N$finalModel, elastic_N$bestTune$lambda)



#predictions#

predictions_half_pace_elastic_test<-elastic_N%>%predict(test_half_pace[,-11])

predictions_half_pace_elastic_train<-elastic_N%>%predict(train_half_pace[,-11])



#performance#

D8<-data.frame(Type="Half-Race-pace",Model="Elastic-Net", "Test RMSE"=
RMSE(predictions_half_pace_elastic_test,test_half_pace[,11]),"Train RMSE"=
RMSE(predictions_half_pace_elastic_train,train_half_pace[,11]))



d8<-data.frame(Type="Half-Race-pace",Model="Elastic-Net", "Test MAE"=
MAE(predictions_half_pace_elastic_test,test_half_pace[,11]),"Train MAE"=
MAE(predictions_half_pace_elastic_train,train_half_pace[,11]))

#Random forest#



set.seed(2021)

half.pace.rf <- randomForest(Race_pace.mins. ~ ., data = train_half_pace)

half.pace.rf.pred_test <- half.pace.rf %>% predict(test_half_pace)

half.pace.rf.pred_train <- half.pace.rf %>% predict(train_half_pace)
```

```
#performance#

#RMSE#

D9 <-data.frame(Type="Half-Race-pace",Model="Random-Forest", "Test RMSE"=
RMSE(half.pace.rf.pred_test,test_half_pace[,11]),"Train RMSE"=
RMSE(half.pace.rf.pred_train,train_half_pace[,11]))



#MAE#

d9 <-data.frame(Type="Half-Race-pace",Model="Random-Forest", "Test MAE"=
MAE(half.pace.rf.pred_test,test_half_pace[,11]),"Train MAE"=
MAE(half.pace.rf.pred_train,train_half_pace[,11]))



#Variable importance plot#

varImpPlot(half.pace.rf,main="(c)")



#ultra#

#LASSO#

#Race Pace#

#taking variables out#

ultra_pace<-ultra[,-13]

ultra_pace <-ultra_pace[,-7]



#splitting the data#

set.seed(2021)

Ind_ultra_pace<-createDataPartition(ultra_pace[,11],p=0.9,list = F)

train_ultra_pace<-ultra_pace[Ind_ultra_pace,]

test_ultra_pace<-ultra_pace[-Ind_ultra_pace,]
```

```
#LASSO model building#
cv_m <- cv.glmnet( data.matrix(train_ultra_pace[ ,-11]),  data.matrix(train_ultra_pace[,11]),
alpha = 1)
best_l <- cv_m$lambda.min
best_m <- glmnet(data.matrix(train_ultra_pace[ ,-11]),  data.matrix(train_ultra_pace[,11]), alpha
= 1, lambda = best_l)
L4<-vip(best_m) + ggtitle("(d)")
#coefficients#
c4<-coef(best_m)


#predictions#
predictions_ultra_pace_LASSO_test<-best_m%>%predict(data.matrix(test_ultra_pace[,-11]))
predictions_ultra_pace_LASSO_train<-best_m%>%predict(data.matrix(train_ultra_pace[,-11]))


#performance#
#RMSE#
D10<-data.frame(Type="Ultra-Race-pace",Model="LASSO", "Test RMSE"=
RMSE(predictions_ultra_pace_LASSO_test,test_ultra_pace[,11])
            ,"Train RMSE"= RMSE(predictions_ultra_pace_LASSO_train,train_ultra_pace[,11]))
#MAE#
d10<-data.frame(Type="Ultra-Race-pace",Model="LASSO", "Test MAE"=
MAE(predictions_ultra_pace_LASSO_test,test_ultra_pace[,11])
            ,"Train MAE"= MAE(predictions_ultra_pace_LASSO_train,train_ultra_pace[,11]))
```

```
#elastic net#

set.seed(2021)

#model building#

CONT <- trainControl(method = "repeatedcv",

                     number = 10,

                     repeats = 5,

                     search = "random",

                     verboseIter = TRUE)


# Train the model

ELASTIC <- train(Race_pace.mins. ~ .,

                 data = train_ultra_pace,

                 method = "glmnet",

                 preProcess = c("center", "scale"),

                 tuneLength = 10,

                 trControl = CONT)

E4<-vip(ELASTIC) + ggtitle("(d)")

#coefficients#

e4<-coef(ELASTIC$finalModel, ELASTIC$bestTune$lambda)


#predictions#

predictions_ultra_pace_elastic_test<-ELASTIC%>%predict(test_ultra_pace[,-11])

predictions_ultra_pace_elastic_train<-ELASTIC%>%predict(train_ultra_pace[,-11])
```

```
#performance#

#RMSE#

D11<-data.frame(Type="Ultra-Race-pace",Model="Elastic-Net","Test RMSE"=
RMSE(predictions_ultra_pace_elastic_test,test_ultra_pace[,11]),"Train RMSE"=
RMSE(predictions_ultra_pace_elastic_train,train_ultra_pace[,11]))

#MAE#

d11<-data.frame(Type="Ultra-Race-pace",Model="Elastic-Net", "Test MAE"=
MAE(predictions_ultra_pace_elastic_test,test_ultra_pace[,11]),"Train MAE"=
MAE(predictions_ultra_pace_elastic_train,train_ultra_pace[,11]))



#Random forest#

set.seed(2021)

ultra.pace.rf <- randomForest(Race_pace.mins. ~ ., data = train_ultra_pace)

ultra.pace.rf.pred_test <- ultra.pace.rf %>% predict(test_ultra_pace)

ultra.pace.rf.pred_train <- ultra.pace.rf %>% predict(train_ultra_pace)



#RMSE#

D12 <-data.frame(Type="Ultra Race pace",Model="Random Forest", "Test RMSE"=
RMSE(ultra.pace.rf.pred_test,test_ultra_pace[,11]),"Train RMSE"=
RMSE(ultra.pace.rf.pred_train,train_ultra_pace[,11]))



#MAE#

d12<-data.frame(Type="Ultra-Race-pace",Model="Random-Forest","Test MAE"=
MAE(ultra.pace.rf.pred_test,test_ultra_pace[,11]),"Train MAE"=
MAE(ultra.pace.rf.pred_train,train_ultra_pace[,11]))
```

```
#Variable importance plot#

varImpPlot(ultra.pace.rf,main="(d)")



#COMBINING#

RMSE.stats<-rbind(D1,D2,D3,D4,D5,D6,D7,D8,D9,D10,D11,D12)



RMSE.stats[,5]<-((RMSE.stats[,3]-RMSE.stats[,4])/RMSE.stats[,4])*100



RMSE.stats<-RMSE.stats %>% rename_at("V5",~"% increase")

RMSE.stats



MAE.stats<-rbind(d1,d2,d3,d4,d5,d6,d7,d8,d9,d10,d11,d12)



MAE.stats[,5]<-((MAE.stats[,3]-MAE.stats[,4])/MAE.stats[,4])*100



MAE.stats<-MAE.stats %>% rename_at("V5",~"% increase")

MAE.stats

ls<-cbind(c1,c2,c3,c4)

en<-cbind(e1,e2,e3,e4)

ls

en

ggarrange(L1, L2,L3,L4, ncol = 2, nrow = 2)

ggarrange(E1, E2,E3,E4, ncol = 2, nrow = 2)
```

B.4 PERFORMANCE METRICS

```
dat<-read.csv('clipboard',sep="\t",header=T)
metric<-function(dat){
s<-3
r<-1
while (s<=nrow(dat)) {

  if (s<=3){
    d<-dat[r:s,]
d[] <- lapply(d, as.character)
d<-d%>% pivot_longer(cols = c("Test.RMSE","Test.MAE"),names_to = "Metric",values_to = "Value")
p1<-ggplot(data = d, aes(Metric, Value, fill = Model)) +
    geom_bar(stat = "identity", position = "dodge") + labs(title="(a)")
}
r<-s+1
s<-s+3
   if( s>3 && s<=6){
    d<-dat[r:s,]
    d[] <- lapply(d, as.character)
    d<-d%>% pivot_longer(cols = c("Test.RMSE","Test.MAE"),names_to = "Metric",values_to = "Value")
    p2<-ggplot(data = d, aes(Metric, Value, fill = Model)) +
      geom_bar(stat = "identity", position = "dodge") + labs(title="(b)")
  }
r<-s+1
s<-s+3
if( s>6 && s<=9){
  d<-dat[r:s,]
  d[] <- lapply(d, as.character)
  d<-d%>% pivot_longer(cols = c("Test.RMSE","Test.MAE"),names_to = "Metric",values_to = "Value")
  p3<-ggplot(data = d, aes(Metric, Value, fill = Model)) +
    geom_bar(stat = "identity", position = "dodge") + labs(title="(c)")
}
r<-s+1
s<-s+3
if( s>9 && s<=12){
  d<-dat[r:s,]
  d[] <- lapply(d, as.character)
  d<-d%>% pivot_longer(cols = c("Test.RMSE","Test.MAE"),names_to = "Metric",values_to = "Value")
  p4<-ggplot(data = d, aes(Metric, Value, fill = Model)) +
    geom_bar(stat = "identity", position = "dodge") + labs(title="(d)")
}
}
ggarrange(p1,p2,p3,p4,ncol = 2, nrow = 2)


}
```

APPENDIX C:
SAS CODE

C.1 Testing for Normality and Homoscedasticity

```
libname MRC "C:\Users\student\Documents\HONOURS\Project\New new";
run;

PROC IMPORT OUT=MRC.half DBMS=XLSX
    DATAFILE="C:\Users\student\Documents\HONOURS\Project\New
new\half.xlsx";
        run;

PROC IMPORT OUT=MRC.ultra DBMS=XLSX
    DATAFILE="C:\Users\student\Documents\HONOURS\Project\New
new\ultra.xlsx";
        run;

DATA mrc.half;
set mrc.half;
drop Racetype;
run;

DATA mrc.ultra;
set mrc.ultra;
drop Racetype;
run;

title 'Comparative Analysis of Age and Gender Categories';
proc univariate data=mrc.half noprint;
   class Agecat Gender;
   histogram 'Finish_time.mins.'n 'Race_pace.mins.'n/
         odstitle = title;
                inset kurtosis skewness;
        qqplot 'Finish_time.mins.'n 'Race_pace.mins.'n;
run;

ods graphics on;

title bold "Half Marathon Finish Time ANOVA for Age Categories";
PROC ANOVA DATA= mrc.HALF PLOTS(MAXPOINTS=50000);
CLASS Agecat;
MODEL 'Finish_time.mins.'n = agecat ;
MEANS agecat / HOVTEST=LEVENE BON;
run;

title bold "Half Marathon Finish Time ANOVA for Gender Categories ";
PROC ANOVA DATA=MRC.HALF PLOTS(MAXPOINTS=50000);
CLASS Gender;
MODEL 'Finish_time.mins.'n = Gender;
MEANS Gender / HOVTEST=LEVENE BON;
run;

title bold "Half Marathon Race Pace ANOVA for Age Categories";
PROC ANOVA DATA=MRC.HALF PLOTS(MAXPOINTS=50000);
CLASS agecat;
MODEL 'Race_pace.mins.'n = agecat;
```

```
MEANS agecat / HOVTEST=LEVENE BON;
run;

title bold "Half Marathon Race Pace ANOVA for Gender Categories";
PROC ANOVA DATA=MRC.HALF PLOTS(MAXPOINTS=50000);
CLASS Gender;
MODEL 'Race_pace.mins.'n = Gender;
MEANS Gender / HOVTEST=LEVENE BON;
run;

title bold "Ultra Marathon Finish Time ANOVA for Age Categories";
PROC ANOVA DATA=MRC.ULTRA PLOTS(MAXPOINTS=50000);
CLASS agecat;
MODEL 'Finish_time.mins.'n = agecat;
MEANS agecat / HOVTEST=LEVENE BON;
run;

title bold "Ultra Marathon Finish Time ANOVA for Gender Categories";
PROC ANOVA DATA=MRC.ULTRA PLOTS(MAXPOINTS=50000);
CLASS Gender;
MODEL 'Finish_time.mins.'n = Gender;
MEANS Gender / HOVTEST=LEVENE BON;
run;

title bold "Ultra Marathon Race Pace ANOVA for Age Categories";
PROC ANOVA DATA=MRC.ULTRA PLOTS(MAXPOINTS=50000);
CLASS agecat;
MODEL 'Race_pace.mins.'n = agecat;
MEANS agecat / HOVTEST=LEVENE BON;
run;

title bold "Ultra Marathon Race Pace ANOVA for Gender Categories";
PROC ANOVA DATA=MRC.ULTRA PLOTS(MAXPOINTS=50000);
CLASS Gender;
MODEL 'Race_pace.mins.'n = Gender;
MEANS Gender / HOVTEST=LEVENE BON;
run;


TITLE "Half Marathon Correlation Analysis";
PROC CORR DATA=MRC.HALF;

proc corr data=mrc.half;
title 'Half Marathon Examination of Correlation Matrix';
run;

proc corr data=mrc.ultra;
title 'Ultra Marathon Examination of Correlation Matrix';
run;
```

C.2 Testing for Independence

```
libname MRC "\\sunrga.stb.sun.ac.za\home\ebw\20941714\Documents\MRC
Project\Data";
run;

PROC IMPORT OUT=MRC.half DBMS=XLSX
    DATAFILE="\\sunrga.stb.sun.ac.za\home\ebw\20941714\Documents\MRC
Project\Data\half.xlsx";
       run;

PROC IMPORT OUT=MRC.ultra DBMS=XLSX
    DATAFILE="\\sunrga.stb.sun.ac.za\home\ebw\20941714\Documents\MRC
Project\Data\ultra.xlsx";
       run;

DATA mrc.half;
set mrc.half;
drop Racetype;
run;
DATA mrc.ultra;
set mrc.ultra;
drop Racetype;
run;

Proc FREQ data= mrc.half;
 tables Gender*Agecat / chisq nocol norow nopercent;
 weight 'Finish_time.mins.'n ;
 title "Half Marathon Finish Time Investigation of Independence"
 run;

 Proc FREQ data= mrc.half;
 tables Gender*Agecat / chisq nocol norow nopercent;
 weight 'Race_pace.mins.'n ;
 title "Half Marathon Race Pace Investigation of Independence"
 run;

 Proc FREQ data= mrc.ultra;
 tables Gender*Agecat / chisq nocol norow nopercent;
 weight 'Finish_time.mins.'n ;
 title "Ultra Marathon Finish Time Investigation of Independence"
 run;

 Proc FREQ data= mrc.ultra;
 tables Gender*Agecat / chisq nocol norow nopercent;
 weight 'Race_pace.mins.'n ;
 title "Ultra Marathon Race Pace Investigation of Independence"
 run;
```