Università degli studi di Padova
Dipartimento di Scienze Statistiche
Corso di Laurea Magistrale in

Scienze Statistiche

ANALISI DEI FATTORI DI RISCHIO PER LA PREVENZIONE DEGLI INFORTUNI DEI CALCIATORI

Relatore: Prof. Mauro Bernardi
Dipartimento di Scienze Statistiche
Correlatrice: Prof. Manuela Cattelan
Dipartimento di Scienze Statistiche

Laureando: Vittorio Costaperaria
Matricola N 1243426

Anno Accademico 2021/2022

Indice

Elenco dei codici

Elenco delle tabelle

Elenco delle figure

Introduzione

Negli ultimi anni i dati provenienti dagli sport sono sempre più richiesti ed utilizzati a fini di analisi statistica. Basta andare a cercare online paper su questo argomento per rendersi conto di quante ricerche, approfondimenti e studi siano stati fatti recentemente per valutare gli sport a 360 gradi. Nella review presentata da Andrade et al. (2020), ad esempio, sono stati presi in considerazione 2961 articoli dal 2014 al 2019 riferiti alla relazione tra carico di lavoro e rischio di infortunio negli sport di squadra, di cui solo il 10% è stato scritto prima del 2016. Questo cambio di tendenza è anche dovuto al fatto che la disponibilità dei dati nello sport è sempre maggiore grazie a tecnologie ogni anno più nuove e ad un sempre maggiore interesse nella cura di ogni aspetto dell'attività fisica (Bourdon et al., 2017).

In questo contesto di analisi dei dati sportivi, per ottimizzare le performance dei singoli atleti e di squadra, l'azienda **Exelio srl**, nata nel 2009, realizza soluzioni tecnologiche per lo sport in generale. Il progetto principale è stato denominato **"GPEXE"**, una soluzione integrata hardware e software basata su GPS e finalizzata alla misurazione delle prestazioni atletiche degli sportivi professionisti (dal sito $www.infojobs.it/exelio - s.r.l.$).
L'azienda, dunque, produce pettorine sportive per gli atleti, in particolare per i giocatori delle squadre di calcio, al cui interno inserisce hardware e software che rilevano dati biometrici ed altre variabili utili a quantificare l'attività fisica svolta. Le sopracitate pettorine, dunque, possono essere usate in qualsiasi momento dai giocatori delle squadre, sia durante gli allenamenti sia durante le partite, nonostante nel secondo caso la rilevazione dei dati di posizione non sia sempre possibile a causa delle interferenze delle infrastrutture che circondano il campo al GPS. In questo modo, il team di preparatori può vedere in tempo reale la performance del

singolo giocatore, nonchè avere un database di tutte le attività fisiche svolte nel tempo dalla propria squadra.

L'obiettivo di questo elaborato è analizzare le informazioni raccolte tramite "GPEXE", in particolare si è cercato di capire quali siano i fattori di rischio per gli infortuni dei giocatori e di prevedere eventuali stop fisici nell'arco del tempo.

In particolare, la tesi si organizza nel modo seguente: nel Capitolo 1, si presentano i due insiemi di dati a disposizione. Il primo è un database fornito dall'azienda Exelio che raccoglie i dati di allenamenti e partite relativi al campionato di Serie A da maggio ad agosto circa del 2020. Questo insieme di dati è stato rielaborato e modificato fino ad arrivare ad ottenere una struttura il più adatta possibile al nostro scopo, costruendo anche la variabile risposta, di tipo dicotomico, che indica se il giocatore in questione si è infortunato oppure no in quel giorno. Il secondo è un database relativo ai soli giocatori che hanno subito uno stop durante questi mesi in cui si hanno maggiori informazioni relative all'infortunio.

Il Capitolo 2 presenta le variabili che verranno usate nell'implementazione dei modelli. In particolare, alcune di queste sono state scelte dopo una revisione della letteratura presente sull'argomento. Si sono, quindi, riportati i risultati ottenuti in vari articoli e l'analisi esplorativa delle suddette variabili. Questa analisi è stata fatta sia comparando le osservazioni delle singole variabili tra le squadre, sia mettendo in evidenza i diversi valori nelle sessioni in cui il giocatore non si infortuna con quelle dove, invece, avviene lo stop dell'atleta.

Nel Capitolo 3, invece, vengono illustrati i modelli che verranno usati. Viene, in particolare, presentato l'approccio bayesiano, che si riesce ad applicare al caso logistico sfruttando la tecnica di data augmentation suggerita da Polson et al. (2013). Scegliendo per le distribuzioni a-priori dei coefficienti della regressione logistica bayesiana una distribuzione Normale e condizionandosi alla variabile generata dalla distribuzione Polya-Gamma, appositamente creata, si ottiene come distribuzione a-posteriori per i parametri del modello una forma chiusa Normale con media e varianza aggiornate. Inoltre, la distribuzione a-posteriori della variabile Polya-Gamma, resta tale con una semplice modifica dei parametri. In questo

modo, si può costruire un algoritmo Gibbs Sampling adatto.

Infine, il Capitolo 4 presenta i risultati ottenuti applicando le tecniche presentate sia ai dati forniti dall'azienda sia a delle simulazioni. Queste ultime, in particolare, sono state implementate dopo un'analisi dell'andamento approssimato delle covariate utilizzate. In questo modo, si è potuto aumentare il numero di giocatori infortunati ispirandosi dai dati realmente raccolti e, sfruttando la possibilità di simulare contemporaneamente più catene fatte partire da punti diversi, si è valutata la convergenza di queste ultime e le nuove stime ottenute.

Capitolo 1

Dataset

1.1 I dati

Il dataset fornito dall'azienda è relativo a 5 squadre di Serie A nel periodo post COVID-19, ossia da inizio maggio a inizio agosto 2020. Lo studio ha preso in considerazione questi mesi in quanto sembrava interessante analizzare l'andamento degli infortuni dopo un periodo di pausa obbligatoria da allenamenti e partite.

Il lockdown causato dalla pandemia di COVID-19, infatti, ha portato negli allenatori, preparatori atletici, ma anche studiosi del settore, grandi incertezze riguardanti i cambiamenti fisiologici indotti nei giocatori professionisti di calcio, in quanto il ritorno allo sport dopo uno stop forzato e inatteso più lungo della normale sosta stagionale, non era mai stato sperimentato. Inoltre, in aggiunta ad un ovvio calo nelle performance dei giocatori, si può ipotizzare che il lockdown abbia portato ad un aumento nel rischio di infortunio, anche perchè è stato rilevato in altri studi che nella preparazione precampionato c'è una maggiore incidenza di infortuni per sovraccarico (Bisciotti et al., 2020).

È stato possibile scaricare dal sito dell'azienda un dataset degli allenamenti e delle partite in forma anonima. Questi dati, in realtà, sono disponibili solo per lo staff della singola squadra in questione e non sono di dominio pubblico, in quanto mostrano nello specifico ogni sessione sostenuta dai giocatori del team di riferimento. Il dataset riassume l'attività del singolo giocatore nelle varie sessioni ed è composto

da 130 variabili (Tabella A.1, in Appendice A), che d'ora in avanti verrà chiamato "Dataset Sintetico". Ogni riga, quindi, rappresenta il singolo allenamento di ciascun giocatore.

Insieme a queste informazioni, l'azienda *Exelio srl* ha fornito anche dati riguardanti gli infortuni occorsi ai giocatori, questo database verrà chiamato d'ora in avanti "Dataset Infortuni".

1.2 Dataset Sintetico

Il "Dataset Sintetico" è composto da massimo 130 variabili, divise in gruppi a seconda della tipologia della stessa. Ogni squadra può scegliere se scaricarle tutte oppure no in base agli interessi dello staff tecnico del team. Le variabili riferite al lavoro cardiocircolatorio, come la frequenza cardiaca, sono disponibili solo se il giocatore indossa anche un cardiofrequenzimetro, quindi di sessione in sessione questi dati possono essere presenti o meno per il singolo giocatore.
Inizialmente il TEAM1 ha al suo interno 95 variabili, il TEAM2 ne ha 79, il TEAM3 81, il TEAM4 88, mentre il TEAM5 ha 106 variabili. Nessuna delle squadre presenti nell'analisi, quindi, ha utilizzato tutte le variabili disponibili nel programma.

1.2.1 Pulizia dei dati

All'interno dei dataset delle squadre, però, si sono riscontrati vari problemi sui dati originali. Nel seguito si spiegano le operazioni fatte per ottenere il dataset finale che verrà poi utilizzato nelle implementazioni dei modelli e delle simulazioni.

Alcune variabili sono definite da "zone", ossia delle fasce di, ad esempio, distanza percorsa o tempo trascorso, che hanno dei valori di default impostati dal programma dell'azienda. Ogni squadra, però, può scegliere di cambiare le soglie a proprio piacimento a seconda dei propri interessi, quindi, per poter confrontare tali variabili tra squadre diverse, si è dovuta effettuare un'attenta analisi preliminare. Per

prima cosa, quindi, si è deciso di confrontare le soglie delle zone delle varie variabili in modo da uniformarle.
Per le zone relative ai *"Metabolic Power Events"* (MPE), ossia quei momenti in cui c'è una forte richiesta da parte del corpo di tanta energia, non tutte le squadre hanno impostato la stessa soglia. Per queste rilevazioni, in particolare, si hanno dei numeri interi che indicano quante volte il corpo ha richiesto un significativo sforzo energetico in una determinata fascia di tempo, distanza o velocità (esempio in Figura 1.1).

Per quanto riguarda la massima velocità raggiunta, le fasce dei vari team sono uniformi, tranne che per il TEAM5 dove le due soglie sono leggermente diverse. Vista la differenza minima nel valore delle soglie citate, ossia 19.01 km/h al posto di 19.8 km/h e 24.98 km/h al posto che 25.2 km/h, si sono considerate tutte le zone dei cinque team come uguali.
Inoltre, per quanto riguarda gli eventi dei "MPE" avvenuti per le fasce della distanza percorsa, il TEAM4 ha soglie diverse da quelle delle altre squadre (0, 20, 30 metri, mentre gli altri team hanno 0, 10, 20 metri). Si sono, quindi, create 2 zone con un'unica soglia posta a 20 metri.

Per la variabile *"speed events"*, vale a dire il numero di scatti effettuati ad una certa velocità o per un minimo di secondi, le zone ad essa relative sono quelle che hanno creato più problemi. In particolare, per poter confrontare quelle calcolate sulla distanza percorsa, le uniche presenti in tutte le squadre, si sono prese come riferimento le fasce del TEAM2, ossia tre zone da 0 a 19.8 km/h, da 19.8 a 25.2 km/h e oltre i 25.2 km/h. Si sono dovute, quindi, aggregare alcune fasce delle altre squadre per avere dei dati uniformi.

Infine, per quanto riguarda le zone relative alla *"potenza"*, sono comuni tra i team solo quelle riferite alla distanza percorsa, ma ogni squadra le ha impostate con soglie diverse. Per non perdere del tutto queste variabili, si è deciso di trovare una soluzione di compromesso con un'unica soglia variabile tra le squadre, ossia tra i 20 e i 25 Watt su kg.

Le altre zone non sono state prese in considerazione in quanto le variabili non erano presenti in tutte le squadre a disposizione.

La variabile *"athlete"*, normalmente codificata come "PLAYER" più il numero identificativo del giocatore, in alcuni casi presenta un asterisco dopo la suddetta

mpe zones	ENABLED					
mpe distance thresholds m	0	Z1	20.00	Z2	30.00	Z3
mpe time thresholds s	0	Z1	3.00	Z2	6.00	Z3
mpe max speed thresholds km/h	0	Z1	19.80	Z2	25.20	Z3

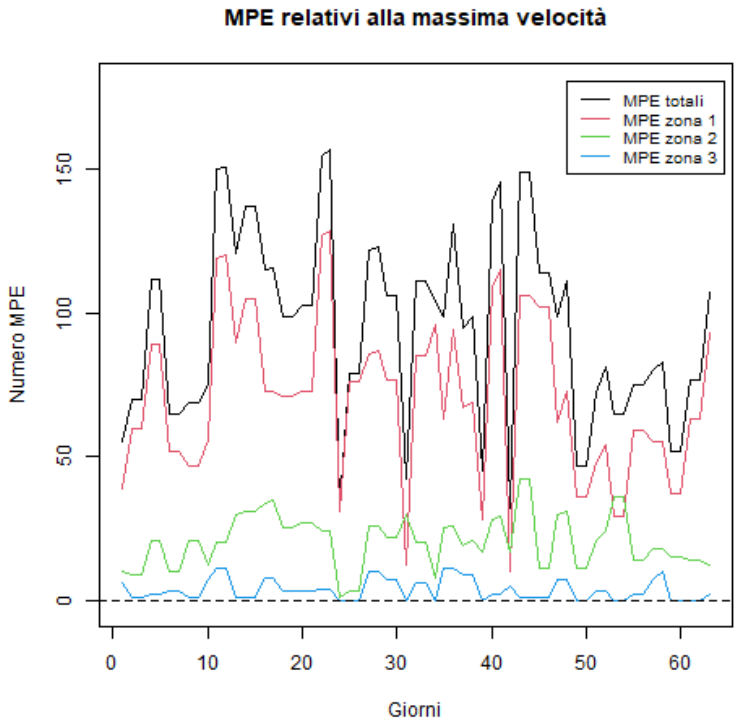

Figura 1.1: Esempio delle *zone* per i MPE

stringa. Questo perchè, in quella sessione, le statistiche del giocatore non sono state prese in considerazione dal team di preparatori per la valutazione delle performance di squadra in quanto il giocatore in questione aveva, per esempio, fatto un allenamento differenziato o era subentrato a partita in corso. Tuttavia, anche in questo caso il software `R` legge in maniera errata il nome del giocatore. Si è, quindi, deciso di togliere l'asterisco dalla stringa per uniformare i nomi.

Tutte le variabili di durata, come ad esempio la lunghezza dell'allenamento o il tempo camminato in quella sessione, non esistendo quella tipologia di variabile su

R, vengono lette come caratteri e di conseguenza non sono confrontabili tra loro. Per questo motivo, si sono trasformate in minuti con decimali per poterle sfruttare al meglio.

La variabile *"next.match"*, che conta i giorni mancanti alla prossima partita della squadra in questione, in alcuni casi assume il valore "None" essendo finito il campionato. Per poter considerare questa variabile come numerica si è modificato quel valore in "Inf".

1.2.2 Creazione del Dataset Sintetico Finale

Ai fini dell'analisi, si sono utilizzate solo le variabili presenti in tutte le squadre. Una volta verificato quali fossero effettivamente queste informazioni, dunque, dal dataset pulito di ogni squadra si sono estratte le 50 variabili comuni presentate in Tabella 1.1.
Si sono poi presi in considerazione solo i giocatori che si sono effettivamente infortunati nel periodo di interesse. Questa informazione è stata trovata nel secondo dataset disponibile, chiamato "Dateset Infortuni". Il numero di giocatori in questione risulta essere 45, di cui 10 si sono infortunati due volte in stagione. In realtà, però, il PLAYER13 e il PLAYER20 non esistono nel dataset dei team. Si è anche notato che il PLAYER43 risulta essere presente in due squadre, TEAM4 e TEAM5, per alcune sessioni. Si sono allora chiesti chiarimenti all'azienda che ha fornito i dati e si è verificato che quel giocatore fa parte solo della rosa del TEAM5.

Le variabili dei giocatori infortunati sono state prese dal momento di inizio degli allenamenti post pausa COVID fino al giorno del loro infortunio, se presente quella sessione per il giocatore in questione. In particolare, si è riscontrato che il PLAYER6 e il PLAYER41 non indossavano la pettorina il giorno dell'infortunio, quindi si sono considerati i dati fino all'ultima data disponibile prima dello stop. Inoltre, il PLAYER33 e il PLAYER38 non hanno osservazioni disponibili in quanto il TEAM5 ha cominciato ad usare le pettorine dalla seconda settimana di allenamenti e questi giocatori si sono infortunati durante i primi giorni di sforzi. Infine, per il PLAYER32 e il PLAYER35 sono stati presi i periodi tra il recupero

del primo infortunio al secondo, in quanto per il primo giocatore non c'erano osservazioni prima del primo stop, mentre per il PLAYER35 c'erano soltanto 2 date esclusa quella del giorno in cui si è infortunato per la prima volta.

Riassumendo, quindi, si sono presi i giocatori con identificativo dal numero 1 al 45, esclusi il 13, il 20, il 33 e il 38. Di questi, per il PLAYER32 e il PLAYER35 è stato preso il periodo tra il primo infortunio e il secondo per mancanza di dati prima del primo stop. Infine, il PLAYER6 e il PLAYER41 non indossavano la pettorina il giorno dell'infortunio, quindi le osservazioni arrivano fino al giorno precedente dello stop.

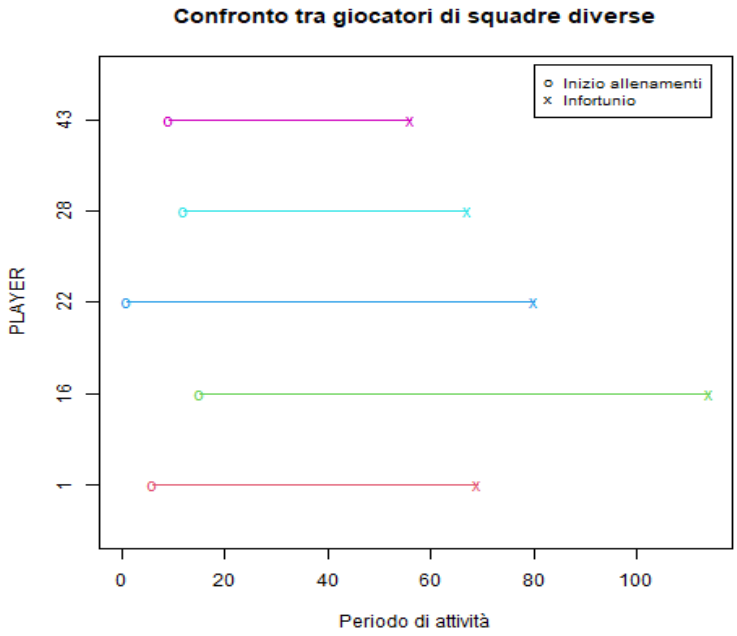

Figura 1.2: Esempio del periodo di attività in giorni di alcuni giocatori. Il giorno "0" corrisponde al 7 maggio 2020.

Una volta creato questo sotto-dataset con solo i giocatori infortunati, si è fatta una pulizia dei dati dalle sessioni di allenamenti e partite non informative. Alcune squadre, infatti, in una singola sessione hanno registrato più *drill*, vale a dire i singoli esercizi da cui può essere composto un allenamento, tuttavia queste informazioni sono già presenti nella sessione completa dell'allenamento o della partita

e si è, dunque, andati ad eliminarle. Nello specifico questo lavoro di pulizia è stato fatto sfruttando prima la variabile *"category"* e poi eliminando le righe che hanno la stessa data di inizio della sessione e che sono cominciate entro 2 ore una dall'altra. Si sarebbe dovuta usare la variabile *"type"*, teoricamente creata proprio per questo tipo di situazione, ma al suo interno non c'è distinzione tra sessione intera (S) e drill (D) e, quindi, non risulta utile a questo procedimento.
Si è anche notato che alcuni team hanno sostenuto un doppio allenamento ad orari diversi, di solito uno al mattino ed uno al pomeriggio, nello stesso giorno. In questo caso non si tratta di informazioni ridondanti. Durante questi passaggi, però, si sono trovati degli errori, infatti nel TEAM3 risultano esserci doppi allenamenti, ma andando a controllare nello specifico i dati è chiaro che si tratta di duplicati creati per errore.

Si sono poi controllate le tipologie di sessione rimaste nella variabile *"category"*, si è deciso di uniformare i nomi in "OFFICIAL MATCH", "FULL TRAINING", "RETURN TO PLAY" e "DIFFERENZIATO". Queste categorie non sono necessariamente presenti in tutte le squadre.
Inoltre, bisogna fare una precisazione per quanto riguarda "DIFFERENZIATO" e "RETURN TO PLAY". In questo dataset, la prima dicitura fa sempre riferimento ad allenamenti non completi non per motivi di infortunio, ma piuttosto a causa di una partita giocata il giorno prima o di carichi di lavoro importanti nei giorni precedenti. "RETURN TO PLAY", invece, si riferisce agli allenamenti dei PLAYER32 e PLAYER35 di ritorno dal primo infortunio.

Nome variabile	Tipo variabile	Descrizione variabile
Fixed		
date.time	data-ora	data e istante preciso di accensione del dispositivo
category	c. nominale	tipologia di sessione compiuta, può cambiare di squadra in squadra
last.match	numerica	giorni passati dall'ultima partita
next.match	numerica	giorni che mancano alla prossima partita
athlete	c. nominale	numero identificativo del giocatore
duration	numerica	indica la durata della sessione del giocatore in minuti e secondi
Main		
total time	numerica	durata massima della sessione tra tutti i giocatori della squadra
distance	numerica	metri percorsi dal giocatore
avg speed	numerica	velocità media del giocatore (km/h)
max speed	numerica	massima velocità raggiunta dal giocatore (km/h)
max acc	numerica	massima accelerazione raggiunta (m/s^2)
max dec	numerica	massima decelerazione raggiunta (m/s^2)
MET, Metabolic Exercise Training		
eq distance	numerica	distanza equivalente che avrebbe percorso l'atleta a velocità costante con la stessa energia usata nell'allenamento
eq distance index	numerica	rapporto tra "eq distance" e "distance"
avg met power	numerica	media del MET, parametri legati all'approccio energetico, velocità per costo energetico
energy	numerica	energia (aerobica) totale spesa (Joule/kg)
an energy	numerica	energia anaerobica (Joule/kg)
an index	numerica	rapporto tra l'energia anaerobica e energia totale
avg VO2	numerica	il consumo medio di ossigeno è un parametro biologico che esprime il volume massimo di ossigeno che un essere umano può consumare nell'unità di tempo per contrazione muscolare (W/kg)
aerobic ratio	numerica	rapporto tra avg "MET power" e "VO2 massimo", dà informazioni sull'intensità media aerobica sostenuta
max met power	numerica	massimo MET (W/kg)
Metabolic Power Events		
met power events	numerica	numero di richieste da parte del corpo di "grande energia"
MPE avg time	numerica	tempo medio di lavoro in un determinato intervallo di tempo (secondi)
MPE avg power	numerica	lavoro medio del Metabolic Power in un determinato intervallo di tempo (W/kg)
MPE rec avg time	numerica	tempo di recupero medio in un determinato intervallo di tempo (secondi)
MPE rec avg power	numerica	recupero medio di Metabolic Power in un determinato intervallo di tempo (W/kg)
MPE t Z1	numerica	indica il numero di eventi che rientrano nella prima soglia, impostata in secondi
MPE t Z2	numerica	indica il numero di eventi che rientrano tra prima e seconda soglia
MPE t Z3	numerica	indica il numero di eventi che rientrano tra seconda e terza soglia
MPE dist Z1	numerica	indica il numero di eventi che rientrano nella prima soglia, impostata in metri
MPE dist Z2	numerica	indica il numero di eventi che rientrano tra prima e seconda soglia
MPE max sp Z1	numerica	indica il numero di eventi che rientrano nella prima soglia, impostata in metri al secondo
MPE max sp Z2	numerica	indica il numero di eventi che rientrano tra prima e seconda soglia
MPE max sp Z3	numerica	indica il numero di eventi che rientrano tra seconda e terza soglia
MECH		
Active Muscle Load	numerica	lavoro totale fatto dall'atleta (Joule/kg)
avg AMP	numerica	"Average Active Muscle Power", potenza media sostenuta dai muscoli attivi, indicatore dell'intensità muscolare (W/kg)
Eccentric Index	numerica	rapporto tra "Active Muscle Power" e "Mechanical Power"
Locomotion		
walk time	durata	minuti e secondi camminati nella sessione
walk distance	numerica	metri camminati nella sessione
walk energy	numerica	energia spesa nella camminata (Joule/kg)
run time	durata	minuti e secondi di corsa nella sessione
run distance	numerica	metri corsi nella sessione
run energy	numerica	energia spesa correndo (Joule/kg)
Speed Zones		
distance sp Z1	numerica	distanza percorsa all'interno della prima soglia di velocità (metri)
distance sp Z2	numerica	distanza percorsa tra prima e seconda soglia
distance sp Z3	numerica	distanza percorsa sopra la seconda soglia
Power Zones		
distance p Z1	numerica	distanza percorsa sotto l'unica soglia di potenza (metri)
distance p Z2	numerica	distanza percorsa oltre l'unica soglia di potenza
Acc Zones		
acc events	numerica	numero di accelerazioni effettuate, tiene conto o di una certa accelerazione raggiunta o di un minimo di secondi di durata dello sforzo
Dec Zones		
dec events	numerica	numero di decelerazioni effettuate, tiene conto o di una certa accelerazione raggiunta o di un minimo di secondi di durata dello sforzo

Tabella 1.1: Descrizione delle variabili presenti nel *Dataset Sintetico Finale*

1.3 Dataset Infortuni

Il secondo dataset messo a disposizione riguarda gli infortuni occorsi ai giocatori delle 5 squadre in analisi nel periodo di riferimento, ossia dall'inizio degli allenamenti post COVID alla fine del campionato (maggio - agosto 2020 circa).

Questi dati non sono scaricabili tramite il software presente nella pettorina, ma sono stati creati dall'azienda, confrontandosi con lo staff tecnico delle 5 squadre prese in considerazione.

Il dataset è composto da 13 variabili che sono presentate nella Tabella 1.2. In realtà, l'ultima variabile *"secondo infortunio"* non era presente inizialmente ed è stata aggiunta per completare le informazioni contenute nella variabile *"recidiva"*.

Nome variabile	Tipo variabile	Descrizione variabile
team	c. nominale	squadra di appartenenza del giocatore
player	c. nominale	numero identificativo del giocatore infortunato
data infortunio	data	giorno in cui si è infortunato il giocatore
evento	c. nominale	in che occasione si è infortunato (allenamento o gara)
tipologia	c. nominale	tipologia dell'infortunio (muscolare o osteo-articolare)
distretto	c. nominale	descrizione più precisa della parte del corpo interessata nell'infortunio
emilato	c. nominale	parte del corpo dell'infortunio (destra o sinistra)
da contatto	c. dicotomica	come è avvenuto l'infortunio
recidiva	c. dicotomica	era già accaduto un infortunio uguale al giocatore
inizio riabilitazione	data	giorno di inizio della riabilitazione
fine riabilitazione	data	giorno di fine della riabilitazione
giorni inattività	numerica	numero di giorni in cui il giocatore non si è potuto allenare
secondo infortunio	c. dicotomica	l'infortunio del giocatore in questione è il secondo nel periodo considerato

Tabella 1.2: Descrizione delle variabili presenti nel *Dataset Infortuni*

1.4 Costruzione della variabile risposta

Come già sottolineato, l'obiettivo dell'elaborato è quello di prevedere se un giocatore si infortuna in una determinata sessione di allenamento o partita. Di conseguenza si è deciso di costruire una risposta dicotomica, che assume valore 1 quando il giocatore $j-esimo$ al tempo $t-esimo$ si infortuna, e 0 altrimenti. La variabile costruita risulta quindi essere:

$$\mathbf{G}_j = (G_{j,1}, G_{j,2}, \ldots, G_{j,T})$$

dove $G_{j,t}$ è interpretabile come la realizzazione di una variabile casuale $Bin(1, \pi_{j,t})$, con

$$\pi_{j,t} = \frac{e^{\psi_{j,t}}}{1 + e^{\psi_{j,t}}},$$
$$\psi_{j,t} = x_{j,t}^{\top}\beta.$$

Considerando che un giocatore si infortuna circa 2 volte a stagione (Ekstrand et al., 2011), ci si aspetta un forte disequilibrio nel numero di 0 e di 1 presenti nel vettore risposta creato (Tabella 1.3).

0	1
1194	41

Tabella 1.3: Composizione del vettore $\mathbf{G}_j$ indicante la variabile risposta.

Capitolo 2

Analisi esplorativa

2.1 Il rischio di infortunio

In studi recenti, si è cercato di collegare lo sforzo fisico e mentale durante gli allenamenti e le partite con il rischio di infortunio dei giocatori (Rossi et al., 2018; Bowen et al., 2017). Gli infortuni degli atleti professionisti, infatti, hanno un grande impatto sull'industria dello sport, vista la loro influenza sia sullo stato mentale delle persone che sulle performance dei team (Hägglund et al., 2013; Hurley, 2016). Inoltre, il costo associato al recupero e alla riabilitazione è spesso oneroso, sia in termini di cure mediche sia in mancati guadagni per la popolarità del giocatore stesso (Lehmann and Schulze, 2008).

Ricerche recenti hanno dimostrato che gli infortuni in Spagna sono causa di circa il 16% delle assenze stagionali dei giocatori di calcio professionistici, corrispondente a un costo di 188 milioni di euro a stagione (Fernández-Cuevas et al., 2010). Non è, quindi, sorprendente che la previsione degli infortuni stia attraendo un sempre maggiore interesse per i ricercatori, manager e allenatori che sono interessati ad intervenire con azioni appropriate per ridurre la probabilità di infortuni dei giocatori (Rossi et al., 2018).

Per quantificare gli sforzi dei giocatori durante la stagione sportiva, si fa riferimento al *"Training and match load"*, ossia al carico di lavoro svolto dal giocatore in allenamento o partita. Questo sforzo, si divide in *"External load"* e *"Internal*

load".
Il primo si riferisce a tutti i movimenti locomotori dei giocatori e può essere misurato tramite GPS e l'accelerometro. Viene quantificato, quindi, in termini di velocità, accelerazione e distanza. Il "carico interno", invece, fa riferimento alla risposta fisiologica dei giocatori all'"External load". Viene determinato usando il battito cardiaco e il "Rating of Perceived Exertion" (RPE), ossia la valutazione dello sforzo percepito, solitamente indicato su una scala da 6 a 20.

Nel calcio professionistico, tuttavia, esistono piccole indicazioni riguardanti la relazione tra questi indicatori e gli infortuni. Uno studio sul calcio d'elite ha trovato una relazione tra infortuni (non da contatto) ai tessuti molli e una maggiore distanza corsa per minuto nella settimana prima dell'infortunio rispetto ai valori medi stagionali dei giocatori. Non è stato, però, trovato un collegamento tra rischio di infortunio e distanza coperta ad alta velocità (Ehrmann et al., 2016).

I risultati di un altro studio, dove sono state utilizzate equazioni di stima generalizzata per modellare l'associazione univariata tra ciascuna variabile di "carico" e gli infortuni per sovra-allenamento nella settimana seguente, hanno mostrato che soprattutto gli indicatori dell'"External load" sono associati ad un maggiore o minore rischio di infortunio. Viene, in particolare, consigliato il monitoraggio del "Training load" e delle decelerazioni cumulate settimanali per prevenire gli infortuni. Inoltre, un alto rapporto tra carico di lavoro acuto e cronico (ACWR), dove con lavoro acuto si fa riferimento agli allenamenti dell'ultima settimana, mentre con lavoro cronico si parla degli sforzi delle ultime 4 settimane, per la distanza percorsa ad alta velocità dovrebbe essere evitato. Al contrario, sono stati trovati anche dei fattori preventivi, ossia è raccomandato un valore medio dell'"ACWR" per accelerazioni, decelerazioni e "sRPE". In conclusione, questo studio ha mostrato un delicato equilibrio per vari indicatori di lavoro esterni e interni per quanto riguarda l'aumento o la diminuzione del rischio di infortunio (Rossi et al., 2018).

Nello studio di Bowen et al. (2017), i dati sul carico di lavoro e sull'incidenza degli infortuni sono stati monitorati per 2 stagioni su giocatori della squadra "primavera" di squadre di calcio professionistiche. Una regressione logistica multipla è stata usata per comparare i carichi di lavoro cumulati (1, 2, 3 e 4 settimane)

e il rapporto tra carico di lavoro acuto e cronico tra giocatori infortunati e non per delle specifiche variabili: distanza totale percorsa, distanza percorsa ad alta velocità, accelerazioni e carico totale.
I risultati di questo studio hanno mostrato che un valore molto alto ($\geq$9254 m) di metri percorsi in accelerazione, ossia se si sono superati i 2.5 m/s^2, considerando le ultime 3 settimane, è stato associato col più alto rischio di infortunio. Inoltre, il rischio di infortunio non da contatto è aumentato significativamente quando un'alta distanza ad alta velocità (HSD) acuta è combinata con una bassa "HSD" cronica. Il rischio di infortunio da contatto, invece, è maggiore quando il rapporto tra lavoro acuto e cronico della distanza totale percorsa e delle accelerazioni è molto alto (maggiore di 1.76 unità).
In generale, quindi, maggiori carichi di lavoro cumulati e acuti sono associati con un alto rischio di infortunio. Tuttavia, un aumento progressivo nel carico di lavoro cronico può sviluppare nel fisico dei giocatori una maggiore tolleranza a elevati carichi di lavoro acuti e una resilienza al rischio di infortunio. Questo risultato segue la teoria della *"General Adaptation Syndrome"* del dottor Selye (Selye, 1951), secondo cui allenamenti sotto l'ottimale sono insufficienti per produrre miglioramenti. Al contrario, però, stimoli oltre l'ottimale possono portare a sovra-allenamento che è stato largamente associato con una maggiore incidenza degli infortuni. Così, un equilibrio appropriato tra allenamenti, partite e periodo di recupero è necessario per raggiungere ottime performance ed evitare infortuni. Tuttavia, questo bilanciamento non è sempre rispettato adeguatamente nei giocatori, come dimostrato dall'elevato tasso di infortuni nel calcio rispetto a molti altri sport di squadra, con una media di circa 50 infortuni per squadra professionistica a stagione. Questa teoria è in parte ripresa anche dal modello del dottor Gabbet del *"Training-Injury Prevention Paradox"*, secondo cui gli atleti abituati ad elevati carichi di lavoro hanno meno infortuni dei soggetti che si allenano con carichi inferiori (Gabbett, 2016).

In un altro studio ancora, si è investigata l'associazione tra distanza percorsa ad alta velocità (HSR >14.4 km/h), sprint (SR >19.8 km/h) e infortuni tra giocatori di calcio d'elite. Sono stati registrati i carichi di lavoro (sRPE $\times$ durata) degli allenamenti e delle partite insieme anche alla distanza percorsa in "HSR" e in "SR" dei giocatori. Queste variabili sono state modellate contro il rischio di infortunio utilizzando una regressione logistica.

Si è trovato che i giocatori che hanno completato “HSR” e “SR” moderate (701 - 750 m e 201 - 350 m rispettivamente) hanno un rischio minore di infortunio rispetto ai gruppi di riferimento (HSR e SR bassi, $\leq$674 m e $\leq$165 m rispettivamente). Il rischio è maggiore, invece, se i cambiamenti settimanali di queste due variabili sono elevati (tra 351 - 455 m per HSR e tra 75 - 105 m per SR). Giocatori con un alto carico di lavoro cronico, infine, hanno un rischio di infortunio significativamente ridotto quando hanno percorso l’ultima settimana ad un “HSR” moderata comparati al gruppo di riferimento con una “HSR” bassa (Malone et al., 2018).

Nello studio di Ehrmann e altri (Ehrmann et al., 2016) si sono registrati allenamenti e partite di calciatori della Australian Hyundai A-League per una stagione intera. Le variabili registrate sono la distanza totale, la distanza corsa ad alta intensità, la distanza corsa a molto alta intensità, il nuovo carico corporeo e i metri percorsi al minuto. Sono stati presi in considerazione gli infortuni non da contatto ai tessuti molli, in particolare la stagione è stata divisa in blocchi di una o quattro settimane a seconda di quando è avvenuto l’infortunio. Questi blocchi sono stati poi comparati tra loro e con le medie stagionali dei giocatori.
Si è trovato che i giocatori hanno percorso un valore significativamente più alto di metri al minuto nelle settimane precedenti all’infortunio comparato con la media stagionale della squadra, indicando che un aumento nell’intensità degli allenamenti e delle partite porta ad un maggiore rischio di infortunio. Inoltre, nella settimana dell’infortunio il nuovo carico corporeo è stato trovato significativamente inferiore rispetto alla media stagionale.
Per questo motivo queste due variabili (metri percorsi al minuto e nuovo carico corporeo) dovrebbero essere considerate come fattori di rischio per infortuni ai tessuti molli. Tuttavia, si è anche trovato che periodi di relativa impreparazione possono lasciare i giocatori incapaci di far fronte a serie di carichi ad alta intensità durante le partite.

In un altro studio ancora, si è esaminata la relazione tra i carichi di lavoro e gli infortuni di giocatori della Premier League inglese per 3 stagioni. Le variabili registrate tramite il GPS, in questo caso, sono state la distanza totale percorsa, la distanza percorsa a bassa intensità, la distanza percorsa ad alta velocità, la distanza percorsa in scatto, le accelerazioni e le decelerazioni. Da queste variabili

sono stati anche calcolati carichi cumulati e il rapporto tra lavoro acuto e cronico (ACWR).
Il più grande rischio di infortunio non da contatto è stato trovato quando l'esposizione cronica a decelerazioni è bassa (<1731 m) e l'"ACWR" è maggiore di 2 unità. Si è anche trovato che il rischio è 5-6 volte maggiore per le accelerazioni e la distanza percorsa a bassa intensità quando il carico di lavoro cronico è basso e l'"ACWR" è maggiore di 2 unità (Bowen et al., 2020).

Gli studi esistenti in letteratura, come quelli visti fino ad ora, forniscono solamente una conoscenza preliminare su quali fattori pesano maggiormente sul rischio di infortunio, mentre una valutazione del potenziale dei modelli statistici nella previsione degli infortuni per ora manca (Rossi et al., 2018).

I club professionistici sono interessati a modelli pratici, usabili e interpretabili in modo da aiutare nel processo decisionale allenatori e preparatori atletici (Kirkendall, 2010). In questa prospettiva, la creazione di modelli di previsione degli infortuni pone molti ostacoli. Da una parte, il modello deve essere molto accurato in quanto modelli che producono molti "falsi allarmi" sono inutili. Dall'altra, un approccio "black box" (es. reti neurali) non è molto pratico visto che non spiega le ragioni dietro all'infortunio (Rossi et al., 2018).

Uno studio recente considera 12 variabili registrate tramite GPS, in più Rossi et al. (2018) hanno costruito una variabile che tiene conto se il giocatore si è infortunato in passato oppure no. Si è, quindi, costruito un modello multidimensionale per prevedere se un giocatore si infortunerà o meno basandosi sui carichi di lavoro più recenti. In particolare, si è prima creato un "training dataset" composto dalle variabili del singolo giocatore e dei suoi allenamenti, compresa un'etichetta che indica se il soggetto si infortunerà nella successiva sessione oppure no. Dopodichè, si è usato un albero di decisione per allenare un classificatore degli infortuni sul "training dataset". L'intero processo di validazione del modello è stato poi ripetuto migliaia di volte. I risultati hanno mostrato che l'albero decisionale creato può predire quasi tutti gli infortuni (80%) e che etichetta correttamente le sessioni in cui il giocatore si infortuna nel 50% dei casi. Come ulteriore test del potenziale predittivo di questo approccio, si è pensato di provare il modello nel caso in cui i dati fossero raccolti di settimana in settimana andando avanti con la stagione.

Dopo un periodo di raccolta di dati, il modello diventa utile per prevenire gli infortuni, infatti dalla sesta settimana alla fine della stagione ha previsto correttamente 9 infortuni su 14 (Rossi et al., 2018).

Un altro studio simile a quello citato, utilizza tecniche di Machine Learning e Data Mining per costruire modelli più robusti per identificare atleti ad alto rischio di infortunio nel calcio e nella pallamano. Il miglior modello in termini di AUC è stato quello che ha sfruttato la tecnica "SmooteBoost" con un classificatore di base "cost-sensitive" $ADTree$ (López-Valenciano et al., 2018).

2.2 Analisi esplorativa

Comparando, dunque, la letteratura e i dati a nostra disposizione, si sono scelte le seguenti variabili iniziali per i giocatori che si sono effettivamente infortunati:

1. distanza totale percorsa
2. distanza percorsa a bassa intesità (inferiore ai 19.8 km/h)
3. distanza percorsa ad alta intensità (tra 19.8 e i 25.2 km/h)
4. distanza percorsa in scatto (superiore ai 25.2 km/h)
5. numero di accelerazioni sopra la soglia (di default sopra i 2.5 m/s^2)
6. numero di decelerazioni sotto la soglia (di default sotto i -2.5 m/s^2)
7. distanza percorsa rispetto alla potenza metabolica (sopra i 20 - 25 Watt/kg)

Per quanto riguarda la *distanza percorsa* in ogni sessione (Figura 2.1), si è trovato che la media tra tutte le squadre è di 5515.5 metri, mentre se si valutano i team singolarmente si è scoperto che il TEAM2 ha la media più alta. Considerando la mediana, invece, se si analizzano tutti i team assieme si ha un valore di circa 5215 metri, mentre la squadra col valore maggiore è il TEAM3 (5716.6 metri).

Verificando la presenza di outliers, ossia di valori estremi, si sono cercati solo quelli troppo bassi, possibile sintomo del fatto che il giocatore in questione si sia fermato anzitempo durante la sessione. In realtà, però, il valore minimo si ha per il PLAYER32 del TEAM5 non nel giorno del suo infortunio, ma in un allenamento il giorno prima di una partita.

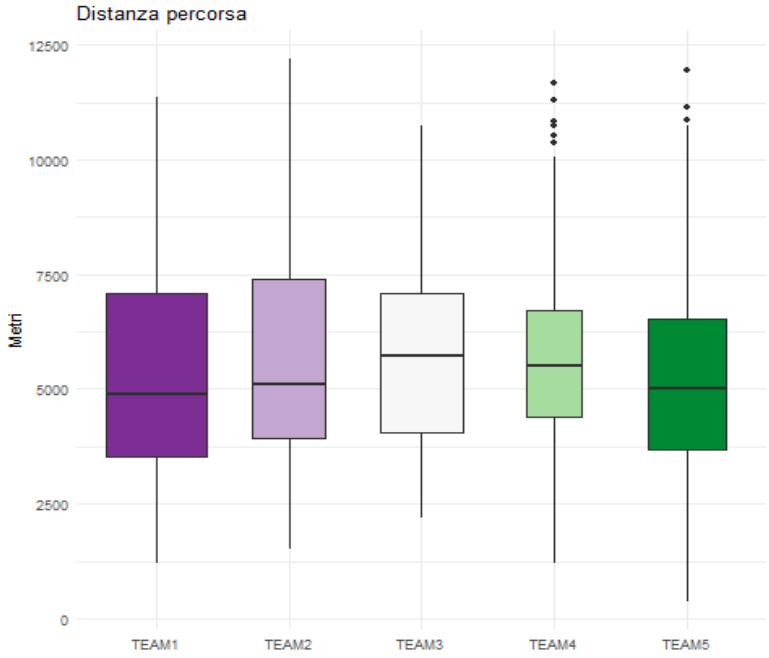

Figura 2.1: Boxplot delle distanze percorse dalle squadre.

La variabile *"distance sp Z1"*, ossia la distanza percorsa ad una velocità inferiore ai 19.8 km/h, mostra dei valori tra le squadre abbastanza discosti (Figura 2.2). Infatti, il valore minimo tra il TEAM3 e il TEAM5 si differenzia di più di 1800 metri corsi. In generale, comunque, le media e le mediane sono tutte tra i 4.5 e i 5.5 chilometri.

La *distanza percorsa nella seconda zona di velocità*, invece, assume dei valori molto inferiori rispetto alla prima zona, con sessioni, in tutte le squadre, dove questa variabile assume anche il valore zero (Figura 2.3). In questo caso è interessante comparare anche i valori massimi: il TEAM4 ha come massima distanza percorsa 520 metri, mentre il TEAM3 ha una sessione con più di 2 chilometri corsi a ritmo sostenuto.

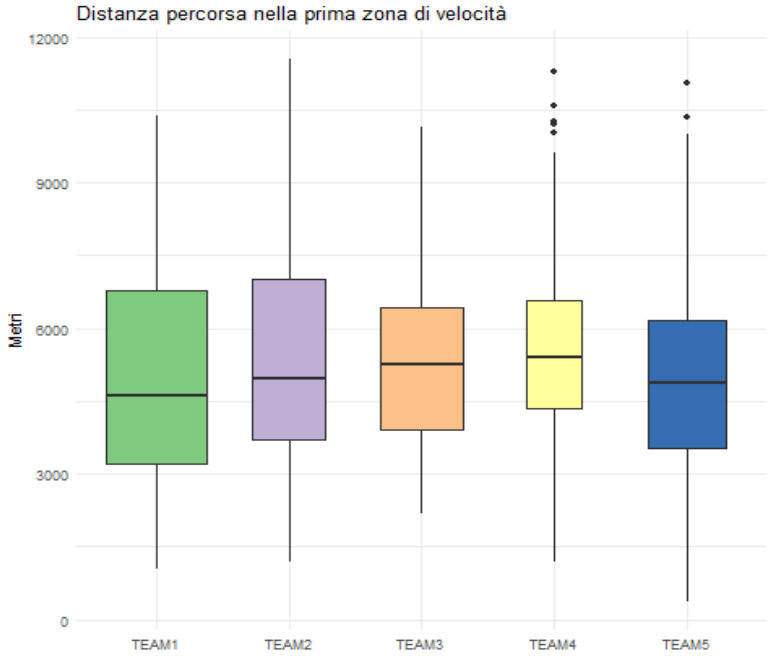

Figura 2.2: Boxplot delle distanze percorse nella prima zona di velocità.

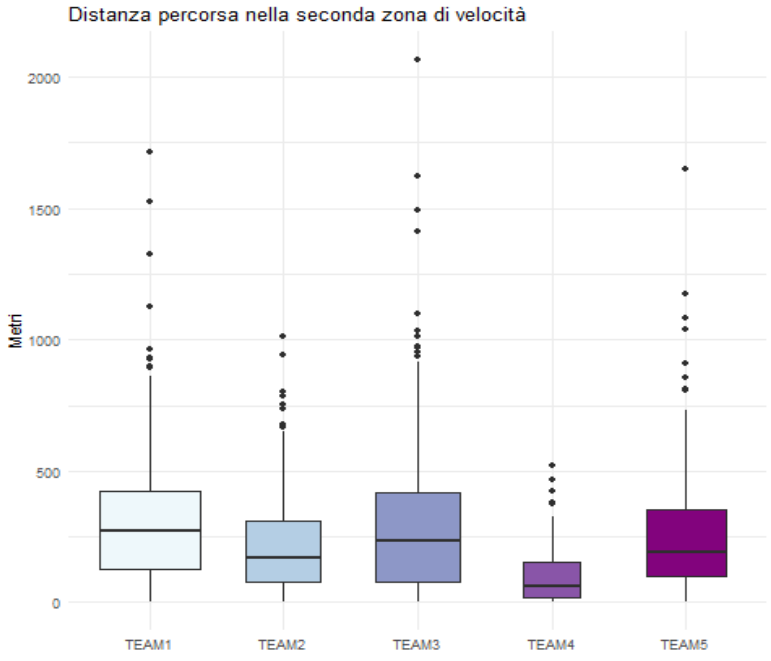

Figura 2.3: Boxplot delle distanze percorse nella seconda zona di velocità.

La *distanza percorsa sopra i 25.2 km/h*, invece, presenta moltissimi zero (circa 1 su 3). Questo significa che il giocatore in quella sessione non ha mai superato questo limite di velocità. La spiegazione probabilmente risiede nel fatto che non in tutti gli allenamenti si fa un lavoro atletico rivolto alla velocità, soprattutto in sessioni di scarico post-partita o in momenti della stagione dove si vuole allenare la resistenza dei giocatori. Per questo motivo, nonostante ai fini dell'analisi non ci siano differenze di sostanza, questa variabile non è stata trasformata in chilometri in quanto, in generale, non sono state rilevate distanze considerevoli in sprint. In generale, infatti, in tutte le squadre i valori medi percorsi in questa zona sono bassi (circa 41 metri a sessione) con picchi che non superano i 400 metri (Figura 2.4).

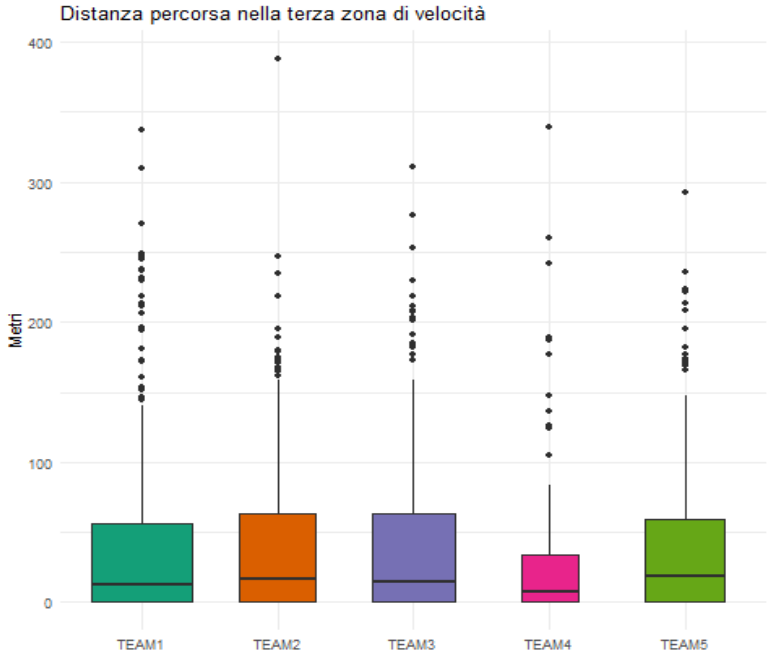

Figura 2.4: Boxplot delle distanze percorse nella terza zona di velocità.

Per concludere la panoramica relativa alle distanze percorse rispetto a diversi indicatori, la variabile *"distance p Z2"* indica i metri percorsi sopra la potenza metabolica di 20-25 W/kg. Per queste rilevazioni, si è dovuto usare un range di valori come soglia in quanto non era possibile uniformare i valori tra le squadre in modo migliore senza eliminare del tutto questa variabile (come spiegato nel Capitolo 1). Nello specifico, il significato di questa soglia è riferito alla spiegazione di "Meta-

bolic Power Event": il VO^2 (energia aerobica) non può, per definizione, superare la potenza metabolica corrispondente al VO^2 massimo dell'atleta, che viene misurato, in Watt/kg, in ogni persona in maniera diretta o indiretta con procedure specifiche. Ogni volta, però, che la potenza metabolica supera il VO^2 attuale del giocatore, per continuare l'esercizio bisogna utilizzare l'energia anaerobica. Nel momento in cui si richiede di sfruttare questa energia, ci si ritrova in una fase critica dello sforzo che corrisponde ad un evento metabolico di potenza (MPE). Quindi, questa variabile conta i metri percorsi nel caso in cui l'atleta supera sia il suo VO^2 attuale sia la soglia prefissata.
Si è trovato che, tranne in pochissime sessione per il TEAM1, il TEAM2 e il TEAM5, si hanno valori sempre superiori ai 50 metri percorsi in questa fascia con picchi di quasi 4 chilometri per il TEAM5. Si hanno, invece, valori massimi bassi sia per il TEAM1 che per il TEAM2, come si vede in Figura 2.5. Per questa variabile, dunque, le medie e le mediane tra le squadre si discostano parecchio, infatti i valori per questi indici di posizione del TEAM3 sono quasi il doppio di quelli del TEAM2.

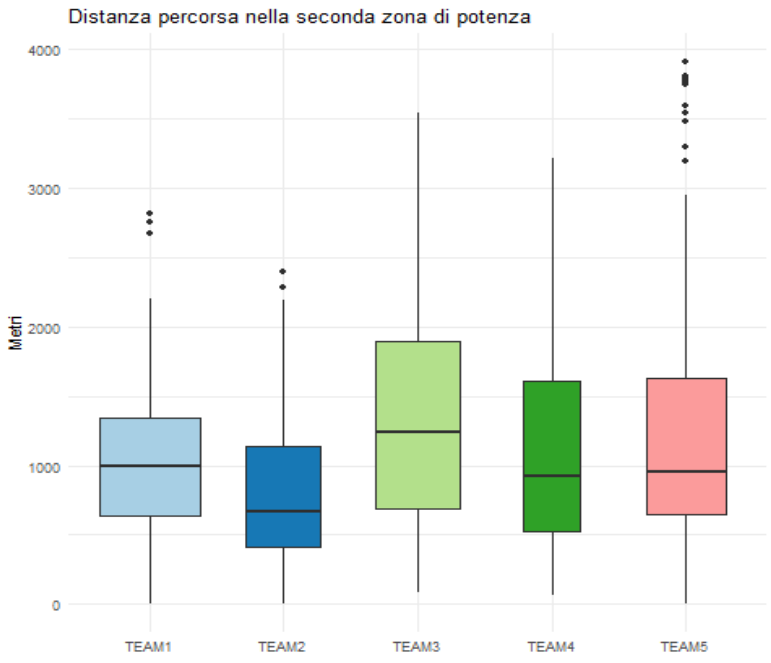

Figura 2.5: Boxplot delle distanze percorse nella seconda zona di potenza.

Infine, le *accelerazioni* sopra i 2.5 m/s^2 e le *decelerazioni* sotto i -2.5 m/s^2 per essere contate, di default devono durare almeno mezzo secondo.
Analizzando le accelerazioni, si sono trovati zeri in tutte le squadre, probabilmente relativi a sessioni di scarico o recupero. Come si può vedere dalla Figura 2.6, il valore massimo si trova nel TEAM5, mentre è interessante notare come la media e la mediana del TEAM3 siano notevolmente maggiori delle altre squadre.

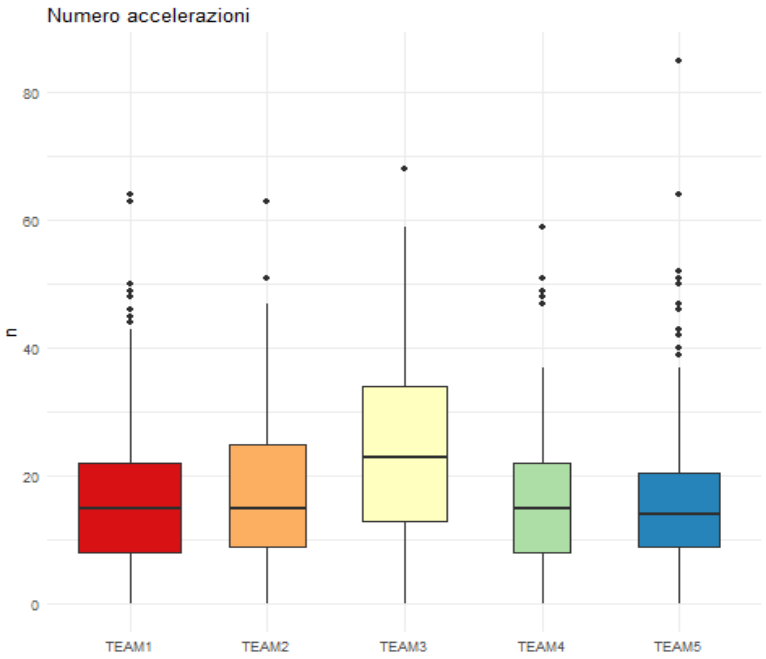

Figura 2.6: Boxplot delle accelerazioni.

Le decelerazioni, invece, hanno i valori degli indici di posizione più simili tra le 5 squadre. Questa variabile è meno intuitiva del numero di accelerazioni, in quanto, solitamente, si pensa a sessioni di allenamento con scatti repentini per migliorare la condizione fisica del giocatore. Anche le decelerazioni, invece, hanno un importante ruolo negli allenamenti di potenziamento e nei cambi di direzione soprattutto in partita.

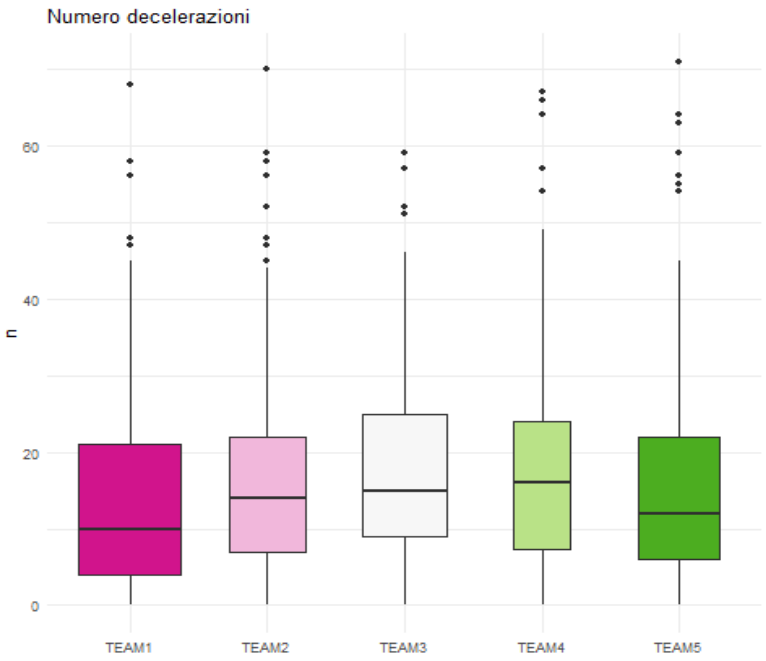

Figura 2.7: Boxplot delle decelerazioni.

2.2.1 Altre variabili utili

Il *Dataset Sintetico Finale*, però, dispone di molte più variabili rispetto a quelle considerate in letteratura, si è dunque deciso di allargare l'analisi esplorativa ad altre variabili di possibile interesse.
In particolare, valutando sia il significato delle singole variabili, sia la correlazioni fra di esse, si sono analizzate la

- media del VO^2,
- velocità massima,
- durata media di un evento di potenza metabolica,
- massima potenza metabolica,
- potenza metabolica media di recupero da un evento di potenza metabolica.

La *media del* VO^2, ossia dell'energia aerobica utilizzata dall'atleta in questione, fornisce una indicazione sulla produzione di energia e, quindi, sugli sforzi sostenuti

in allenamento o partita. Tuttavia, si potrebbe pensare che un alto valore di questa variabile indichi che si sono fatte sessioni dure dal punto di vista fisico. In realtà, come spiegato quando si è parlato della variabile *"distance p Z2"*, i maggiori sforzi si fanno quando il corpo ha bisogno anche dell'energia anaerobica. Infatti, guardando i valori in Figura 2.8, si nota che il TEAM1 è quello con la produzione di VO^2 medio più alta, ma valutando le altre variabili presentate fino ad ora, è una delle squadre che possono sembrare meno allenate.

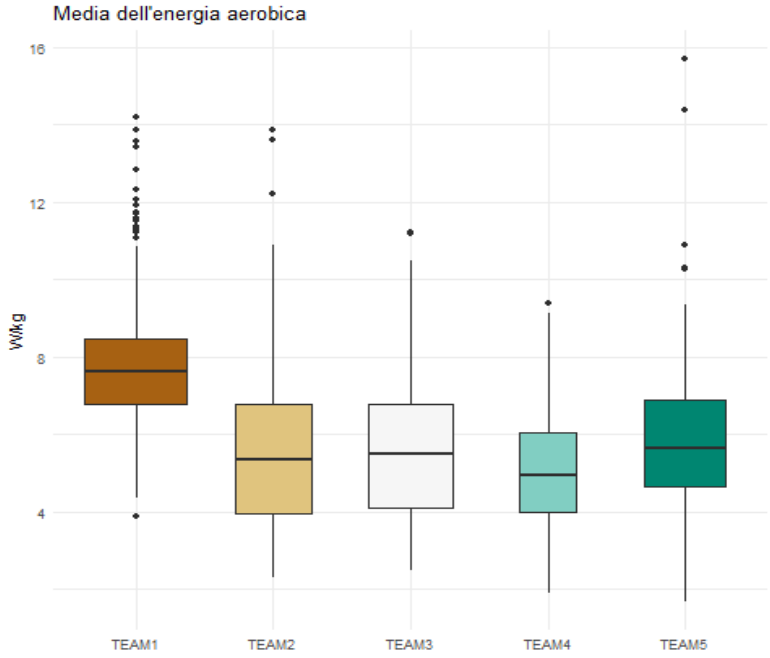

Figura 2.8: Boxplot della media del VO^2.

La *velocità massima* assume valori molto simili tra le squadre, in particolare varia tra un minimo di circa 13 km/h, presumibilmente registrato in una sessione di scarico, ed un massimo di quasi 34 km/h. In realtà, come si vede in Figura 2.9, nel TEAM5 ci sono due osservazioni di molto superiori a questo valore. È evidente che per un calciatore arrivare a oltre 50 km/h di velocità sia attualmente impossibile, si è quindi ritenuto che fossero degli errori. Verificando nel dataset, infatti, si è notato che questi due outliers accadono nello stesso istante durante la partita del 7 luglio 2020 per il PLAYER32 e il PLAYER45, confermando, quindi, l'ipotesi di una rilevazione errata.

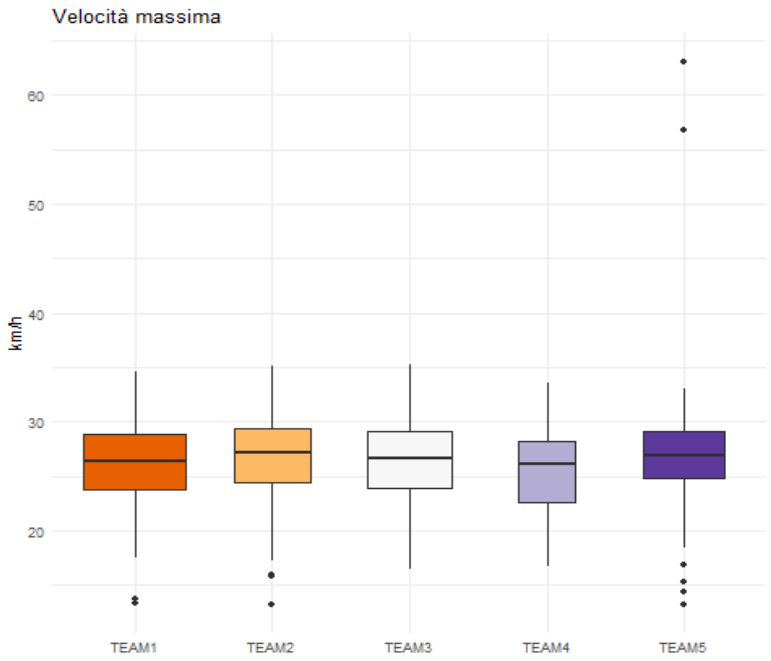

Figura 2.9: Boxplot della velocità massima.

La *durata media dei Metabolic Power Event* fornisce un'informazione su quanto sia durato lo sforzo una volta che il corpo dell'atleta ha dovuto richiedere l'utilizzo anche dell'energia anaerobica. In pratica, è la media della durata di tutti gli eventi metabolici avvenuti in quella sessione. Valori alti, quindi, indicano che nella sessione questo tipo di eventi sono stati particolarmente lunghi, ma non necessariamente che ce ne sono stati tanti e che, quindi, la sessione in questione è stata super impegnativa. I valori nulli, in realtà veramente pochi (Figura 2.10), sono riferiti, invece, a sessioni probabilmente di scarico.
In generale, la media di questa variabile tra il TEAM1 e il TEAM4 ha una differenza di quasi 2 secondi, mentre controllando le osservazioni massime, si può notare come il TEAM3 assume valori molto bassi rispetto alle altre squadre.

La *massima potenza metabolica* viene stimata come il valore massimo tra il prodotto della velocità e il costo energetico della sessione. Per esempio, se due atleti percorrono la stessa distanza, ma il primo sta molto più tempo del secondo, si avrà che il valore della media della potenza metabolica sarà notevolmente più alto per il secondo giocatore in questione. Di conseguenza, anche il valore massimo di questa variabile sarà maggiore per il secondo atleta. Questa misura fornisce, quindi,

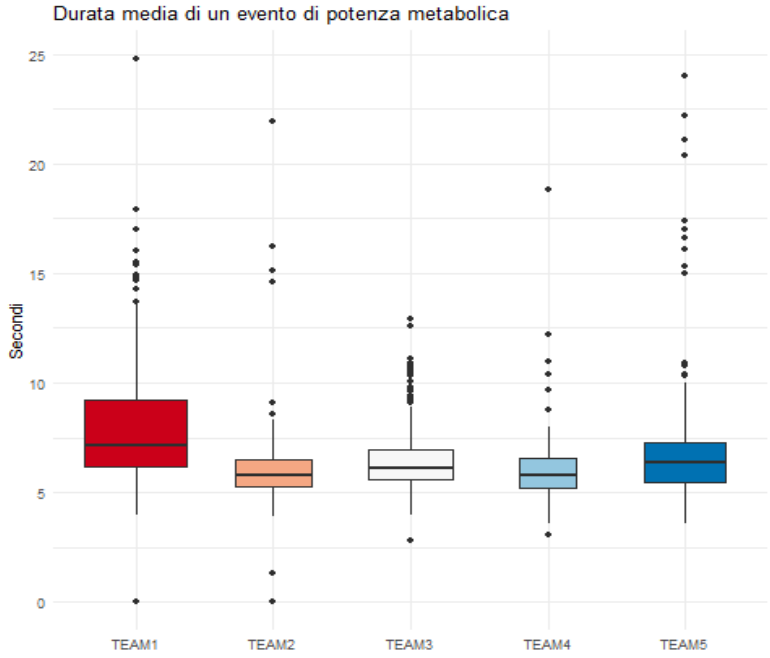

Figura 2.10: Boxplot della durata media dei MPE.

un'informazione sull'intensità massima del lavoro svolto durante l'allenamento o la partita. Tralasciando un unico valore molto alto per il TEAM5 (Figura 2.11), i valori assunti da questa variabile nelle 5 squadre sono molto simili tra loro con una media ed una mediana che si attestano intorno ai 100 W/kg.

La *media della potenza metabolica di recupero da un MPE*, infine, indica quanta potenza metabolica in media l'atleta è riuscito a recuperare dopo un MPE. Fornisce, quindi, un'informazione sulla qualità del recupero del giocatore. In particolare, valori bassi di questa misura indicano che l'atleta non ha avuto tempo per recuperare da eventi in cui è stato richiesto l'utilizzo anche dell'energia anaerobica e, quindi, particolarmente faticosi. A livello di media, mediana e minimo i valori sono vicini tra le squadre (Figura 2.12), mentre il TEAM1 e il TEAM2 assumono in un paio di sessioni le osservazioni maggiori.

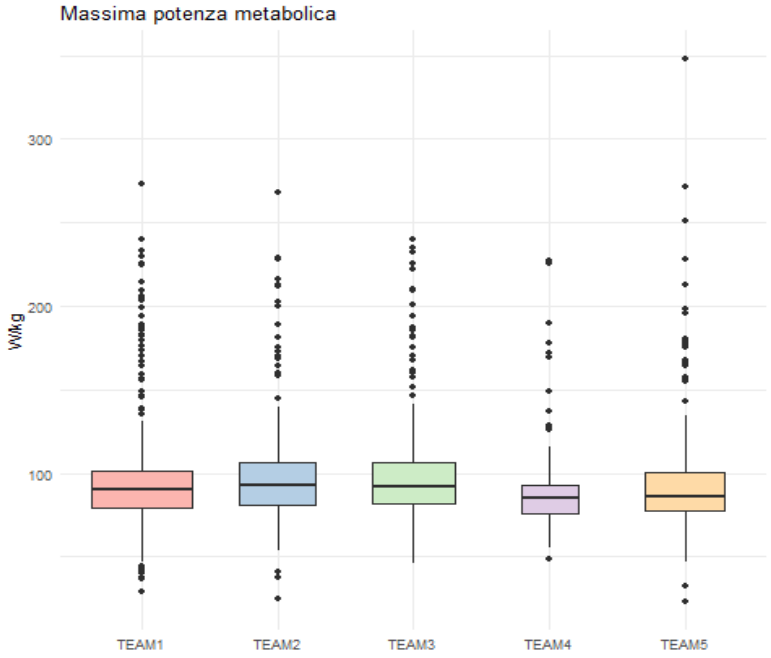

Figura 2.11: Boxplot della massima potenza metabolica.

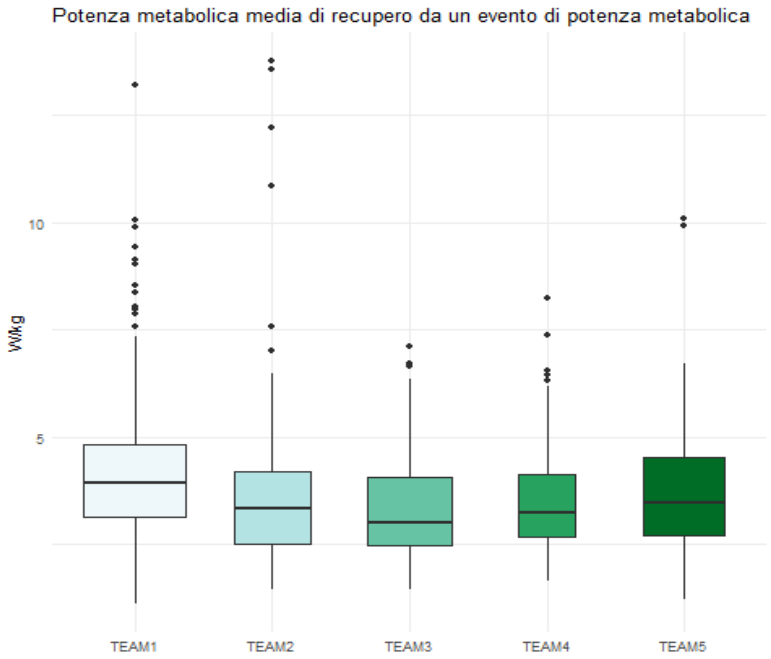

Figura 2.12: Boxplot della potenza metabolica media di recupero da un MPE.

2.3 Relazione con gli infortuni

Fino ad ora si sono considerate le variabili singolarmente, ma visto che l'obiettivo di questo elaborato è capire la loro relazione con gli infortuni si è considerata utile anche un'analisi preliminare tra le covariate prese in considerazione e se il giocatore si è infortunato oppure no nella sessione di riferimento.

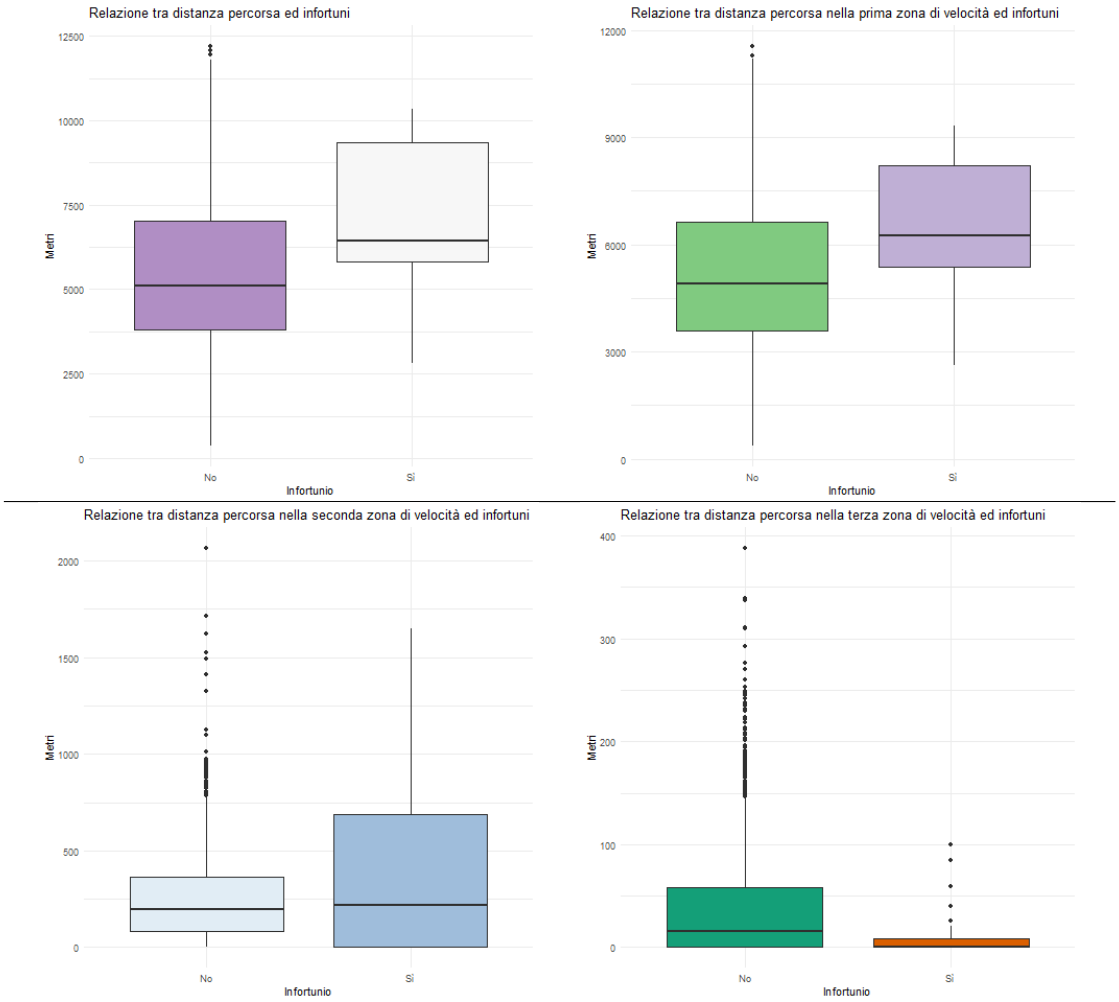

Figura 2.13: Relazione tra infortuni e variabili di distanza percorsa.

Bisogna tenere presente che si stanno mettendo in relazione le osservazioni nelle sessioni in cui i giocatori non si sono infortunati e la sessione in cui gli atleti hanno

subito l'infortunio e che quindi, potenzialmente, hanno terminato in anticipo la partita o l'allenamento.
Inoltre, il numero di sessioni in cui un atleta si infortuna sono in numero di molto inferiori rispetto a quelle dove non si inofortuna (Tabella 1.3).
Le variabili di distanza percorsa e di conteggio di eventi nel giorno dell'infortunio, dunque, dovrebbero avere dei valori inferiori rispetto alle altre sessioni. Per le altre variabili, ci si potrebbe aspettare una differenza non eccessiva.

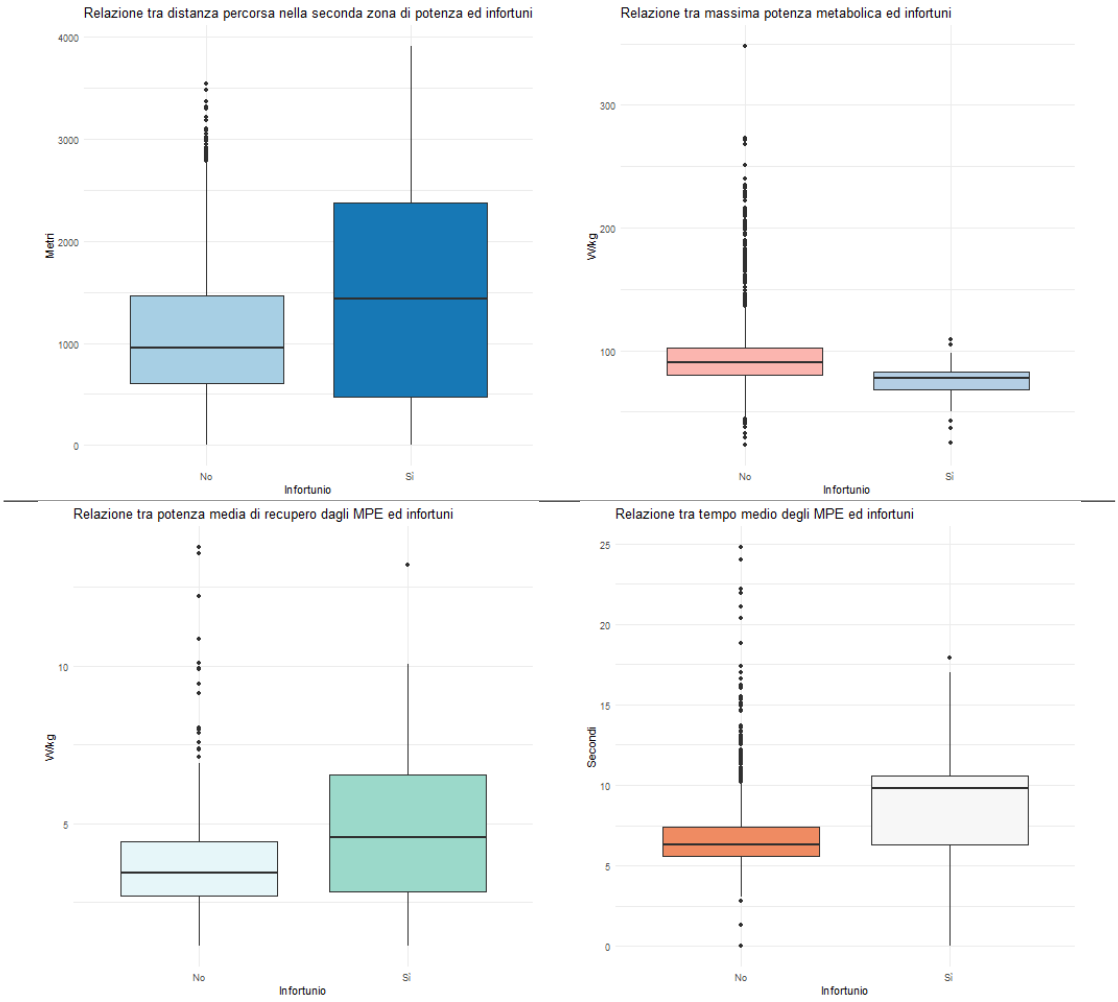

Figura 2.14: Relazione tra infortuni e variabili riferite alla potenza metabolica.

In realtà, però, come si vede nelle Figure 2.13, 2.14 e 2.15, la *distanza percorsa totale* è risultata in media e mediana maggiore nel giorno dell'infortunio, così

come la *distanza percorsa nella prima zona di velocità*, la *distanza percorsa nella seconda zona di velocità* e la *distanza percorsa nella seconda zona di potenza*. Interessante anche notare che la *massima potenza metabolica*, il numero di *accelerazioni*, di *decelerazioni* e la *velocità massima* assumono valori inferiori nel giorno dell'infortunio.

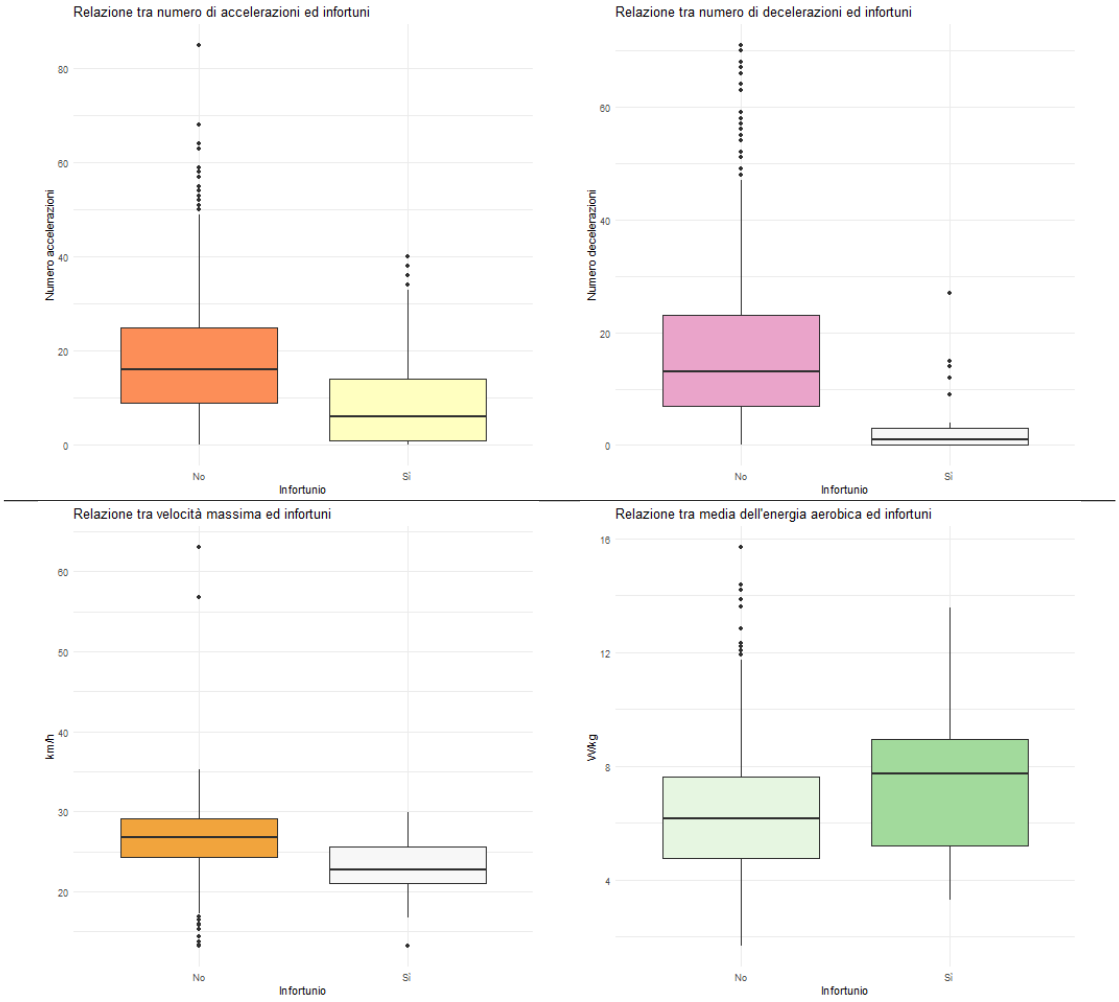

Figura 2.15: Relazione tra infortuni e le altre variabili presentate.

Capitolo 3

Modello logistico bayesiano

3.1 Introduzione all'inferenza bayesiana

L'approccio bayesiano all'inferenza statistica deve il proprio nome al matematico Thomas Bayes, e presenta il vantaggio di aggiungere una propria conoscenza a-priori all'evidenza dei dati. La novità è che come parte del modello, viene introdotta una distribuzione a-priori $\pi(\boldsymbol{\theta})$, per il parametro θ. Questa mostra uno stato di conoscenza o ignoranza dei parametri prima di ottenere i dati (Box and Tiao, 2011). La distribuzione a-priori, dunque, deve essere scelta in modo da rispecchiare le informazioni già a disposizione, provenienti dalla letteratura o da conoscenze o supposizioni, e assegnare i valori che si ritengono essere i più plausibili per quel parametro.

Dato $\mathbf{y} = (y_1, \ldots, y_n)$, vettore dei dati, indichiamo con $f(\mathbf{y} \mid \boldsymbol{\theta})$ la sua funzione di densità. Nel caso di osservazioni indipendenti, la funzione di verosimiglianza risulta quindi :

$$L(\boldsymbol{\theta}) = f(y_1, \ldots, y_n \mid \boldsymbol{\theta}) = \prod_{i=1}^{n} f(y_i \mid \boldsymbol{\theta}), \tag{3.1}$$

e grazie al teorema di Bayes si ha:

$$\pi(\boldsymbol{\theta} \mid \mathbf{y}) = \frac{f(\mathbf{y} \mid \boldsymbol{\theta})\pi(\boldsymbol{\theta})}{f(\mathbf{y})} = \frac{f(\mathbf{y} \mid \boldsymbol{\theta})\pi(\boldsymbol{\theta})}{\int_\Theta f(\mathbf{y} \mid \boldsymbol{\theta})\pi(\boldsymbol{\theta}) d\boldsymbol{\theta}}. \tag{3.2}$$

dove il termine al denominatore, può essere trattato come una costante, perché

non dipende dal parametro d'interesse.
Otteniamo, quindi, che la distribuzione a-posteriori (*full conditional*) del parametro, ovvero condizionata ai dati osservati, è proporzionale a

$$\pi(\boldsymbol{\theta} \mid \mathbf{y}) \propto f(\mathbf{y} \mid \boldsymbol{\theta})\pi(\boldsymbol{\theta}). \tag{3.3}$$

Da questa distribuzione si può fare inferenza sui parametri (Box and Tiao, 2011). Il problema maggiore relativo alle tecniche di inferenza bayesiana, è legato al calcolo esplicito dell'integrale in 3.2, che rappresenta la costante di normalizzazione (Gilks et al., 1995). Se il vettore dei parametri ha un'alta dimensionalità, infatti, la difficoltà nel calcolare l'integrale è molto elevata. Fortunatamente, è possibile superare questa difficoltà se si usa una a-priori coniugata per i parametri: una distribuzione si dice coniugata se $\pi(\boldsymbol{\theta} \mid \mathbf{y})$ segue la stessa distribuzione di $\pi(\boldsymbol{\theta})$. Se non disponiamo di abbastanza informazioni a-priori sul parametro, si possono assumere delle distribuzioni non informative, facendosi guidare dall'informazione osservata in $\mathbf{y}$. Una funzione di densità non informativa è, quindi, una funzione piatta sullo spazio di $\boldsymbol{\theta}$, che assegna probabilità simili a tutti i possibili valori del parametro. Al contrario invece, assumere una a-priori concentrata su alcuni valori dello spazio di $\boldsymbol{\theta}$ abbassa l'importanza dell'informazione fornita dall'osservazione del vettore dei dati $\mathbf{y}$.

Infine, poiché l'informazione per l'inferenza è racchiusa nella distribuzione a-posteriori $\pi(\boldsymbol{\theta} \mid \mathbf{y})$, si possono ricavare le stime degli indici di posizione e di variabilità da questa, e, inoltre, costruire intervalli di credibilità di livello α, tali per cui

$$Pr(\boldsymbol{\theta} \in [a, b] \mid \mathbf{y}) = 1 - \alpha. \tag{3.4}$$

3.2 Markov chain Monte Carlo

La densità a-posteriori non è, quindi, analiticamente del tutto nota nella maggior parte dei casi, in quanto non conosciamo il valore della costante di normalizzazione. Si usano allora tecniche di approssimazione *Monte Carlo*, che semplificano di molto lo studio della distribuzione a-posteriori. In particolare, il metodo di si-

mulazione chiamato “Markov chain Monte Carlo (MCMC)” estrae campioni dalla distribuzione di riferimento facendo simulare una catena di Markov opportunamente costruita per un lungo periodo (Gilks et al., 1995).
Gli algoritmi MCMC permettono di generare una catena di Markov ergodica e invariante per una distribuzione target del tipo 3.3. Tuttavia, gli stati della catena sono correlati tra loro per costruzione e la stima Monte Carlo perde in efficienza ma rimane consistente. Per superare queste problematiche è necessario simulare un campione molto numeroso e ridurne la correlazione attraverso un filtraggio della serie, per esempio estraendo 1 valore generato ogni 10.

3.2.1 Markov chains

Supponiamo di generare una sequenza di variabili casuali $\{X_0, X_1, X_2, \ldots\}$, in modo che ad ogni tempo $t \geq 0$ lo stato successivo, X_{t+1}, non dipende più dalla storia passata della catena $\{X_0, X_1, \ldots, X_{t-1}\}$, ma viene campionato dalla distribuzione $P(X_{t+1}|X_t)$. Questa sequenza è chiamata *catena di Markov* (di primo ordine) e si definisce $P(.|.)$ come la *matrice di transizione* della catena, nel caso con spazio degli stati discreto, o *densità di transizione kernel* nel caso con spazio degli stati continuo.
Sotto condizioni di regolarità, la catena, andando avanti nel tempo, dimentica il suo stato iniziale e eventualmente converge ad un'unica distribuzione stazionaria (o invariante), $\pi(.)$, che non dipende da t o da X_0.
Così, dopo un *burn-in* sufficientemente grande, diciamo di m interazioni, i punti $\{X_l$, con $l = m+1, \ldots, T\}$ saranno campioni dipendenti approssimativamente da $\pi(.)$.
Si può quindi sfruttare i risultati della catena di Markov per stimare il valore atteso di $h(\boldsymbol{X})$, dove $\boldsymbol{X}$ ha distribuzione $\pi(.)$ e $h(.)$ è una qualche funzione di interesse.
Si parlerà di *media ergodica* e la convergenza della catena è garantita dal teorema di ergodicità (Gilks et al., 1995).

3.2.2 Algoritmo Metropolis-Hastings

Come detto, sappiamo che una catena di Markov può essere usata per stimare il valore atteso di $f(\boldsymbol{X})$, ma come si fa a costruirla in modo che la distribuzione stazionaria sia precisamente la distribuzione di interesse?
L'algoritmo *Metropolis-Hastings* proposto da (Hastings, 1970) è uno dei più famosi ed utilizzati per la costruzione di catene di Markov: si supponga di essere nello stato X della catena e di voler esplorare un nuovo stato X', proposto a partire da una densità $P(\boldsymbol{X'} \mid \boldsymbol{X})$. Ad ogni iterazione il valore X' viene accettato con probabilità $\alpha(\boldsymbol{X}, \boldsymbol{X'})$. Una proprietà fondamentale per la convergenza dell'algoritmo è quella di reversibilità, tale per cui

$$\pi(\boldsymbol{X} \mid \mathbf{y})P(\boldsymbol{X'} \mid \boldsymbol{X})\alpha(\boldsymbol{X}, \boldsymbol{X'}) = \pi(\boldsymbol{X'} \mid \mathbf{y})P(\boldsymbol{X} \mid \boldsymbol{X'})\alpha(\boldsymbol{X'}, \boldsymbol{X}). \tag{3.5}$$

Si definisce

$$\alpha(\boldsymbol{X}, \boldsymbol{X'}) = \min\left\{1, \frac{w(\boldsymbol{X'} \mid \boldsymbol{X})}{w(\boldsymbol{X} \mid \boldsymbol{X'})}\right\}, \quad \text{dove} \quad w(\boldsymbol{X'} \mid \boldsymbol{X}) = \frac{\pi(\boldsymbol{X'} \mid \mathbf{y})}{q(\boldsymbol{X'} \mid \boldsymbol{X})},$$

così, soddisfatta la condizione 3.5, si garantisce la costruzione di una catena ergodica e con distribuzione invariante $\pi(\boldsymbol{X} \mid \mathbf{y})$. I pesi w così definiti sono detti *importance weigths*, ma non sono gli unici che portano a catene di Markov ergodiche e con distribuzione invariante 3.3 (Gilks et al., 1995).

3.2.3 Algoritmo Gibbs Sampler

Il *Gibbs sampler* (Geman and Geman, 1984) è un caso particolare dell'algoritmo *Metropolis-Hastings*, usato nel caso multivariato, che consente, tramite un'adeguata funzione di proposta $P(\boldsymbol{X'} \mid \boldsymbol{X})$, di avere una probabilità di accettazione α pari ad 1.

Data la distribuzione $p(\boldsymbol{X})$, assumiamo che per ogni elemento X_j del vettore $\boldsymbol{X} = (X_1, \ldots, X_p)$ sia nota la distribuzione condizionata $p(\boldsymbol{X_j} \mid \boldsymbol{X}_{(j)})$ dove $\boldsymbol{X}_{(j)}$ indica il vettore di $\boldsymbol{X}$ senza la j-esima componente (Gilks et al., 1995). Indichiamo l'$i-$esimo valore campionato come $\boldsymbol{X}^{(i)} = (X_1^{(i)} \ldots X_p^{(i)})$ e fissiamo arbitrariamente un valore $X_j^{(0)}$ per ogni elemento di $\boldsymbol{X^{(0)}}$. A questo punto per ogni campione $i-$esimo si campiona ogni $X_j^{(i)}$ dalla sua distribuzione condizionata ai più recenti

valori disponibili del vettore $\boldsymbol{X}_{(j)}^{(i)}$. Il campione estratto in questo modo riesce ad approssimare la distribuzione congiunta e, esaminando i vettori di valori estratti X_j, è possibile ottenere un approssimazione della distribuzione marginale.

Il campionamento di Gibbs, quindi, risulta particolarmente adatto nel contesto dell'inferenza bayesiana, in cui la distribuzione a-posteriori è specificata come una distribuzione condizionata. È importante specificare opportunamente i punti iniziali dell'algoritmo, stimandoli o determinandoli casualmente all'interno di un range valido. Poiché sono necessarie delle iterazioni di "riscaldamento" per arrivare alla stazionarietà nei valori della distribuzione congiunta, si elimina un parte iniziale di osservazioni (periodo di *burn-in*), al fine di ottenere una migliore approssimazione.

3.3 Modello a 2 stati

Per cercare di capire quali siano i fattori di rischio degli infortuni dei giocatori e per prevedere eventuali stop fisici nell'arco del tempo si è, inizialmente, andati a considerare il processo come una catena di Markov non-omogenea con 2 possibili stati (0 = non infortunato, 1 = infortunato), dove, per le prime $T_j - 1$ osservazioni, il giocatore $G_{j,t}$ risulta essere nello stato "0", mentre al tempo T_j si trova nello stato "1".
In realtà, in questa formulazione iniziale, la probabilità di passare di stato non viene influenzata da fattori esterni sotto forma di covariate, ma solo dalla lunghezza della serie stessa.
Si avranno, quindi, per ogni giocatore una sequenza di allenamenti e partite dove il soggetto non si è infortunato che si conclude con il giorno dell'infortunio.
Questo processo si traduce nelle seguenti condizioni iniziali:

$$\begin{aligned} p(G_{j,t} = 0 | G_{j,t-1} = 0) &= 1 - \pi_{j,t}^{1|0} \\ p(G_{j,t} = 1 | G_{j,t-1} = 0) &= \pi_{j,t}^{1|0} \\ p(G_{j,t} = 1 | G_{j,t-1} = 1) &= 1 \\ p(G_{j,t} = 0 | G_{j,t-1} = 1) &= 0, \end{aligned}$$

con $j = 1, \ldots, J$, identificativo del giocatore, e $t = 1, \ldots, T_j$.
La catena di Markov assume, dunque, la seguente forma:

$$G_{j,t} \sim \mathsf{MC}(\Pi), \qquad \Pi = \begin{bmatrix} 1 - \pi_{j,t}^{1|0} & \pi_{j,t}^{1|0} \\ 0 & 1 \end{bmatrix}.$$

Assumendo di conoscere $G_{j,0}$=0 e posto $S_{j,t}^{1|0} = I(G_{j,t} = 1 | G_{j,t-1} = 0)$, $S_{j,t}^{0|0} = 1 - S_{j,t}^{1|0}$ $= I(G_{j,t} = 0 | G_{j,t-1} = 0)$, dove $I()$ è la funzione indicatrice, la verosimiglianza per

il calciatore j risulta essere:

$$\begin{aligned} L(G_{j,t}, \pi_j) = p(G_{j,1}, \ldots, G_{j,T_j}) &= \prod_{t=1}^{T_j} p(G_{j,t} \mid G_{j,t-1}) \\ &= \prod_{t=1}^{T_j} (\pi_{j,t}^{1|0})^{S_{j,t}^{1|0}} (1 - \pi_{j,t}^{1|0})^{S_{j,t}^{0|0}} \\ &= \prod_{t=1}^{T_j} {\pi_j}^{S_{j,t}^{1|0}} (1 - \pi_j)^{S_{j,t}^{0|0}}, \end{aligned}$$

dove si è posto $\pi_{j,t}^{1|0} = \pi_j$ per semplicità di notazione.
Assumendo, inoltre, una distribuzione a-priori Dirichlet (α_0, β_0) per il parametro π_j, con forma:

$$\pi(\pi_j) = \frac{\Gamma(\alpha_0 + \beta_0)}{\Gamma(\alpha_0)\Gamma(\beta_0)} \pi_j^{\alpha_0 - 1} (1 - \pi_j)^{\beta_0 - 1},$$

la distribuzione a-posteriori per il parametro π_j risulta, quindi, essere:

$$\begin{aligned} \pi(\pi_j \mid G_{j,t}) &\propto L(G_{j,t}, \pi_j)\pi(\pi_j) \\ &\propto \prod_{t=1}^{T_j} {\pi_j}^{S_{j,t}^{1|0}} (1 - \pi_j)^{S_{j,t}^{0|0}} \pi_j^{\alpha_0 - 1} (1 - \pi_j)^{\beta_0 - 1} \\ &\propto {\pi_j}^{\sum_{t=1}^{T_j} S_{j,t}^{1|0} + \alpha_0 - 1} (1 - \pi_j)^{\sum_{t=1}^{T_j} S_{j,t}^{0|0} + \beta_0 - 1}, \end{aligned}$$

che non è altro che il nucleo di una distribuzione Dirichlet $(\alpha_0 + \sum_{t=1}^{T_j} S_{j,t}^{1|0}\ ,\ \beta_0 + \sum_{t=1}^{T_j} S_{j,t}^{0|0})$.

3.4 Modello con covariate

Per poter capire quali siano i fattori di rischio di un infortunio di un giocatore di calcio, si sono inserite nel modello alcune variabili prese dal Dataset Sintetico Finale. Per fare ciò, però, non si può più utilizzare la distribuzione di Dirichlet come a-priori in quanto, nonostante si ottenga a-posteriori una distribuzione con forma chiusa, non permette l'inserimento delle covariate nel modello.

3.4.1 Introduzione alla Polya-Gamma

L'inferenza bayaesiana per il modello logistico è stata a lungo un problema a causa della funzione di verosimiglianza del modello, poiché non è facile trovare una a-priori per i parametri che consenta di ottenere una forma chiusa. A questo scopo Polson et al. (2013) usano una tecnica di *data augmentation*, costruendo una nuova famiglia di distribuzioni, chiamata Polya-Gamma, che nasce dall'identità integrale

$$\frac{(\exp\{\psi\})^a}{(1+\exp\{\psi\})^b} = 2^{-b} e^{k\psi} \int_0^\infty e^{-\omega\psi^2/2} p(\omega)\, d\omega \quad \text{con} \quad k = a - b/2. \tag{3.6}$$

Ricordando che il modello logistico è specificato come segue:

$$\begin{aligned} Y_i &\sim \mathsf{Be}(\pi_i) \\ \pi_i &= \frac{e^{\psi_i}}{1+e^{\psi_i}} \\ \psi_i &= \mathbf{x}_i^\top \boldsymbol{\beta}, \end{aligned} \tag{3.7}$$

con una a-priori per $\boldsymbol{\beta}$ gaussiana, $\boldsymbol{\beta} \sim \mathsf{N}(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$, aumentando il dataset con la variabile latente Polya-Gamma $\omega_i \sim \mathsf{PG}(1,0)$, $i = 1, ..., n$ la distribuzione condizionata del vettore $\boldsymbol{\beta}$ è una Normale multivariata. La funzione di verosimiglianza aumentata è:

$$
\begin{aligned}
L\left(\boldsymbol{\beta} \mid \mathbf{y}, \boldsymbol{\omega}, \mathbf{X}\right) &= \prod_{i=1}^{n} p\left(Y_i = y_i \mid \mathbf{x}_i, \omega_i, \boldsymbol{\beta}\right) \pi\left(\omega_i\right) \\
&= \prod_{i=1}^{n} \left(\frac{\exp\{\mathbf{x}_i^{\mathsf{T}}\boldsymbol{\beta}\}}{1+\exp\{\mathbf{x}_i^{\mathsf{T}}\boldsymbol{\beta}\}}\right)^{y_i} \left(1 - \frac{\exp\{\mathbf{x}_i^{\mathsf{T}}\boldsymbol{\beta}\}}{1+\exp\{\mathbf{x}_i^{\mathsf{T}}\boldsymbol{\beta}\}}\right)^{1-y_i} \pi\left(\omega_i\right) \\
&= \prod_{i=1}^{n} \frac{\left(\exp\{\mathbf{x}_i^{\mathsf{T}}\boldsymbol{\beta}\}\right)^{y_i}}{1+\exp\{\mathbf{x}_i^{\mathsf{T}}\boldsymbol{\beta}\}} \pi\left(\omega_i\right) \\
&= \prod_{i=1}^{n} \frac{1}{2} \exp\left\{y_i\left(\mathbf{x}_i^{\mathsf{T}}\boldsymbol{\beta}\right) - \frac{\mathbf{x}_i^{\mathsf{T}}\boldsymbol{\beta}}{2}\right\} \\
&\qquad \times \exp\left\{-\frac{\omega_i\left(\mathbf{x}_i^{\mathsf{T}}\boldsymbol{\beta}\right)^2}{2}\right\} \pi\left(\omega_i\right) \\
&= \frac{1}{2^n} \exp\left\{\sum_{i=1}^{n} \tilde{y}_i\left(\mathbf{x}_i^{\mathsf{T}}\boldsymbol{\beta}\right)\right\} \\
&\qquad \times \exp\left\{-\sum_{i=1}^{n} \frac{\omega_i\left(\mathbf{x}_i^{\mathsf{T}}\boldsymbol{\beta}\right)^2}{2}\right\} \pi\left(\omega_i\right), \qquad (3.8)
\end{aligned}
$$

dove $\tilde{y}_i = y_i - \frac{1}{2}$, $\mathbf{X} = \begin{bmatrix}\mathbf{1} & \mathbf{x}_1^{\mathsf{T}} & \dots & \mathbf{x}_p^{\mathsf{T}}\end{bmatrix}$ e $\boldsymbol{\omega} = (\omega_1, \dots, \omega_n)$. Dunque, la distribuzione condizionata di $\boldsymbol{\beta}$ risulta essere:

$$
\begin{aligned}
\pi(\boldsymbol{\beta} \mid \mathbf{y}, \mathbf{X}, \boldsymbol{\omega}) &\propto \exp\left\{\sum_{i=1}^{n} \tilde{y}_i\left(\mathbf{x}_i\boldsymbol{\beta}\right)\right\} \exp\left\{-\sum_{i=1}^{n} \frac{\omega_i\left(\mathbf{x}_i\boldsymbol{\beta}\right)^2}{2}\right\} \pi(\boldsymbol{\beta}) \\
&\propto \exp\left\{-\frac{1}{2}\sum_{i=1}^{n} \omega_i\left(\mathbf{x}_i\boldsymbol{\beta} - \frac{\tilde{y}_i}{\omega_i}\right)^2\right\} \pi(\boldsymbol{\beta}) \\
&\propto \exp\left\{-\frac{1}{2}(\tilde{\tilde{\mathbf{y}}} - \mathbf{X}\beta)^{\mathsf{T}}\mathbf{W}(\tilde{\tilde{\mathbf{y}}} - \mathbf{X}\beta)\right\} \pi(\boldsymbol{\beta}) \qquad (3.9)
\end{aligned}
$$

dove $\tilde{\tilde{\mathbf{y}}} = \left(\frac{\tilde{y}_1}{\omega_1}, \dots, \frac{\tilde{y}_n}{\omega_n}\right)$, $\mathbf{W} = \operatorname{diag}(\omega_1, \dots, \omega_n)$. L'equazione (3.9) è proporzionale al nucleo di una Normale multivariata, di parametri $\boldsymbol{\mu}^*$ e $\boldsymbol{\Sigma}^*$, con

$$
\boldsymbol{\mu}^* = \boldsymbol{\Sigma}^*\left(\mathbf{X}^{\mathsf{T}}\mathbf{W}\tilde{\tilde{\mathbf{y}}} + \boldsymbol{\Sigma}_0^{-1}\boldsymbol{\mu}_0\right) \qquad (3.10)
$$

$$
\boldsymbol{\Sigma}^* = \left(\mathbf{X}^{\mathsf{T}}\mathbf{W}\mathbf{X} + \boldsymbol{\Sigma}_0^{-1}\right)^{-1}, \qquad (3.11)
$$

dove $\boldsymbol{\mu}_0$ e $\boldsymbol{\Sigma}_0$ sono appunto rispettivamente il vettore delle medie e la matrice di

covarianza della distribuzione a-priori di $\boldsymbol{\beta}$. Inoltre, la distribuzione condizionata di $\boldsymbol{\omega}$ segue a sua volta una densità Polya-Gamma, $\pi(\boldsymbol{\omega}_i \mid \mathbf{y}, \mathbf{X}, \boldsymbol{\beta}) \sim \mathsf{PG}(1, \mathbf{x}_i^\mathsf{T}\boldsymbol{\beta})$ (si veda Polson et al., 2013). Dunque, con l'introduzione della variabile latente $\boldsymbol{\omega}$ si è in possesso di una nuova tecnica bayesiana per lo studio di dati dicotomici che permette l'uso del *Gibbs Sampling*, in quanto le distribuzioni condizionate delle variabili $\boldsymbol{\beta}$ e $\boldsymbol{\omega}$ sono note in forma chiusa.

3.4.2 Applicazione al modello

Nel seguito, dunque, si sfrutterà il risultato appena mostrato per poter inserire nel modello delle covariate.
Assumendo, quindi, di conoscere $G_{j,0}$=0 e posto $S_{j,t}^{1|0} = I(G_{j,t} = 1|G_{j,t-1} = 0)$, $S_{j,t}^{0|0}$ $= 1 - S_{j,t}^{1|0} = I(G_{j,t} = 0|G_{j,t-1} = 0)$, la verosimiglianza, aumentando il dataset con la variabile latente Polya-Gamma $\boldsymbol{\omega}$, per il calciatore j risulta essere:

$$
\begin{aligned}
L(\boldsymbol{G_{j,t}}, \boldsymbol{\pi_j}) &= \prod_{t=1}^{T_j} (\pi_{j,t}^{1|0})^{S_{j,t}^{1|0}} (1-\pi_{j,t}^{1|0})^{S_{j,t}^{0|0}} \pi(\omega_t) \\
&= \prod_{t=1}^{T_j} (\pi_{j,t}^{1|0})^{S_{j,t}^{1|0}} (1-\pi_{j,t}^{1|0})^{S_{j,t}^{0|0}} \pi(\omega_t) \\
&= \prod_{t=1}^{T_j} \Big(\frac{\exp\{x_t^\top \beta\}}{1+\exp\{x_t^\top \beta\}} \Big)^{S_{j,t}^{1|0}} \Big(1 - \frac{\exp\{x_t^\top \beta\}}{1+\exp\{x_t^\top \beta\}} \Big)^{S_{j,t}^{0|0}} \pi(\omega_t) \\
&= \prod_{t=1}^{T_j} \Big(\frac{\exp\{x_t^\top \beta\}}{1+\exp\{x_t^\top \beta\}} \Big)^{S_{j,t}^{1|0}} \Big(\frac{1+\exp\{x_t^\top \beta\} - \exp\{x_t^\top \beta\}}{1+\exp\{x_t^\top \beta\}} \Big)^{S_{j,t}^{0|0}} \pi(\omega_t) \\
&= \prod_{t=1}^{T_j} \frac{(\exp\{x_t^\top \beta\})^{S_{j,t}^{1|0}}}{(1+\exp\{x_t^\top \beta\})^{S_{j,t}^{1|0}+S_{j,t}^{0|0}}} \pi(\omega_t) \\
&= \prod_{t=1}^{T_j} \Big(\frac{1}{2} \Big)^{S_{j,t}^{0|0}+S_{j,t}^{1|0}} \exp\Big\{ \Big(S_{j,t}^{1|0} - \frac{S_{j,t}^{0|0}+S_{j,t}^{1|0}}{2} \Big) x_t^\top \beta \Big\} \\
&\qquad \times \exp\Big\{ -\frac{\omega_t (x_t^\top \beta)^2}{2} \Big\} \pi(\omega_t) \\
&= \prod_{t=1}^{T_j} \Big(\frac{1}{2} \Big)^{S_{j,t}^{0|0}+S_{j,t}^{1|0}} \exp\Big\{ \Big(\frac{-S_{j,t}^{0|0}+S_{j,t}^{1|0}}{2} \Big) x_t^\top \beta \Big\} \\
&\qquad \times \exp\Big\{ -\frac{\omega_t (x_t^\top \beta)^2}{2} \Big\} \pi(\omega_t) \\
&= \prod_{t=1}^{T_j} \Big(\frac{1}{2} \Big)^{S_{j,t}^{0|0}+S_{j,t}^{1|0}} \exp\Big\{ \Big(\frac{-1+S_{j,t}^{1|0}+S_{j,t}^{1|0}}{2} \Big) x_t^\top \beta \Big\} \\
&\qquad \times \exp\Big\{ -\frac{\omega_t (x_t^\top \beta)^2}{2} \Big\} \pi(\omega_t) \\
&= \prod_{t=1}^{T_j} \Big(\frac{1}{2} \Big)^{S_{j,t}^{0|0}+S_{j,t}^{1|0}} \exp\Big\{ \tilde{y}_{j,t} x_t^\top \beta \Big\} \exp\Big\{ -\frac{\omega_t (x_t^\top \beta)^2}{2} \Big\} \pi(\omega_t),
\end{aligned}
$$

con $j = 1, \ldots, J$, identificativo del giocatore, e $t = 1, \ldots, T_j$.
Assumendo, inoltre, per $\boldsymbol{\beta}$ una distribuzione a-priori Normale $(\boldsymbol{\mu_0}, \boldsymbol{\Sigma_0})$ e posto $\boldsymbol{\tilde{y}_{j,t}}$ $= S_{j,t}^{1|0} - \frac{1}{2}$, allora la distribuzione a-posteriori per $\boldsymbol{\beta}$ risulta essere:

$$\begin{aligned}\pi(\boldsymbol{\beta} \mid \boldsymbol{X}, \boldsymbol{G_{j,t}}, \boldsymbol{\omega_t}) &\propto \exp\Big\{ \sum_{t=1}^{T_j} \tilde{y}_{j,t}(x_t^\top \beta)\Big\} \exp\Big\{ -\sum_{t=1}^{T_j} \frac{\omega_t (x_t^\top \beta)^2}{2}\Big\}\pi(\beta) \\ &\propto \exp\Big\{ -\frac{1}{2}\sum_{t=1}^{T_j} \omega_t \Big(x_t^\top \beta - \frac{\tilde{y}_{j,t}}{\omega_t}\Big)^2\Big\}\pi(\beta) \\ &\propto \exp\Big\{ -\frac{1}{2}(\tilde{\tilde{y}} - X\beta)^\top W(\tilde{\tilde{y}} - X\beta)\Big\}\pi(\beta),\end{aligned}$$

con $\tilde{\tilde{\boldsymbol{y}}} = \left(\frac{\tilde{y}_{j,1}}{\omega_1}, \ldots, \frac{\tilde{y}_{j,T_j}}{\omega_{T_j}}\right)$ e $\boldsymbol{W} = \mathrm{diag}(\omega_1, \ldots, \omega_{T_j})$.

L'equazione risulta essere proporzionale al nucleo di una Normale multivariata di parametri $\boldsymbol{\mu^*}$ e $\boldsymbol{\Sigma^*}$, con:

$$\begin{aligned}\mu^* &= \Sigma^*(X^\top W\tilde{\tilde{y}} + \Sigma_0^{-1}\mu_0) \\ \Sigma^* &= (X^\top WX + \Sigma_0^{-1})^{-1}.\end{aligned}$$

3.5 Modello con covariate per tutti i giocatori

Il risultato precedente è facilmente estendibile al caso in cui si considerano tutti i giocatori insieme.
Assumendoli indipendenti tra loro, la verosimiglianza congiunta, imponendo le stesse assunzioni del caso con un singolo giocatore, diventa:

$$
\begin{aligned}
L(\boldsymbol{G_{j,t}}, \boldsymbol{\pi_j}) &= \prod_{j=1}^{J}\prod_{t=1}^{T_j}(\pi_{j,t}^{1|0})^{S_{j,t}^{1|0}}(1-\pi_{j,t}^{1|0})^{S_{j,t}^{0|0}}\pi(\omega_{j,t}) \\
&= \prod_{j=1}^{J}\prod_{t=1}^{T_j}\pi_j{}^{S_{j,t}^{1|0}}(1-\pi_j)^{S_{j,t}^{0|0}}\pi(\omega_{j,t}) \\
&= \prod_{j=1}^{J}\prod_{t=1}^{T_j}\Big(\frac{\exp\{x_{j,t}^\top\beta\}}{1+\exp\{x_{j,t}^\top\beta\}}\Big)^{S_{j,t}^{1|0}}\Big(1-\frac{\exp\{x_{j,t}^\top\beta\}}{1+\exp\{x_{j,t}^\top\beta\}}\Big)^{S_{j,t}^{0|0}}\pi(\omega_{j,t}) \\
&= \prod_{j=1}^{J}\prod_{t=1}^{T_j}\Big(\frac{\exp\{x_{j,t}^\top\beta\}}{1+\exp\{x_{j,t}^\top\beta\}}\Big)^{S_{j,t}^{1|0}}\Big(\frac{1+\exp\{x_{j,t}^\top\beta\}-\exp\{x_{j,t}^\top\beta\}}{1+\exp\{x_{j,t}^\top\beta\}}\Big)^{S_{j,t}^{0|0}}\pi(\omega_{j,t}) \\
&= \prod_{j=1}^{J}\prod_{t=1}^{T_j}\frac{(\exp\{x_{j,t}^\top\beta\})^{S_{j,t}^{1|0}}}{(1+\exp\{x_{j,t}^\top\beta\})^{S_{j,t}^{1|0}+S_{j,t}^{0|0}}}\pi(\omega_{j,t}) \\
&= \prod_{j=1}^{J}\prod_{t=1}^{T_j}\Big(\frac{1}{2}\Big)^{S_{j,t}^{0|0}+S_{j,t}^{1|0}}\exp\Big\{\Big(S_{j,t}^{1|0}-\frac{S_{j,t}^{0|0}+S_{j,t}^{1|0}}{2}\Big)x_{j,t}^\top\beta\Big\}\exp\Big\{-\frac{\omega_{j,t}(x_{j,t}^\top\beta)^2}{2}\Big\}\pi(\omega_{j,t}) \\
&= \prod_{j=1}^{J}\prod_{t=1}^{T_j}\Big(\frac{1}{2}\Big)^{S_{j,t}^{0|0}+S_{j,t}^{1|0}}\exp\Big\{\Big(\frac{-S_{j,t}^{0|0}+S_{j,t}^{1|0}}{2}\Big)x_{j,t}^\top\beta\Big\}\exp\Big\{-\frac{\omega_{j,t}(x_{j,t}^\top\beta)^2}{2}\Big\}\pi(\omega_{j,t}) \\
&= \prod_{j=1}^{J}\prod_{t=1}^{T_j}\Big(\frac{1}{2}\Big)^{S_{j,t}^{0|0}+S_{j,t}^{1|0}}\exp\Big\{\tilde{y}_{j,t}x_t^\top\beta\Big\}\exp\Big\{-\frac{\omega_{j,t}(x_t^\top\beta)^2}{2}\Big\}\pi(\omega_{j,t}),
\end{aligned}
$$

con $j = 1, \ldots, J$, identificativo del giocatore, e $t = 1, \ldots, T_j$.
A priori $\boldsymbol{\beta}$ si assume con una distribuzione Normale $(\boldsymbol{\mu_0}, \boldsymbol{\Sigma_0})$, posto, inoltre, $\boldsymbol{\tilde{y}_{j,t}}$ $= S_{j,t}^{1|0} - \frac{1}{2}$, allora la distribuzione a-posteriori risulta essere:

$$\pi(\boldsymbol{\beta} \mid \boldsymbol{X}, \boldsymbol{G_{j,t}}, \boldsymbol{\omega_{j,t}}) \propto \exp\Big\{ \sum_{j=1}^{J} \sum_{t=1}^{T_j} \tilde{y_{j,t}} (x_{j,t}^{\top} \beta) \Big\} \exp\Big\{ - \sum_{j=1}^{J} \sum_{t=1}^{T_j} \frac{\omega_{j,t} (x_{j,t}^{\top} \beta)^2}{2} \Big\} \pi(\beta)$$

$$\propto \exp\Big\{ - \frac{1}{2} \sum_{j=1}^{J} \sum_{t=1}^{T_j} \omega_{j,t} (x_{j,t}^{\top} \beta - \frac{\tilde{y_{j,t}}}{\omega_{j,t}})^2 \Big\} \pi(\beta)$$

$$\propto \exp\Big\{ - \frac{1}{2} (\tilde{\tilde{y}}_j - X_j \beta)^{\top} W_j (\tilde{\tilde{y}}_j - X_j \beta) \Big\} \pi(\beta),$$

con
$\boldsymbol{\tilde{\tilde{y}_j}} = \left(\frac{\tilde{y}_{1,1}}{\omega_{1,1}}, \ldots, \frac{\tilde{y}_{1,T_1}}{\omega_{1,T_1}}, \ldots, \frac{\tilde{y}_{J,1}}{\omega_{J,1}}, \ldots, \frac{\tilde{y}_{J,T_j}}{\omega_{J,T_j}} \right)$,
$\boldsymbol{W_j} = diag(\omega_{1,1}, \ldots, \omega_{1,T_1}, \ldots, \omega_{J,1}, \ldots, \omega_{J,T_J})$ e
$\boldsymbol{X_J} = [\mathbf{1}_{\sum_{j=1}^{J} T_j}, x_1, \ldots, x_p]$, con le dimensioni delle variabili $\boldsymbol{x_i}$ uguali a un vettore con $\sum_{j=1}^{J} T_j$ righe.

L'equazione risulta essere proporzionale al nucleo di una Normale multivariata di parametri $\boldsymbol{\mu^*}$ e $\boldsymbol{\Sigma^*}$, con:

$$\mu^* = \Sigma^* ((X_j^{\top} W_j \tilde{\tilde{y}}_j) + \Sigma_0^{-1} \mu_0)$$
$$\Sigma^* = ((X_j^{\top} W_j X_j) + \Sigma_0^{-1})^{-1}.$$

Non ci sono, quindi, modifiche rispetto al modello per il singolo giocatore, tranne per le dimensioni delle matrici $\boldsymbol{X}$, $\boldsymbol{W}$ e il vettore $\boldsymbol{y}$ (e di conseguenza $\tilde{y}$ e $\tilde{\tilde{y}}$).
Da notare che i parametri stimati non si riferiscono al singolo giocatore, ma sono comuni a tutti i soggetti presi in esame.

Algoritmo 1 *Gibbs Sampling*: regressione logistica bayesiana

∘ Inizializzazione di $\boldsymbol{\beta}^{(0)}$ simulando dalla distribuzione a-priori $\sim \mathsf{N}(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$

for $b = 1$ to B **do**:

∘ Simulare dalla distribuzione condizionata:

$$\widehat{\omega}_i^{(b)} \mid \mathbf{x}_i, \boldsymbol{\beta} \sim \mathsf{PG}(1, \mathbf{x}_i^{\mathsf{T}} \boldsymbol{\beta}^{(b-1)}) \qquad i = 1, \ldots, n;$$

∘ aggiornamento della distribuzione di $\boldsymbol{\beta}$

$$\boldsymbol{\mu}^* = \boldsymbol{\Sigma}_0 \left(\mathbf{X}^{\mathsf{T}} \mathbf{W} \tilde{\tilde{\mathbf{y}}} + \boldsymbol{\Sigma}_0^{-1} \boldsymbol{\mu}_\theta \right)$$

$$\boldsymbol{\Sigma}^* = \left(\mathbf{X}^{\mathsf{T}} \mathbf{W} \mathbf{X} + \boldsymbol{\Sigma}_0^{-1} \right)^{-1}$$

∘ simulare nuovi valori $\boldsymbol{\beta}^{(b)}$ dalla distribuzione condizionata:

$$\boldsymbol{\beta} \mid \mathbf{y}, \mathbf{X}, \widehat{\boldsymbol{w}}^{(b)} \sim \mathsf{N}\left(\boldsymbol{\mu}^*, \boldsymbol{\Sigma}^*\right)$$

end for

Capitolo 4

Risultati

Nel seguito si riportano i risultati ottenuti sfruttando i procedimenti visti nella sezione 3.5. Si è applicato sia il *modello logistico*, sia il *modello logistico bayesiano* utilizzando inizialmente le covariate più presenti in letteratura ed in particolare quelle considerate in Rossi et al. (2018). Dopodichè si è modificato il set di variabili. Essendo, però, il numero di infortuni relativamente piccolo, si sono effettuate anche delle simulazioni aumentando il numero di giocatori infortunati. In particolare, si è generato il vettore di 0 ed 1 della variabile risposta utilizzando come covariate le distribuzioni che approssimano le variabili "reali" e come parametri iniziali i valori dei $\hat{\beta}_i$ stimati nel modello con i dati "veri". In questo modo è noto il processo generatore sia della variabile risposta che delle covariate.

4.1 Dati reali

Si è inizialmente implementato un modello logistico con, quindi, funzione legame logit per avere un modello base di riferimento. Come detto, si sono inizialmente inserite come covariate le variabili maggiormente usate in letteratura, ossia *distanza percorsa* in chilometri, *accelerazioni*, *decelerazioni*, *distanza percorsa nella prima zona di velocità* in chilometri, *distanza percorsa nella seconda zona di velocità* in chilometri, *distanza percorsa nella terza zona di velocità* in metri e *distanza percorsa nella seconda zona di potenza* in chilometri.

Come si può vedere in Tabella 4.1, non tutti i coefficienti sono significativi, ma ottengo già delle prime informazioni utili.

	Stima	Errore std.	Valore z	Pr(>|z|)
(Intercetta)	-5.2145	0.6203	-8.407	0.0000
Distanza	-0.1555	0.2730	-0.570	0.5689
Accelerazioni	-0.0286	0.0210	-1.360	0.1738
Decelerazioni	-0.2666	0.0510	-5.224	0.0000
Sp.dist.Z1	0.8307	0.2391	3.474	0.0005
Sp.dist.Z2	1.4032	0.8374	1.676	0.0938
Sp.dist.Z3	-0.0041	0.0094	-0.436	0.6632
Power.dist.Z2	0.0167	0.3033	0.055	0.9560

Tabella 4.1: Stime di massima verosimiglianza dei parametri del modello logistico.

Infatti, si può calcolare sia la stima della probabilità di infortunio ($\hat{\pi} = 0.031$), sia l'*odds ratio* (OR), che ci dà una importante interpretazione sul significato dei coefficienti e, quindi, sull'importanza delle variabili nella possibilità di infortunarsi oppure no. In particolare, denota che se aumento il valore della variabile X_i di una unità, lasciando le altre inalterate, aumento la probabilità di infortunio di $e^{\hat{\beta}_i}$ volte. Se il valore ottenuto è maggiore di 1, quindi, la variabile associata aumenta il rischio di infortunio, se è inferiore il rischio diminuisce. Questi valori, però, vanno contestualizzati, nel senso che a seconda della variabile presa in considerazione, l'incremento di una unità del suo valore ha un significato diverso.
In questa applicazione si è ottenuto che, per i coefficienti significativi nel modello, all'aumentare di una unità del conteggio delle decelerazioni il rischio di infortunio diminuisce (OR = 0.766), mentre se incrementiamo di un chilometro la distanza percorsa nella prima e seconda zona di velocità, aumenta notevolmente la possibilità di infortunarsi (OR = 2.295 e 4.07 rispettivamente). Da notare anche che, in questo modello, se le covariate assumessero valore nullo la stima della probabilità di infortunio sarebbe di 0.0067.

L'applicazione della regressione logistica bayesiana, sfruttando l'algoritmo Gibbs Sampling (1) sui giocatori che si sono realmente infortunati nel periodo di interesse dell'analisi, ha portato a dei risultati a-posteriori simili a quelli della regressione logistica visto che la distribuzione a-priori per β è stata impostata come Normale multivariata con media nulla e matrice di varianza-covarianza diagonale con valori molto grandi. In questo modo, si è data ampia libertà ai parametri di variare se-

guendo i dati in possesso senza imporre una distribuzione a-priori eccessivamente vincolante. Per verificare che i $\hat{\beta}_i$ ottenuti a-posteriori convergessero ad un unico valore, si sono avviate simultaneamente più catene togliendo una parte iniziale di burn-in.
In particolare, si è inizialmente applicato il modello al caso con 7 covariate ispirate dalla letteratura, più l'intercetta.
In generale, per la maggior parte delle variabili i valori dei parametri associati si discostano da zero, anche se le deviazioni standard delle stime, confrontate con le medie, sono relativamente grandi (Tabella 4.2). Questo risultato va spiegato dal piccolo numero di osservazioni a disposizione. Ne consegue anche che gli intervalli di credibilità al 95% risultano abbastanza grandi, infatti, solo per il parametro dell'*intercetta*, delle *decelerazioni* e della *distanza percorsa nella prima zona di velocità*, i limiti non comprendono lo 0. Questo significa che per queste variabili non c'è mai un cambio dell'interpretazione visto che il parametro ad esse relativo non cambia di segno nell'intervallo proposto.

	Stima	Errore std.	Intervalli HPD 95%	
			Inferiore	Superiore
(Intercetta)	-5.3764	0.6366	-6.5882	-4.1225
Distanza	-0.1842	0.2821	-0.7347	0.3644
Accelerazioni	-0.0308	0.0215	-0.0741	0.0104
Decelerazioni	-0.2797	0.0532	-0.3808	-0.1776
Sp.dist.Z1	0.8783	0.2429	0.4017	1.3735
Sp.dist.Z2	1.4769	0.8564	-0.1573	3.2042
Sp.dist.Z3	-0.0071	0.0101	-0.0269	0.0122
Power.dist.Z2	0.0148	0.3096	-0.5779	0.6251

Tabella 4.2: Stime a-posteriori dei parametri del modello.

I valori dei parametri nelle 5 catene impostate, tolto il burn-in dei primi 1000 dati generati su 5000, sono oscillati intorno alle rispettive medie a-posteriori e si sono distribuiti approssimativamente come delle distribuzioni Normali (Figura 4.1 e 4.2), come ci si aspettava dalla teoria. Inoltre, anche le diagnostiche proposte da Gelman e Rubin (si veda Gelman (1992)) per verificare la convergenza vengono soddisfatte, con catene che nonostante siano state fatte partire da punti diversi si sono attestate intorno ad un unico valore e una misura del *potential scale reduction factor* sempre inferiore a 1.1.

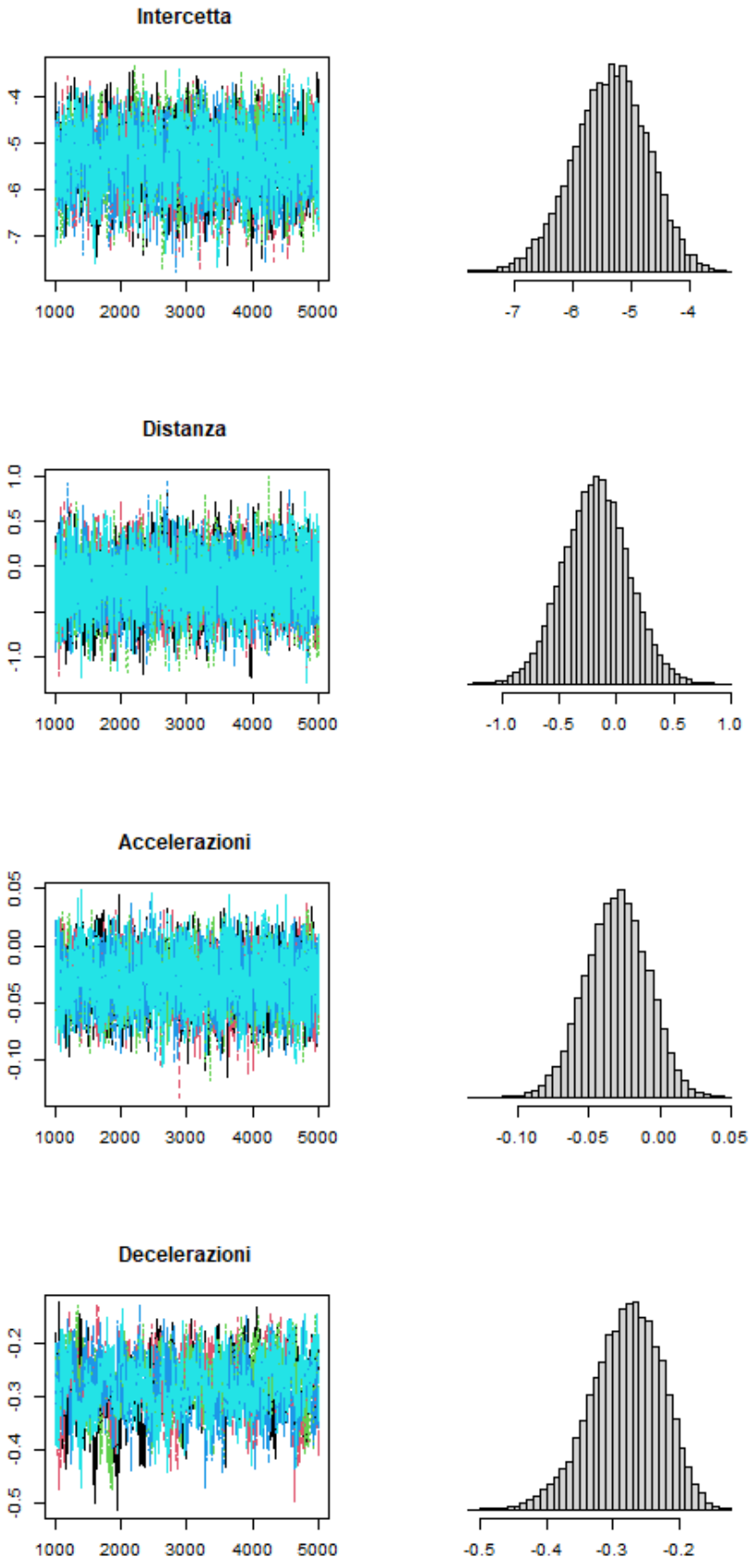

Figura 4.1: Convergenza delle catene e istogrammi dei valori a-posteriori dei parametri.

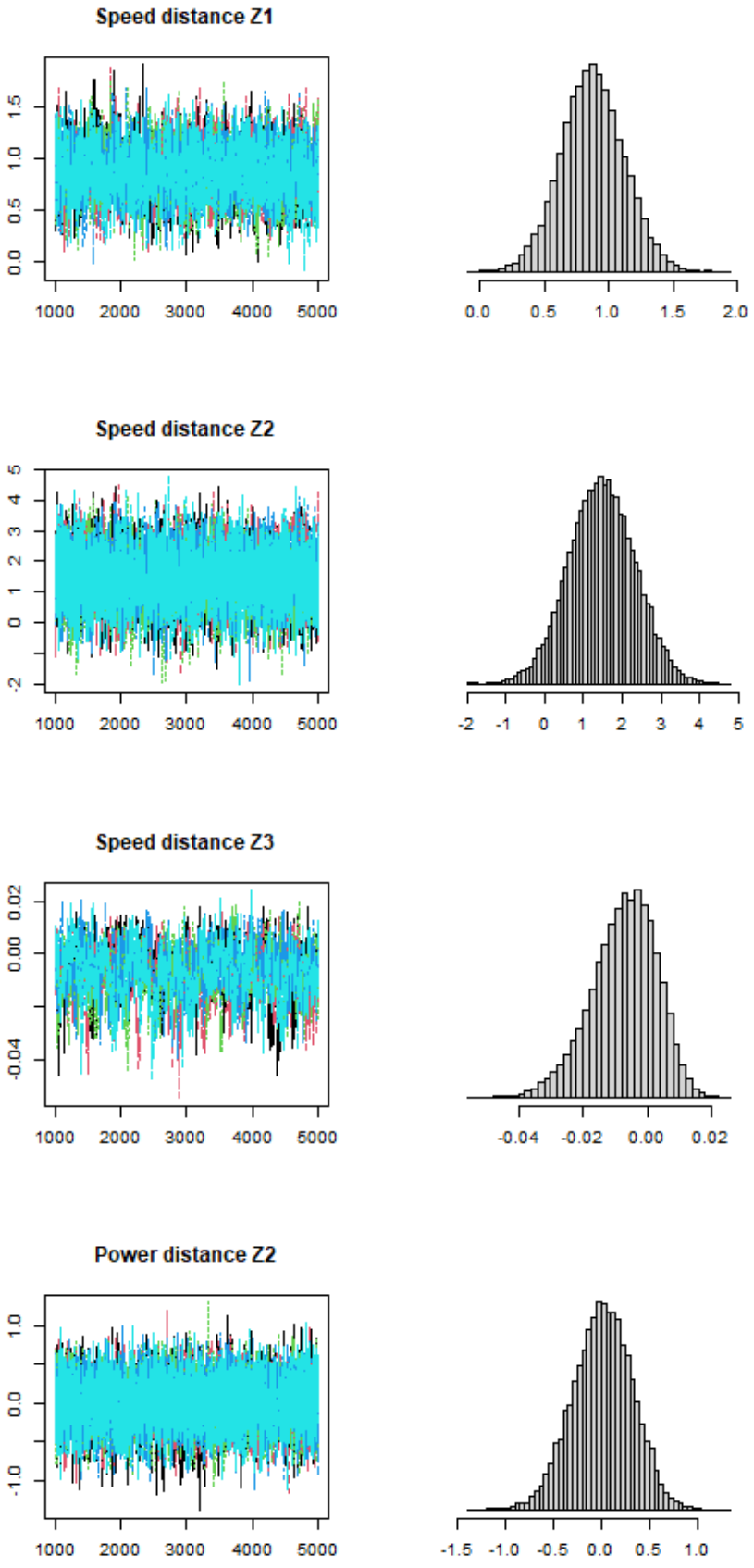

Figura 4.2: Convergenza delle catene e istogrammi dei valori a-posteriori dei parametri.

I valori delle medie a-posteriori, essendo nel caso di una regressione logistica bayesiana, vanno interpretati come nel caso del modello logistico semplice. Per fornire qualche informazione in più sul peso delle covariate, si sono fatti variare i valori di una variabile alla volta lasciando le altre inalterate, calcolando poi le rispettive probabilità di infortunio derivanti. In questo modo, si vuole capire meglio come i valori delle covariate possano influenzare le probabilità degli infortuni dei giocatori.

La stima della probabilità di infortunio è di circa 0.03, ma si è rilevato che facendo oscillare i valori della distanza dal suo minimo al suo massimo, la stima passa da 0.016 a 0.054 circa. Modificando i valori delle accelerazioni, il dato più interessante si ottiene quando si fissa questa variabile col suo massimo dove la probabilità di infortunio diminuisce fino al 0.7%. Per quanto riguarda il numero di decelerazioni, invece, la stima varia di molto se si cambia questo valore tra media, minimo, massimo, primo quartile e terzo quartile. In particolare, se non si effettuano decelerazioni in nessuna delle sessioni considerate, $\hat{\pi}$ raggiunge il valore di 0.21, completamente opposto al 0.0000000008 nel caso in cui effettuo 71 decelerazioni a sessione. Per la distanza percorsa nelle zone di velocità, invece, facendo oscillare le osservazioni della distanza percorsa nella terza zona di velocità non ci sono discostamenti significativi dalla stima di π, mentre nelle prime due zone, al crescere dei chilometri corsi, aumenta notevolmente la probabilità di infortunio con un picco di 0.52 nel caso in cui in tutte le sessioni si corressero circa 2 km a ritmo basso (il massimo per questa variabile), lasciando le altre variabili inalterate. Infine, considerando la distanza percorsa nella seconda zona di potenza, provando vari valori, la stima della probabilità di infortunio rimane praticamente identica.

4.1.1 Modifica del set di covariate

Stimato il modello con 7 covariate più l'intercetta, si è deciso di modificare i regressori. In particolare, avendo a disposizione molte variabili, si è fatta prima un'analisi della correlazione, dopodichè si è implementata una regressione logistica con le variabili scelte e si è fatta *backward selection* per ottenere un set di covariate significative (al 10%). Infine, questo nuovo set di regressori è stato utilizzato nella regressione logistica bayesiana, sfruttando sempre il *Gibbs Sampling* e la possibilità di simulare simultaneamente più catene, impostando le stesse modalità del modello

presentato in precedenza.
Nello specifico, le variabili che sono risultate significative da questa analisi sono:

- distanza percorsa
- numero di decelerazioni
- distanza percorsa nella seconda zona di velocità
- media del VO^2, energia aerobica
- velocità massima raggiunta
- tempo medio dei MPE
- recupero medio di potenza dai MPE

In questo caso, la *backward selection* non ha ritenuto significativa l'intercetta.
I valori a-posteriori dei parametri, riportati in Tabella 4.3, avendo un set di variabili in parte diverso da quello del modello precedente, hanno assunto valori diversi con deviazioni standard in generale più contenute rispetto al caso precedente.

	Stima	Errore std.	Intervalli HPD 95%	
			Inferiore	Superiore
Distanza	1.0377	0.1739	0.6884	1.368
Decelerazioni	-0.1989	0.0484	-0.2877	-0.1068
Sp.dist.Z2	1.423	0.6333	0.2489	2.6699
avg.VO2	-0.7593	0.2282	-1.2067	-0.3214
Vel.max	-0.1677	0.0310	-0.2289	-0.1089
MPE.avg.time	0.1262	0.0692	-0.0140	0.2554
MPE.rec.avg.p	0.5323	0.2151	0.1270	0.9539

Tabella 4.3: Stime a-posteriori dei parametri del modello.

In particolare, i parametri relativi alle decelerazioni e alla distanza percorsa nella seconda zona di velocità assumono dei valori molto simili al modello precedente, quello riferito alla distanza, invece, cambia completamente passando ad essere un fattore che aumenta il rischio di infortunio (OR = 2.8). I parametri delle nuove variabili inserite, due sono negativi, media dell'energia aerobica e velocità massima, due sono positivi, durata media di un MPE e potenza media di recupero da un

MPE. Viste le varianze minori rispetto al modello precedente, anche gli intervalli di credibilità sono più stretti e l'unico parametro che comprende anche lo 0 e che, quindi, cambia di segno da un limite all'altro è quello riferito al *tempo medio di un MPE*.
Analizzando come si comporta la stima della probabilità di infortunio facendo variare i valori di una covariata alla volta, si è trovato che se in tutte le sessioni si corressero 8.5 chilometri come distanza totale, questa stima supererebbe il 50%. Mentre facendo oscillare il numero di decelerazioni tra 0 e 71, ossia il minimo e il massimo di questa variabile, la probabilità di infortunio passerebbe da 0.1 a 0.0000002. La distanza percorsa nella seconda zona di velocità, invece, non influisce molto, se non nel caso in cui si corressero sempre 2 chilometri (valore massimo) a questa andatura, con una stima di 0.14. Al diminuire della quantità di energia aerobica prodotta, la stima cresce fino al 21%, se invece si fissa col suo valore massimo (15.71 W/kg) si ottiene una probabilità del 0.001. La velocità massima raggiunta fa discostare maggiormente la stima di infortunio dal valore osservato nel caso del suo valore minimo, facendola aumentare fino a al 9.6%. Modificando le rilevazioni del tempo medio di un MPE a sessione, invece, i valori della stima superano 0.3 solo per durate superiori ai 10 secondi, mentre se si fissasse il recupero medio di potenza dopo un MPE col suo valore massimo, ossia 13.78 W/kg, la stima della probabilità di infortunio raggiungerebbe 0.46.

Analizzando i risultati delle 5 catene, tolto il burn-in, queste esplorano, per ciascun parametro, i punti intorno alla media a-posteriori, come si può vedere dalle Figure 4.3 e 4.4, e le distribuzioni si possono assolutamente considerare approssimativamente Normali per tutti i $\hat{\beta}_i$. Inoltre, anche le diagnostiche proposte da Gelman e Rubin per verificare la convergenza vengono raggiunte, con in particolare un valore del *potential scale reduction factor* di nuovo sempre inferiore a 1.1.

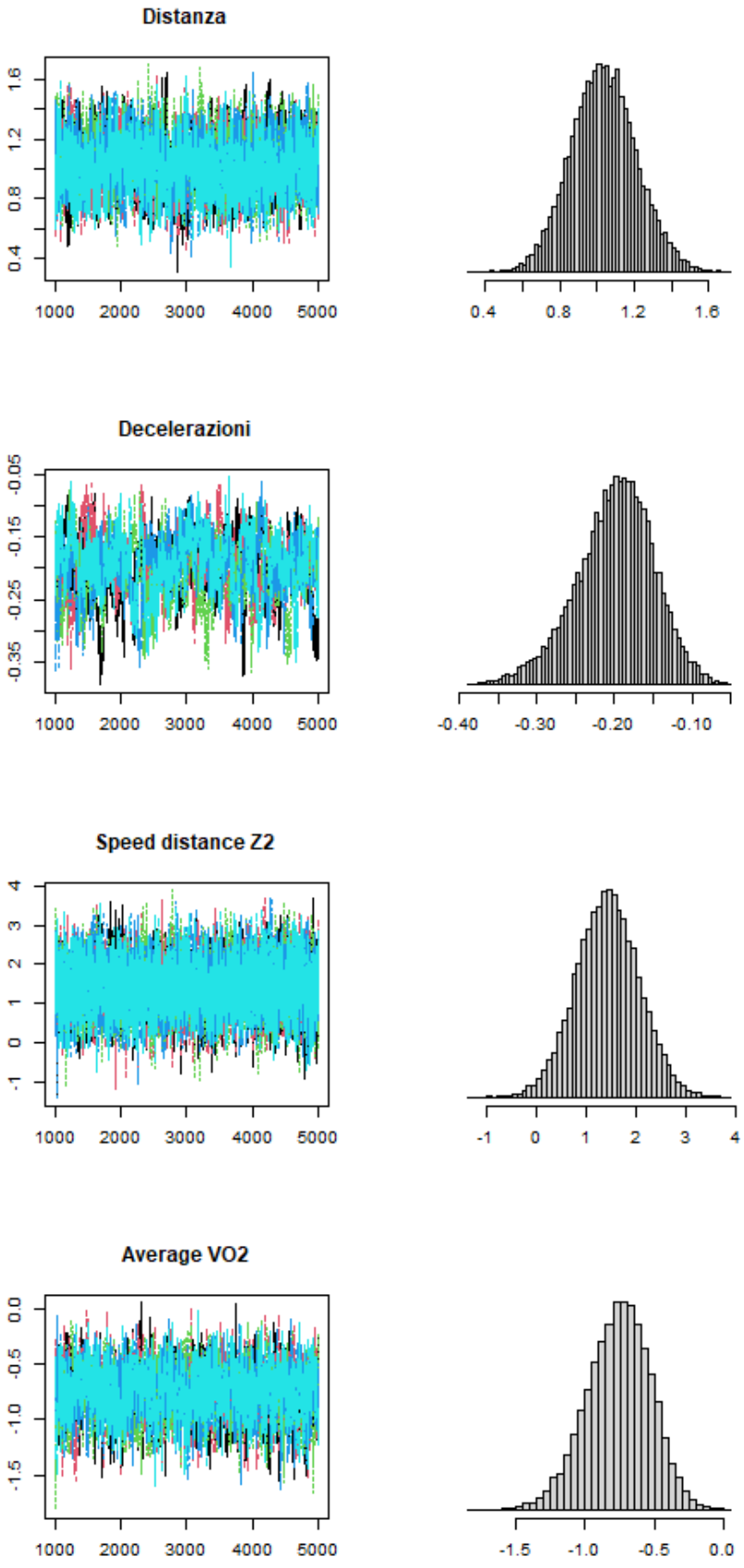

Figura 4.3: Convergenza delle catene e istogrammi dei valori a-posteriori dei parametri.

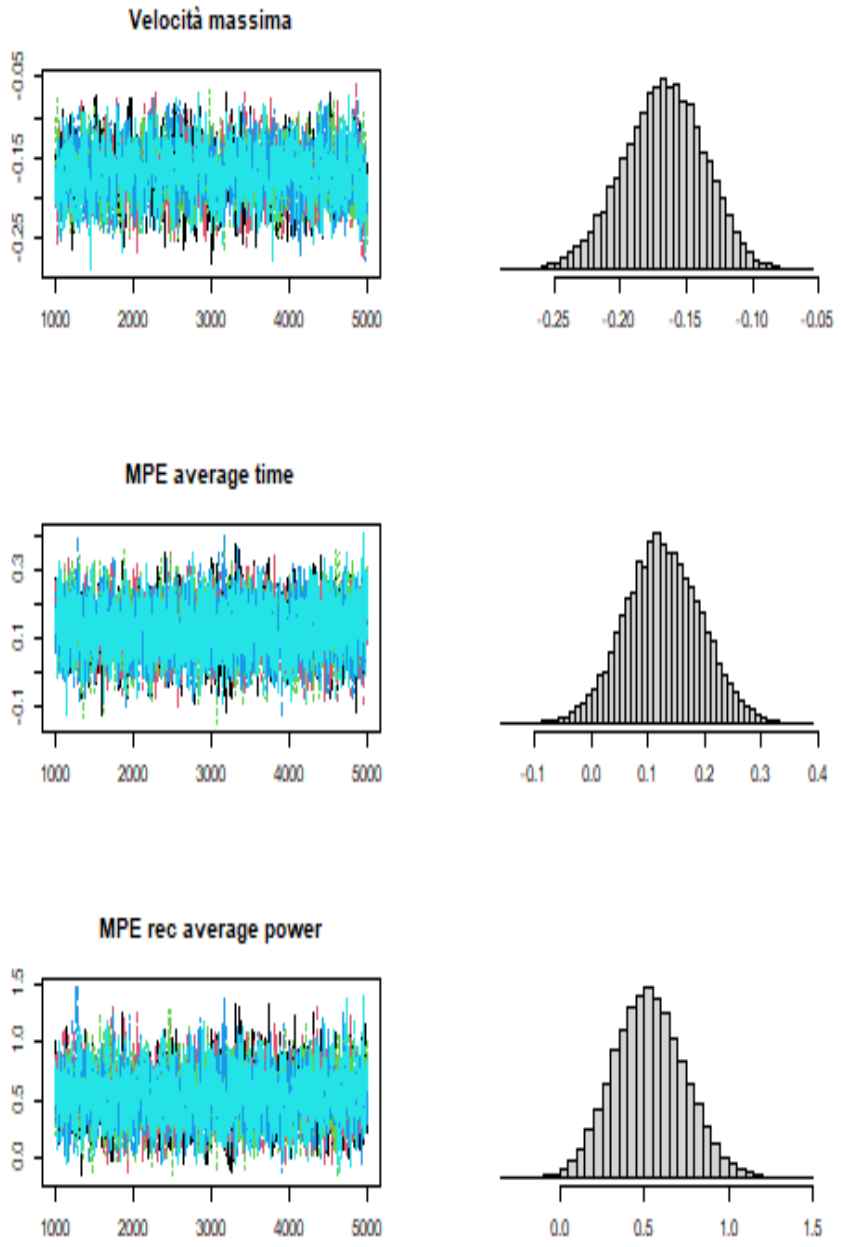

Figura 4.4: Convergenza delle catene e istogrammi dei valori a-posteriori dei parametri.

4.2 Simulazioni

Preso atto della disponibilità di pochi giocatori infortunati si è fatto uso di simulazioni per superare questa limitazione.

Come anticipato in precedenza, si è creata la sequenza di zero ed uno, ossia la variabile risposta, inserendo nel modello logistico una funzione legame logit dove ψ è, in pratica, noto, visto che nella matrice X si sono inseriti i valori delle distribuzioni approssimate delle covariate e i valori dei β di partenza si sono impostati simili a quelli stimati nel modello con i dati reali. In questo modo, si è semplicemente verificato che la sequenza creata fosse sufficientemente lunga per il numero di giocatori impostati ed eventualmente si è leggermente modificato il valore dei β iniziali per ottenere questo risultato. A questo punto, si è stimato il modello come nel caso con i dati reali, con covariate le distribuzioni approssimate delle variabili utilizzate.

Per fare le simulazioni, dunque, si sono approssimate le variabili inserite nei modelli con delle distribuzioni, in modo da poter ricreare il fenomeno generatore dei dati. Per fare ciò, si sono visualizzati graficamente i dati delle singole covariate, comparandoli con delle distribuzioni note.
In particolare, la *distanza* è stata considerata come chilometri percorsi e come si vede dalla Figura 4.5, i valori sembrano distribuirsi come una Normale con media 5.5115 chilometri e standard deviation di 2.26 chilometri.
Discorso analogo per la *distanza percorsa nella prima zona di velocità*, che a livello di distribuzione, considerando tutti i team assieme, si comporta in maniera molto simile ad una Normale con media 4.9 km e standard deviation 2.1 km (Figura 4.5).

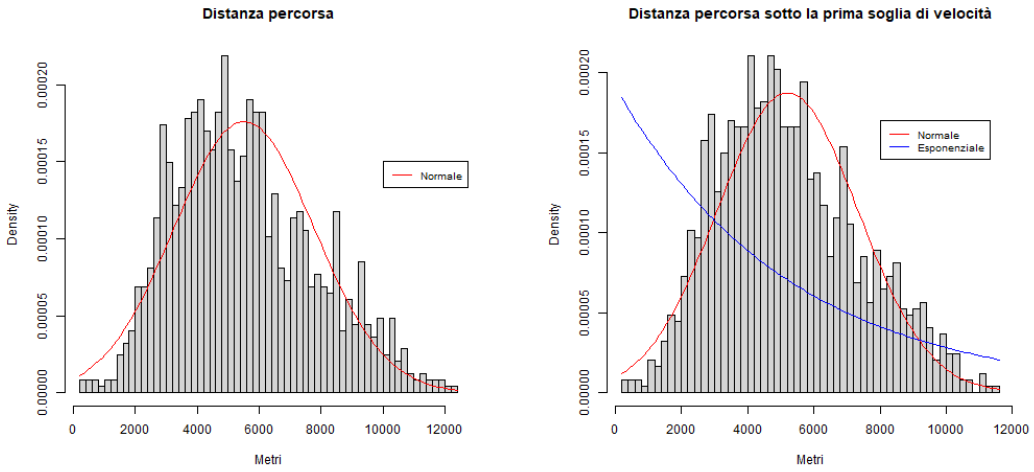

Figura 4.5: Distribuzione della distanza percorsa e della distanza percorsa nella prima zona di velocità.

Le *distanze percorse nella seconda e terza zona di velocità*, all'opposto, come si può vedere in Figura 4.6, si distribuiscono in maniera più simile ad una Esponenziale che ad una Normale. Nello specifico, i valori, analizzando le squadre congiuntamente, danno una distanza media percorsa nella seconda zona di 264 metri.
Nella terza zona, invece, la grande mole di zeri, influenza la media di metri corsi, con un valore di poco superiore a 40 metri.

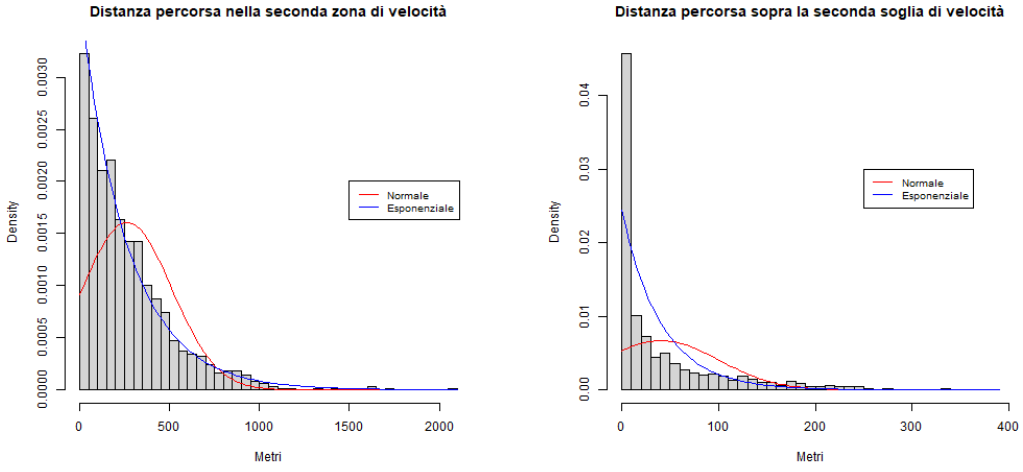

Figura 4.6: Distribuzione della distanza percorsa nella seconda e terza zona di velocità.

La *distanza percorsa nella seconda zona di potenza* (Figura 4.7), non ha un andamento così chiaro come le altre variabili viste fino ad ora. Tuttavia, si è comunque considerato che la distribuzione gaussiana con media 1.1 km e deviazione standard 0.7 km fosse la più adatta.

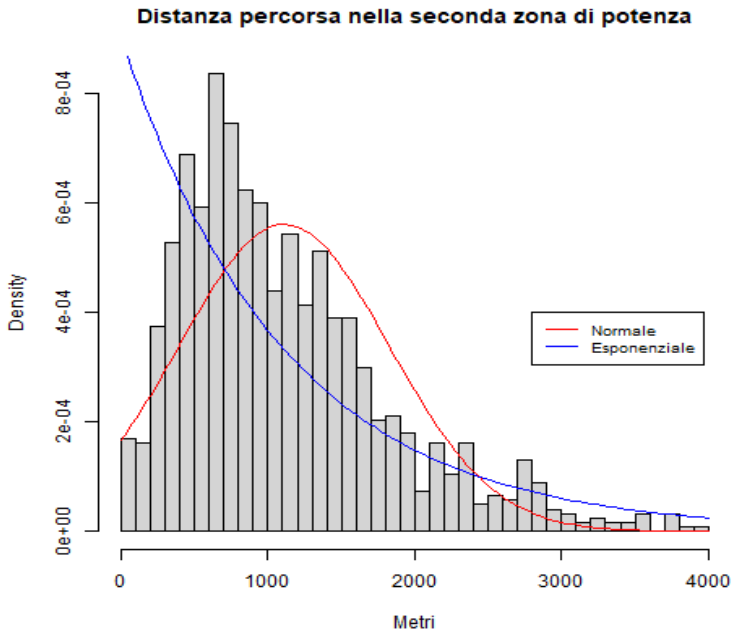

Figura 4.7: Distribuzione della distanza percorsa nella seconda zona di potenza.

Per quanto riguarda il numero di *accelerazioni* e *decelerazioni*, queste sono due variabili di conteggio, ma verificando il loro andamento, come si vede in Figura 4.8, si sono simulate come delle distribuzioni Esponenziali con valori attesi rispettivamente 18 e 15 eventi.

La *potenza metabolica media di recupero da un MPE* e il *tempo medio di un MPE*, sembrano seguire meglio l'andamento di una distribuzione t di Student centrata sulle relative mediane con 15 e 30 gradi di libertà rispettivamente, ma controllando bene le code sarebbero troppo corte (Figura 4.9). Di conseguenza, si è preferito approssimarle anche in questo caso con delle variabili Normali con media rispettivamente 3.7 W/kg e 7 secondi e deviazioni standard 1.5 W/kg e 2.4 secondi.

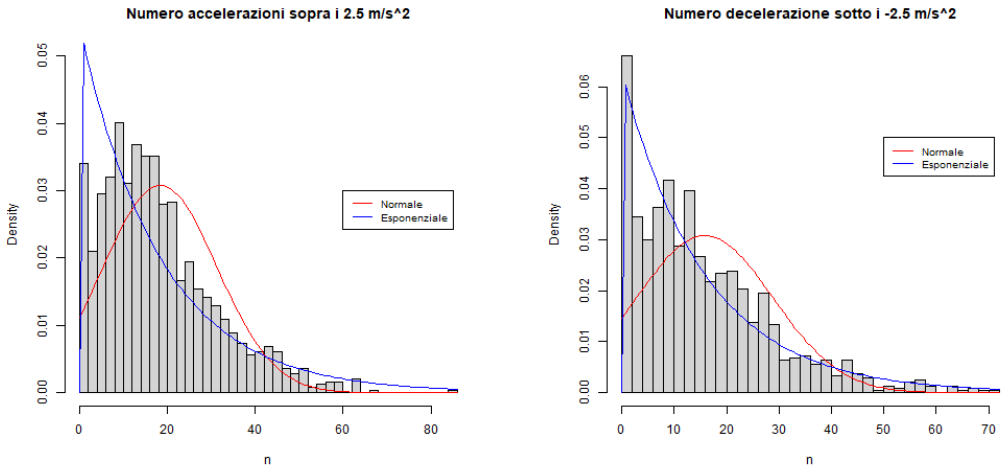

Figura 4.8: Distribuzione delle accelerazioni e decelerazioni.

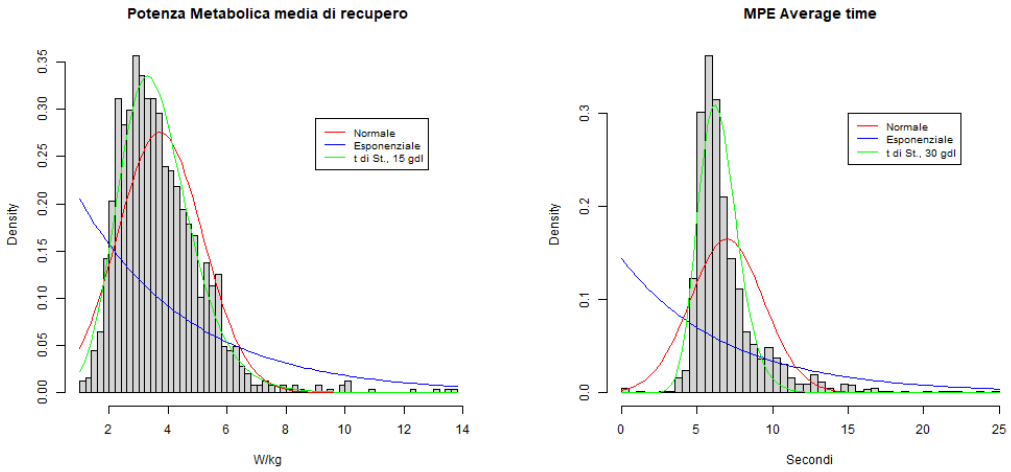

Figura 4.9: Distribuzione delle variabili *MPE.rec.avg.time* e *MPE.avg.time*.

Infine, il consumo di VO^2 *medio* e la *velocità massima* entrambe si distribuiscono approssimativamente come delle variabili Normali con relative medie 6.3 W/kg e 26.5 km/h e standard deviation 2 W/kg e 3.8 km/h (Figura 4.10).

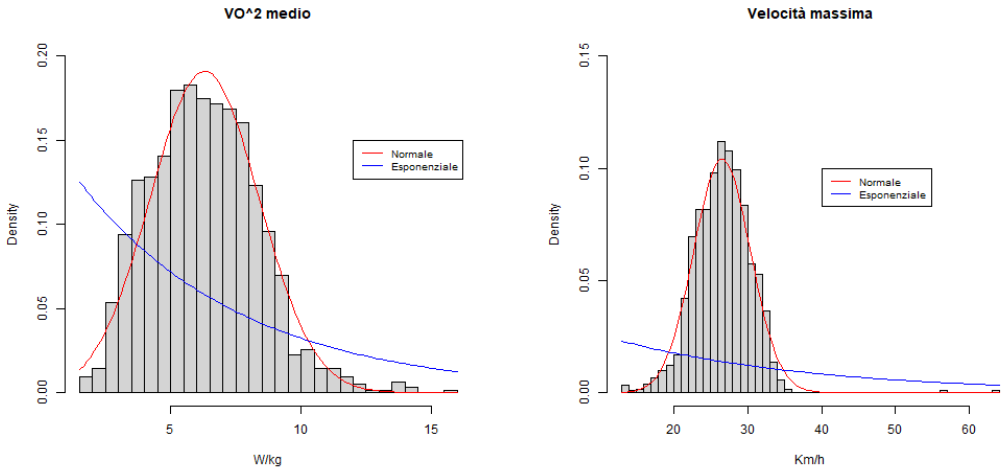

Figura 4.10: Distribuzione del consumo medio di VO^2 e della velocità massima.

Conclusa la panoramica su come si distribuiscono approssimativamente le covariate utilizzate nei modelli implementati, queste sono state utilizzate come variabili nelle simulazioni.
Nello specifico, si sono create due simulazioni per ciascun modello, la prima generando una sequenza di 100 giocatori, la seconda aumentando il numero di giocatori a 500 e si è impostato che in media si infortunino circa al 30-esimo giorno, valore calcolato dai dati reali (Tabella 1.3). Come detto, come covariate si sono utilizzate le distribuzioni a cui le variabili si approssimano e come valori di partenza dei parametri si sono impostati i valori stimati nei modelli con le vere variabili.
Quello che si vuole andare a verificare è se le distribuzioni a-posteriori dei parametri sono ancora Normali, se i valori dei parametri simulati rimangono simili ai valori dei parametri stimati e se i risultati delle simulazioni si concentrano maggiormente intorno al loro valore medio. Per fare ciò, si sono valutati graficamente i risultati ottenuti, tramite istogrammi e boxplot.

Le due simulazioni del modello con intercetta e le 7 covariate ispirate dalla letteratura hanno dato in entrambi casi risultati molto buoni. Infatti, le distribuzioni marginali a-posteriori dei parametri sono gaussiane, il valore del parametro stimato è sempre compreso nelle catene simulate, le nuove medie non sono distanti da questo valore e le deviazioni standard si riducono notevolmente. I valori dei parametri simulati, quindi, si concentrano maggiormente intorno alla loro media (Appendi-

ce B). Questi risultati, indicano che da una parte i valori dei parametri stimati inizialmente sono attendibili, ma dall'altra che l'utilizzo di un numero maggiore di giocatori concentra maggiormente i valori. Anche gli intervalli di credibilità, infatti, si riducono notevolmente di ampiezza. L'unico parametro che comprende il valore nullo nei suoi limiti è per entrambe le simulazioni quello riferito alla *distanza percorsa nella seconda zona di potenza*.
Interessante, in realtà, notare che le stime dei parametri nelle due simulazioni non sono proprio identiche, i $\hat{\beta}_i$ relativi all'intercetta e alla distanza percorsa nella seconda zona di potenza sono relativamente differenti, con in particolare quello relativo alla seconda covariata citata che passa da negativo a positivo (Tabella 4.4 e 4.5).

	Stima_{100}	Errore std._{100}	Intervalli HPD 95%	
			Inferiore	Superiore
(Intercetta)	-4.2825	0.6179	-5.4906	-3.0747
Distanza	-0.2197	0.0556	-0.3275	-0.1091
Accelerazioni	-0.0252	0.0086	-0.0420	-0.0093
Decelerazioni	-0.3237	0.0395	-0.3977	-0.2543
Sp.dist.Z1	0.6929	0.0694	0.5622	0.8262
Sp.dist.Z2	1.090	0.3869	0.3406	1.8598
Sp.dist.Z3	-0.0089	0.0036	-0.01612	-0.0023
Power.dist.Z2	-0.0949	0.1752	-0.4248	0.2561

Tabella 4.4: Medie a-posteriori delle simulazioni del primo modello con 100 giocatori.

	Stima_{500}	Errore std._{500}	Intervalli HPD 95%	
			Inferiore	Superiore
(Intercetta)	-5.6517	0.2869	-6.1929	-5.0706
Distanza	-0.1076	0.0244	-0.1545	-0.0592
Accelerazioni	-0.0300	0.0041	-0.0383	-0.02241
Decelerazioni	-0.2972	0.0157	-0.3284	-0.2670
Sp.dist.Z1	0.7820	0.0317	0.7194	0.8434
Sp.dist.Z2	1.2118	0.1715	0.8685	1.5451
Sp.dist.Z3	-0.0108	0.0017	-0.0141	-0.0076
Power.dist.Z2	0.1423	0.0764	-0.0031	0.2962

Tabella 4.5: Medie a-posteriori delle simulazioni del primo modello con 500 giocatori.

Le simulazioni riferite al secondo modello proposto, senza intercetta e con 7 covariate in parte diverse dal primo, con 100 e 500 giocatori, producono dei risul-

tati delle medie delle stime a-posteriori molto più vicini tra loro, senza differenze sostanziali, come si può vedere in Tabella 4.6 e 4.7. Anche in questo caso, come era prevedibile aumentando la numerosità campionaria, le deviazioni standard diminuiscono al crescere dei giocatori, con conseguente diminuzione dell'ampiezza degli intervalli di credibilità. In queste simulazioni, inoltre, non c'è alcun cambio di segno tra i due limiti proposti.
Le distribuzioni marginali a-posteriori dei parametri sono sempre approssimativamente gaussiane e il valore del parametro stimato è sempre vicino alla media dei valori generati nelle 5 catene (Appendice B).

	Stima_{100}	Errore std._{100}	Intervalli HPD 95%	
			Inferiore	Superiore
Distanza	1.008	0.0916	0.8354	1.1971
Decelerazioni	-0.3043	0.0345	-0.3738	-0.2399
Sp.dist.Z2	1.5544	0.4964	0.5838	2.5188
avg.VO2	-0.8499	0.0910	-1.0370	-0.6828
Vel.max	-0.2829	0.0296	-0.3403	-0.2245
MPE.avg.time	0.1273	0.0602	0.0117	0.2481
MPE.rec.avg.p	0.4360	0.0966	0.2535	0.6309

Tabella 4.6: Medie a-posteriori delle simulazioni del secondo modello con 100 giocatori.

	Stima_{500}	Errore std._{500}	Intervalli HPD 95%	
			Inferiore	Superiore
Distanza	1.007	0.0395	0.9244	1.0848
Decelerazioni	-0.2798	0.0155	-0.3068	-0.2501
Sp.dist.Z2	1.4996	0.2097	1.0982	1.899
avg.VO2	-0.9014	0.0389	-0.9753	-0.8196
Vel.max	-0.2606	0.0128	-0.2857	-0.2355
MPE.avg.time	0.0836	0.0245	0.0369	0.1312
MPE.rec.avg.p	0.4459	0.0430	0.3621	0.5277

Tabella 4.7: Medie a-posteriori delle simulazioni del secondo modello con 500 giocatori.

Conclusioni

In questo elaborato si è proposto un nuovo metodo di analisi dei fattori di rischio degli infortuni dei calciatori di squadre d'elite del campionato italiano di Serie A. In particolare, si è studiata la relazione tra la probabilità di infortunio dei giocatori e alcune variabili scelte dal dataset fornito dall'azienda Exelio. Si è studiata questa relazione sia utilizzando il modello logistico classico basato sulla verosimiglianza, sia tramite l'approccio bayesiano. In quest'ultimo caso, in particolare, sfruttando la tecnica di data augmentation con la variabile casuale Polya-Gamma che permette di ottenere a-posteriori una funzione condizionata in forma chiusa, si è potuto implementare il Gibbs Sampling.

Alla luce dei risultati ottenuti nei modelli implementati, si ritiene che il modello con 7 covariate senza intercetta ottenuto tramite backward selection di un set di variabili più ampio, possa dare delle informazioni interessanti sulla criticità di alcune variabili nel rischio di infortunio degli atleti. Monitorando maggiormente queste rilevazioni, si potrebbe considerare di preparare allenamenti ad-hoc in alcuni momenti della stagione sportiva per scongiurare eventuali stop fisici.

Ci sono, tuttavia, degli importanti aspetti non presi in considerazione in questa tesi. Il periodo a cui fa riferimento questo elaborato è sia limitato, sia inatteso nel normale svolgimento della stagione sportiva. I giocatori presi in considerazione si sono tutti infortunati in questo lasso di tempo e non si è fatta alcuna distinzione tra infortuni muscolari e non o infortuni da contatto e non. La differente squadra di appartenenza dell'atleta non viene considerata come informazione nei modelli. Le osservazioni delle variabili prese in considerazione non vengono pesate temporalmente, c'è solo la distinzione se le rilevazioni sono fatte in una sessione in cui il giocatore si è infortunato oppure no. Non si hanno informazioni su eventuali infortuni pregressi degli atleti che potrebbero indicare una maggiore o minore

attitudine a problemi muscolari.

Considerati questi aspetti, dunque, ci potrebbero essere molti sviluppi futuri in questo ambito per migliorare l'analisi proposta ed eventualmente aiutare a prevedere gli infortuni dei calciatori.

Appendice A

Dataset Sintetico completo

#	Nome variabile	Tipo variabile	Descrizione variabile
Fixed			
1	date.time	carattere	data e ora di accensione del dispositivo
2	category	carattere	tipologia di sessione compiuta, può cambiare di squadra in squadra
3	tags	logical	ulteriori informazioni sulla sessione
4	notes	logical	ulteriori informazioni sulla sessione
5	last.match	numerica	giorni passati dall'ultima partita
6	next.match	numerica	giorni che mancano alla prossima partita
7	type	carattere	la sessione (S) può essere composta da più drill (D)
8	athlete	carattere	identificatore o nome del giocatore

9	role	carattere	ruolo del giocatore
10	duration	numerica	indica la durata della sessione del giocatore in minuti e secondi
Main			
11	total time	numerica	durata massima della sessione tra tutti i giocatori
12	distance	numerica	metri percorsi dal giocatore
13	avg speed	numerica	velocità media in km all'ora
14	avg HR	numerica	frequenza cardiaca media in battiti al minuto, presente solo se usato il cardiofrequenzimetro
15	avg HRR percentuale	numerica	è la differenza tra la frequenza cardiaca massima e la frequenza cardiaca a riposo, cioè minima
16	RPE	numerica	scala di percezione dello sforzo, serve per valutare la percezione soggettiva dello sforzo fisico
17	RPE duration	numerica	durata della sessione
18	TL	numerica	Training Load, prodotto di RPE e RPE duration espresso in Arbitrary Units
19	max speed	numerica	massima velocità raggiunta in km all'ora
20	max acc	numerica	massima accelerazione raggiunta in metri al secondo quadrato

21	max dec	numerica	massima decelerazione raggiunta in metri al secondo quadrato
22	max HR	numerica	frequenza cardiaca masima raggiunta nella sessione
23	max HRR percentuale	numerica	max HR espresso come percentuale della riserva individuale di battito cardiaco
Metabolic Exercise Training			
24	eq distance	numerica	distanza equivalente che avrebbe percorso l'atleta a velocità costante con la stessa energia usata nell'allenamento
25	eq distance index	numerica	rapporto tra eq distance e distance
26	avg met power	numerica	media del MET, parametri legati all'approccio energetico (velocità per costo energetico)
27	energy	numerica	energia (aerobica) totale spesa, Joule su kg
28	an energy	numerica	energia anaerobica, joule su kg
29	an index	numerica	rapporto tra l'energia anaerobica e energia totale

30	avg VO2	numerica	Il consumo medio di ossigeno è un parametro biologico che esprime il volume massimo di ossigeno che un essere umano può consumare nell'unità di tempo per contrazione muscolare. Questo valore è espresso in Watt su kg
31	aerobic ratio	numerica	rapporto tra avg MET power e VO2 massimo, dà informazioni sull'intensità media aerobica sostenuta
32	max met power	numerica	massimo MET, Watt su kg
Metabolic Power Events			
33	met power events	numerica	numero di richieste da parte del corpo di grande energia
34	MPE avg time	numerica	work average time (in seconds)
35	MPE avg power	numerica	work average metabolic power (in seconds), Watt su kg
36	MPE rec avg time	numerica	tempo di recupero medio, in secondi
37	MPE rec avg power	numerica	recovery average metabolic power, Watt su kg
38	MPE t Z1	numerica	indica il numero di eventi che durano meno della prima soglia (tempo in secondi)

39	MPE t Z2	numerica	indica il numero di eventi che durano tra prima e seconda soglia
40	MPE t Z3	numerica	indica il numero di eventi che durano tra seconda e terza soglia
41	MPE dist Z1	numerica	indica il numero di eventi che durano meno della prima soglia (distanza in metri)
42	MPE dist Z2	numerica	indica il numero di eventi che durano tra prima e seconda soglia
43	MPE dist Z3	numerica	indica il numero di eventi che durano tra seconda e terza soglia
44	MPE max sp Z1	numerica	indica il numero di eventi che durano meno della prima soglia (velocità in metri al secondo)
45	MPE max sp Z2	numerica	indica il numero di eventi che durano tra prima e seconda soglia
46	MPE max sp Z3	numerica	indica il numero di eventi che durano tra seconda e terza soglia

MECH

47	Active Muscle Load	numerica	lavoro totale fatto dall'atleta in Joule su kg

48	AMP	numerica	Average Active Muscle Power, potenza media sostenuta dai muscoli attivi, indicatore dell'intensità muscolare, Watt su kg
49	Eccentric Index	numerica	rapporto tra Active Muscle Power e Mechanical Power
Locomotion			
50	walk time	carattere	minuti e secondi camminati nella sessione
51	walk distance	numerica	metri camminati nella sessione
52	walk energy	numerica	energia spesa nella camminata, Joule su kg
53	run time	carattere	minuti e secondi di corsa nella sessione
54	run distance	numerica	metri corsi nella sessione
55	run energy	numerica	energia sepsa correndo, Joule su kg
56	forward distance	carattere	metri corsi in avanti
57	backward distance	carattere	metri corsi all'indietro
58	left distance	carattere	metri corsi verso sinistra
59	right distance	carattere	metri corsi verso destra
Speed Zones			

60	speed events	numerica	numero di scatti, tengono conto o di una certa velocità raggiunta o di un minimo di secondi di durata della corsa
61	distance sp Z1	numerica	distanza percorsa (in metri) all'interno della prima soglia
62	distance sp Z2	numerica	distanza percorsa (in metri) tra prima e seconda soglia
63	distance sp Z3	numerica	distanza percorsa (in metri) tra seconda e terza soglia
64	distance sp Z4	numerica	distanza percorsa (in metri) tra terza e quarta soglia
65	distance sp Z5	numerica	distanza percorsa (in metri) tra quarta e quinta soglia
66	distance sp Z6	numerica	distanza percorsa (in metri) tra quinta e sesta soglia
67	time sp Z1	carattere	tempo speso (in minuti e secondi) all'interno della prima soglia
68	time sp Z2	carattere	tempo speso (in minuti e secondi) tra prima e seconda soglia
69	time sp Z3	carattere	tempo speso (in minuti e secondi) tra seconda e terza soglia

70	time sp Z4	carattere	tempo speso (in minuti e secondi) tra terza e quarta soglia
71	time sp Z5	carattere	tempo speso (in minuti e secondi) tra quarta e quinta soglia
72	time sp Z6	carattere	tempo speso (in minuti e secondi) tra quarta e quinta soglia
Power Zones			
73	distance p Z1	numerica	distanza percorsa (in metri) all'interno della prima soglia
74	distance p Z2	numerica	distanza percorsa (in metri) tra prima e seconda soglia
75	distance p Z3	numerica	distanza percorsa (in metri) tra seconda e terza soglia
76	distance p Z4	numerica	distanza percorsa (in metri) tra terza e quarta soglia
77	distance p Z5	numerica	distanza percorsa (in metri) tra quarta e quinta soglia
78	distance p Z6	numerica	distanza percorsa (in metri) tra quinta e sesta soglia
79	time p Z1	carattere	tempo speso (in minuti e secondi) all'interno della prima soglia

80	time p Z2	carattere	tempo speso (in minuti e secondi) tra prima e seconda soglia
81	time p Z3	carattere	tempo speso (in minuti e secondi) tra seconda e terza soglia
82	time p Z4	carattere	tempo speso (in minuti e secondi) tra terza e quarta soglia
83	time p Z5	carattere	tempo speso (in minuti e secondi) tra quarta e quinta soglia
84	time p Z6	carattere	tempo speso (in minuti e secondi) tra quinta e sesta soglia
HR Zones			
85	distance HR Z1	numerica	distanza percorsa (in metri) all'interno della prima soglia
86	distance HR Z2	numerica	distanza percorsa (in metri) tra prima e seconda soglia
87	distance HR Z3	numerica	distanza percorsa (in metri) tra seconda e terza soglia
88	distance HR Z4	numerica	distanza percorsa (in metri) tra terza e quarta soglia
89	distance HR Z5	numerica	distanza percorsa (in metri) tra quarta e quinta soglia

90	distance HR Z6	numerica	distanza percorsa (in metri) tra quinta e sesta soglia
91	time HR Z1	carattere	tempo speso (in minuti e secondi) all'interno della prima soglia
92	time HR Z2	carattere	tempo speso (in minuti e secondi) tra prima e seconda soglia
93	time HR Z3	carattere	tempo speso (in minuti e secondi) tra seconda e terza soglia
94	time HR Z4	carattere	tempo speso (in minuti e secondi) tra terza e quarta soglia
95	time HR Z5	carattere	tempo speso (in minuti e secondi) tra quarta e quinta soglia
96	time HR Z6	carattere	tempo speso (in minuti e secondi) tra quinta e sesta soglia

Acc Zones

97	acc events	numerica	numero di accelerazioni, tengono conto o di una certa accelerazione raggiunta o di un minimo di secondi di durata dello sforzo
98	distance acc Z1	numerica	distanza percorsa (in metri) all'interno della prima soglia
99	distance acc Z2	numerica	distanza percorsa (in metri) tra prima e seconda soglia

100	distance acc Z3	numerica	distanza percorsa (in metri) tra seconda e terza soglia
101	distance acc Z4	numerica	distanza percorsa (in metri) tra terza e quarta soglia
102	distance acc Z5	numerica	distanza percorsa (in metri) tra quarta e quinta soglia
103	distance acc Z6	numerica	distanza percorsa (in metri) tra quinta e sesta soglia
104	time acc Z1	carattere	tempo speso (in minuti e secondi) all'interno della prima soglia
105	time acc Z2	carattere	tempo speso (in minuti e secondi) tra prima e seconda soglia
106	time acc Z3	carattere	tempo speso (in minuti e secondi) tra seconda e terza soglia
107	time acc Z4	carattere	tempo speso (in minuti e secondi) tra terza e quarta soglia
108	time acc Z5	carattere	tempo speso (in minuti e secondi) tra quarta e quinta soglia
109	time acc Z6	carattere	tempo speso (in minuti e secondi) tra quinta e sesta soglia

Dec Zones

110	dec events	numerica	numero di decelerazioni, tengono conto o di una certa accelerazione raggiunta o di un minimo di secondi di durata dello sforzo
111	distance dec Z1	numerica	distanza percorsa (in metri) all'interno della prima soglia
112	distance dec Z2	numerica	distanza percorsa (in metri) tra prima e seconda soglia
113	distance dec Z3	numerica	distanza percorsa (in metri) tra seconda e terza soglia
114	distance dec Z4	numerica	distanza percorsa (in metri) tra terza e quarta soglia
115	distance dec Z5	numerica	distanza percorsa (in metri) tra quarta e quinta soglia
116	distance dec Z6	numerica	distanza percorsa (in metri) tra quinta e sesta soglia
117	time dec Z1	carattere	tempo speso (in minuti e secondi) all'interno della prima soglia
118	time dec Z2	carattere	tempo speso (in minuti e secondi) tra prima e seconda soglia
119	time dec Z3	carattere	tempo speso (in minuti e secondi) tra seconda e terza soglia

120	time dec Z4	carattere	tempo speso (in minuti e secondi) tra terza e quarta soglia
121	time dec Z5	carattere	tempo speso (in minuti e secondi) tra quarta e quinta soglia
122	time dec Z6	carattere	tempo speso (in minuti e secondi) tra quinta e sesta soglia
IMU Events			
123	impacts	numerica	numero totale di impatti, ossia di scontri con altri giocatori (metri al secondo quadrato)
124	impacts Z1	numerica	numero di impatti nella prima soglia di intensità
125	impacts Z2	numerica	numero di impatti tra prima e seconda soglia di intensità
126	impacts Z3	numerica	numero di impatti tra seconda e terza soglia di intensità
127	jumps	numerica	numero totale di salti (altezza in metri)
128	jumps Z1	numerica	numero di salti nella prima soglia di altezza
129	jumps Z2	numerica	numero di salti tra prima e seconda soglia di altezza
130	jumps Z3	numerica	numero di salti tra seconda e terza soglia di altezza

Tabella A.1: Descrizione di tutte le variabili presenti nel *Dataset Sintetico*

Appendice B

Analisi grafica delle simulazioni

B.1 Simulazione con 7 covariate più l'intercetta, 100 giocatori

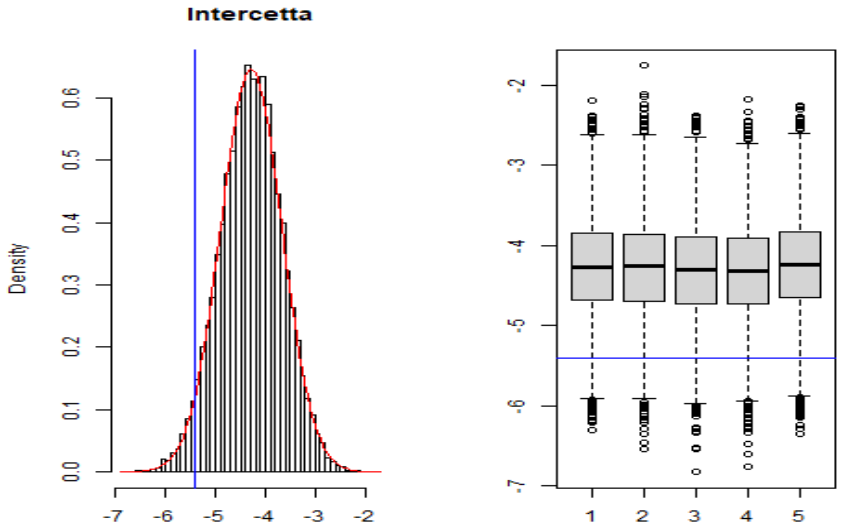

Figura B.1: Simulazione del parametro relativo all'intercetta. La curva rossa indica la distribuzione Normale con media e varianza del parametro, la linea blu rappresenta il valore del parametro stimato.

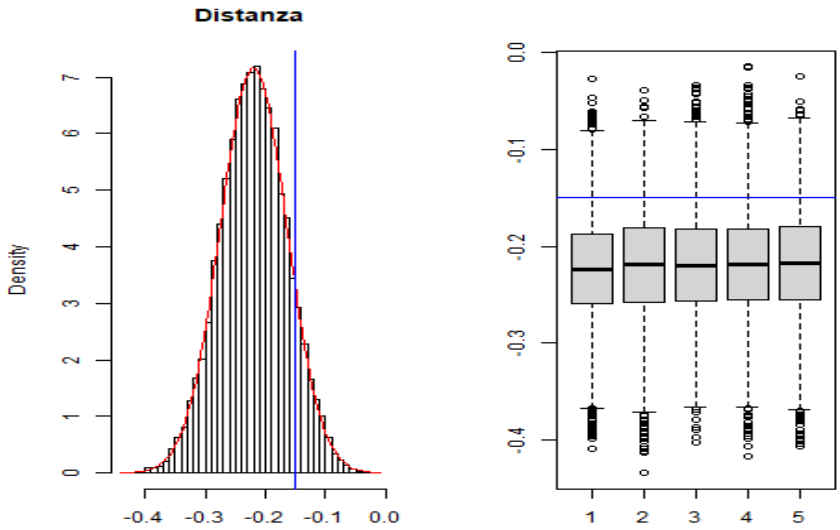

Figura B.2: Simulazione del parametro relativo alla distanza. La curva rossa indica la distribuzione Normale con media e varianza del parametro, la linea blu rappresenta il valore del parametro stimato.

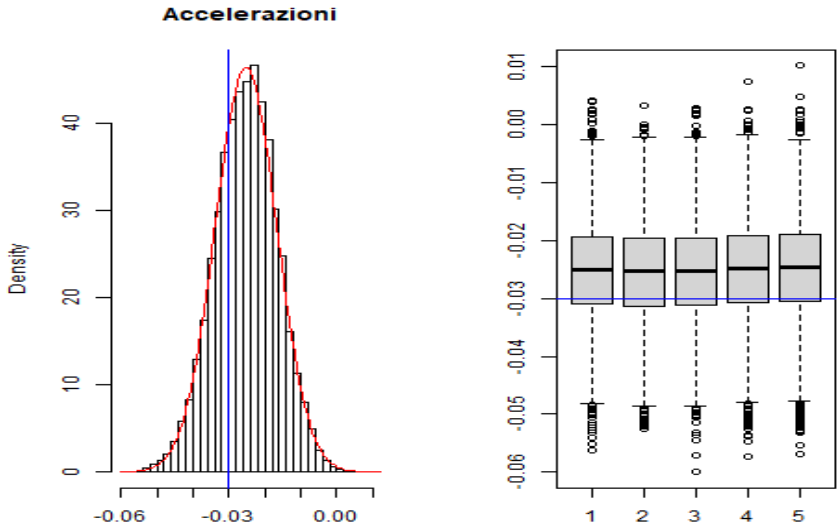

Figura B.3: Simulazione del parametro relativo alle accelerazioni. La curva rossa indica la distribuzione Normale con media e varianza del parametro, la linea blu rappresenta il valore del parametro stimato.

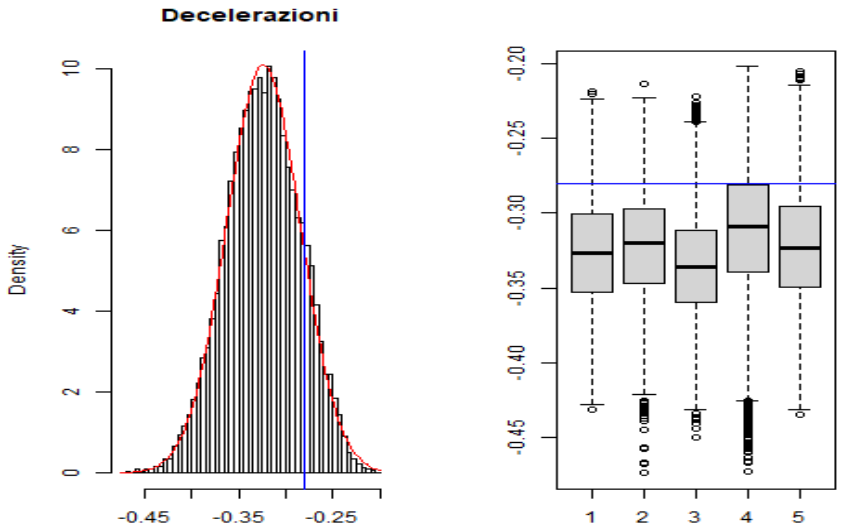

Figura B.4: Simulazione del parametro relativo alle decelerazioni. La curva rossa indica la distribuzione Normale con media e varianza del parametro, la linea blu rappresenta il valore del parametro stimato.

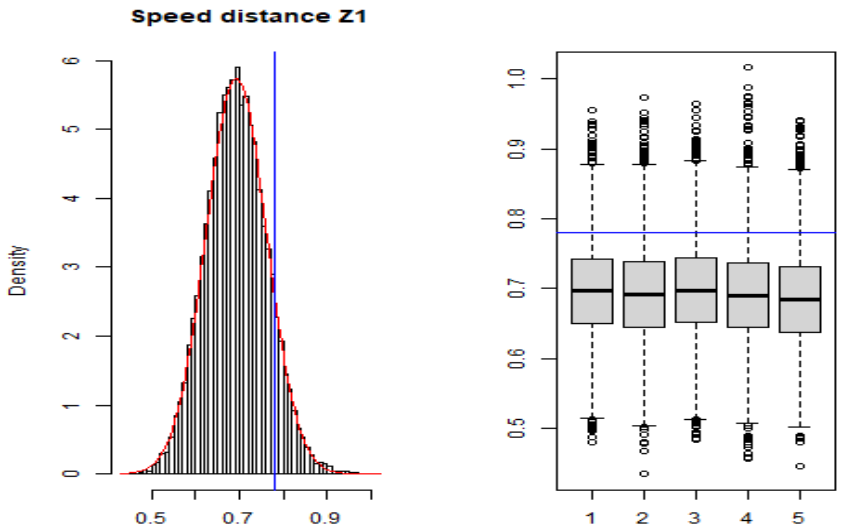

Figura B.5: Simulazione del parametro relativo alla distanza percorsa nella prima zona di velocità. La curva rossa indica la distribuzione Normale con media e varianza del parametro, la linea blu rappresenta il valore del parametro stimato.

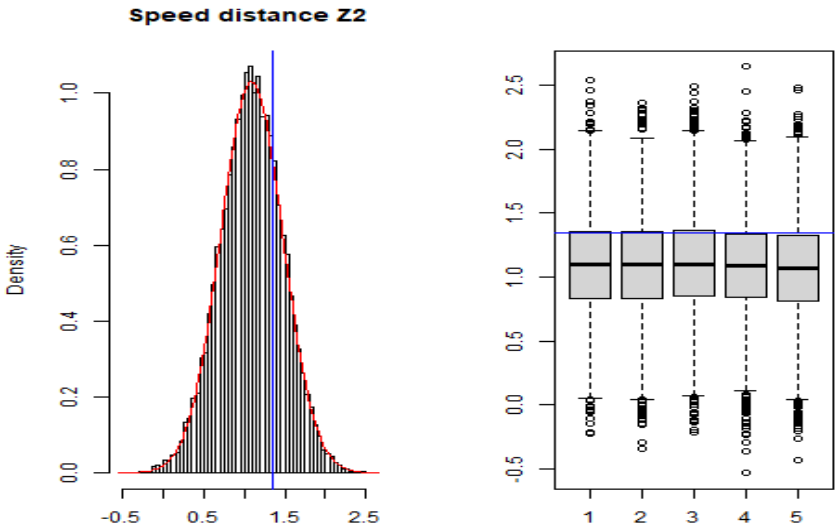

Figura B.6: Simulazione del parametro relativo alla distanza percorsa nella seconda zona di velocità. La curva rossa indica la distribuzione Normale con media e varianza del parametro, la linea blu rappresenta il valore del parametro stimato.

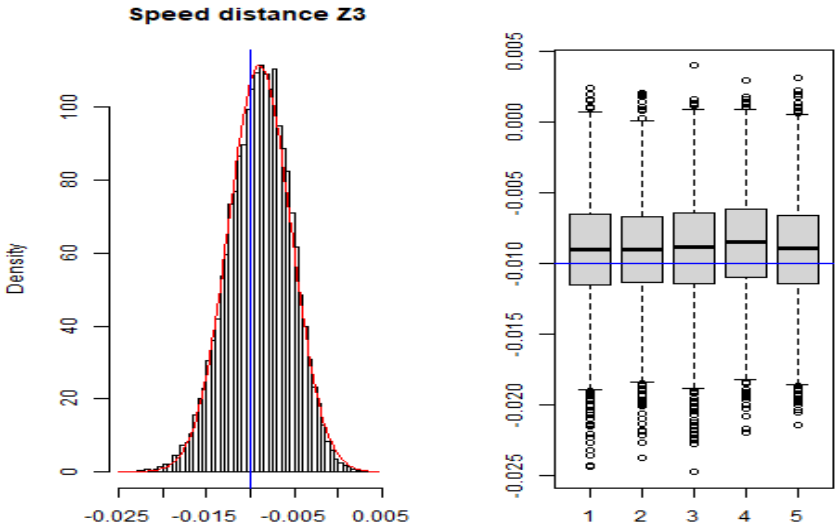

Figura B.7: Simulazione del parametro relativo alla distanza percorsa nella terza zona di velocità. La curva rossa indica la distribuzione Normale con media e varianza del parametro, la linea blu rappresenta il valore del parametro stimato.

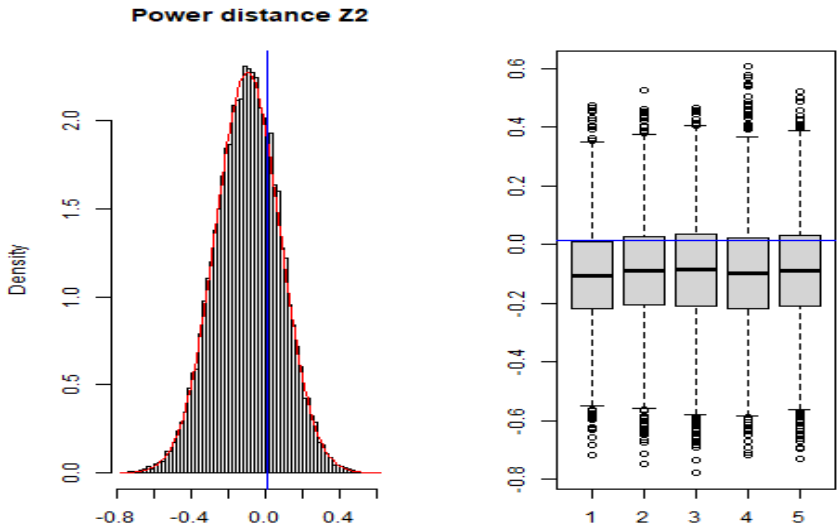

Figura B.8: Simulazione del parametro relativo alla distanza percorsa nella seconda zona di potenza. La curva rossa indica la distribuzione Normale con media e varianza del parametro, la linea blu rappresenta il valore del parametro stimato.

B.2 Simulazione con 7 covariate più l'intercetta, 500 giocatori

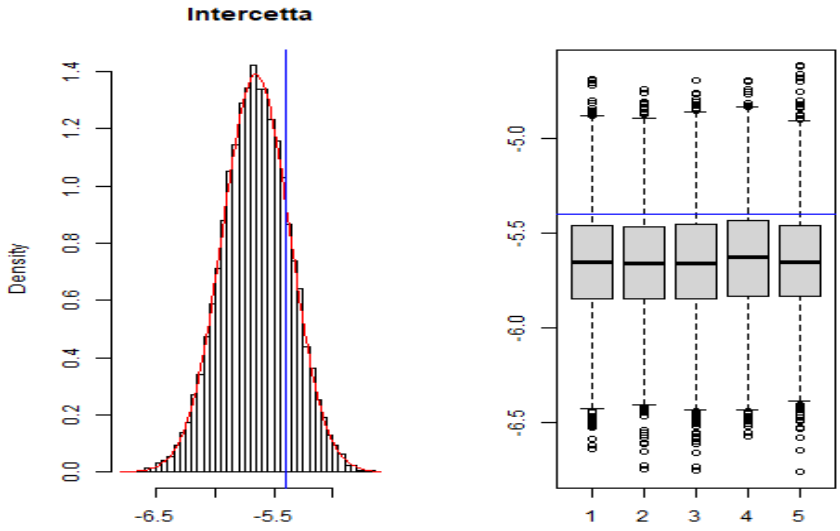

Figura B.9: Simulazione del parametro relativo all'intercetta. La curva rossa indica la distribuzione Normale con media e varianza del parametro, la linea blu rappresenta il valore del parametro stimato.

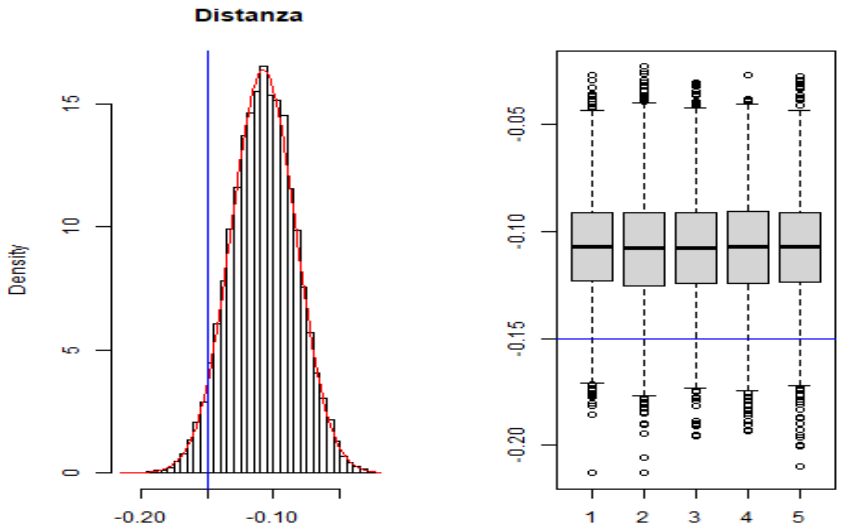

Figura B.10: Simulazione del parametro relativo alla distanza. La curva rossa indica la distribuzione Normale con media e varianza del parametro, la linea blu rappresenta il valore del parametro stimato.

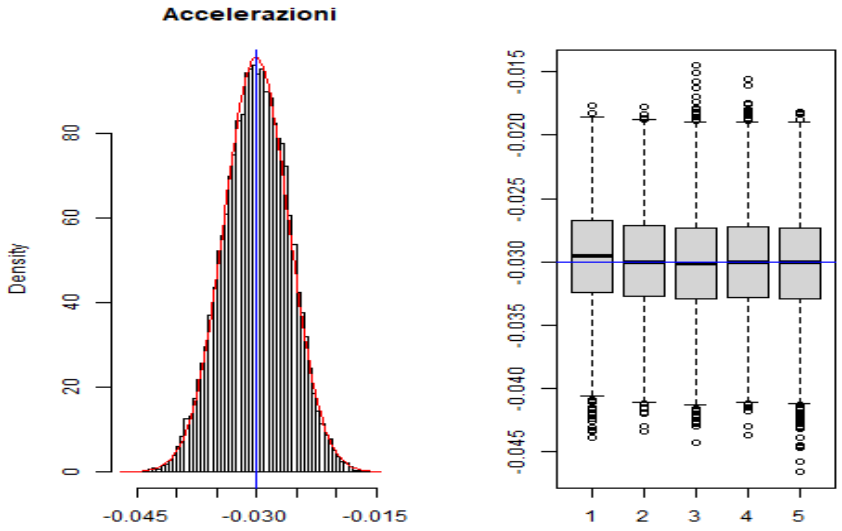

Figura B.11: Simulazione del parametro relativo alle accelerazioni. La curva rossa indica la distribuzione Normale con media e varianza del parametro, la linea blu rappresenta il valore del parametro stimato.

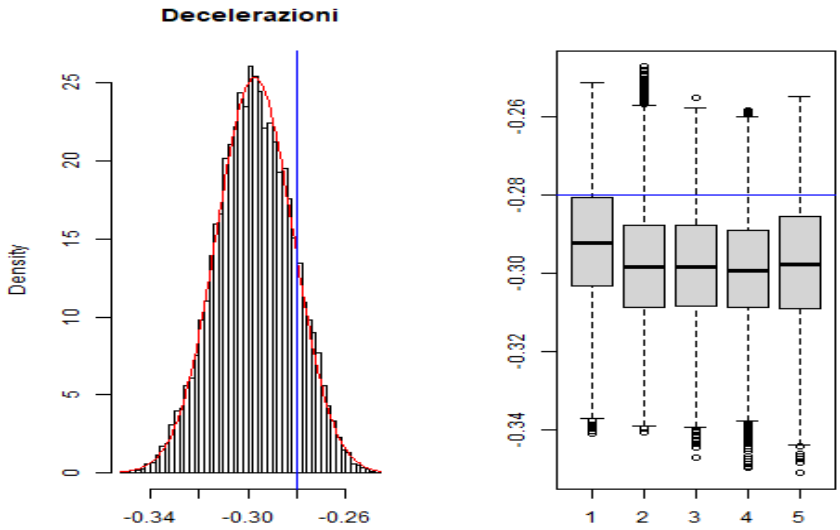

Figura B.12: Simulazione del parametro relativo alle decelerazioni. La curva rossa indica la distribuzione Normale con media e varianza del parametro, la linea blu rappresenta il valore del parametro stimato.

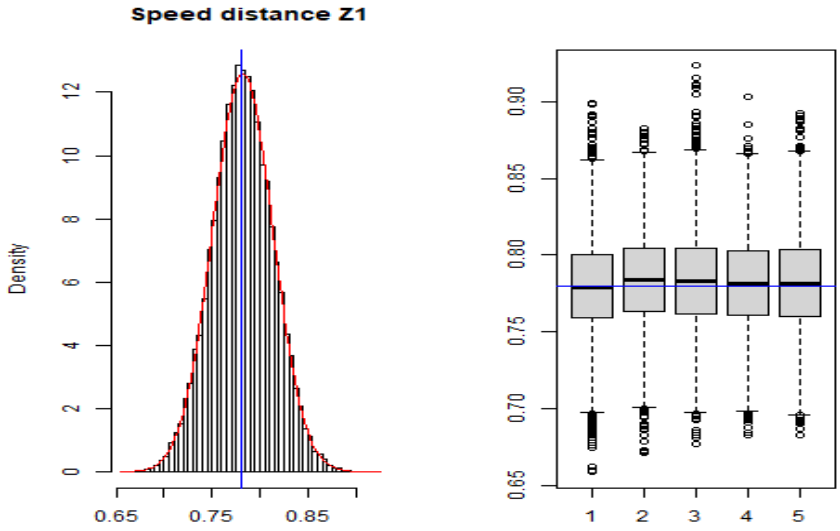

Figura B.13: Simulazione del parametro relativo alla distanza percorsa nella prima zona di velocità. La curva rossa indica la distribuzione Normale con media e varianza del parametro, la linea blu rappresenta il valore del parametro stimato.

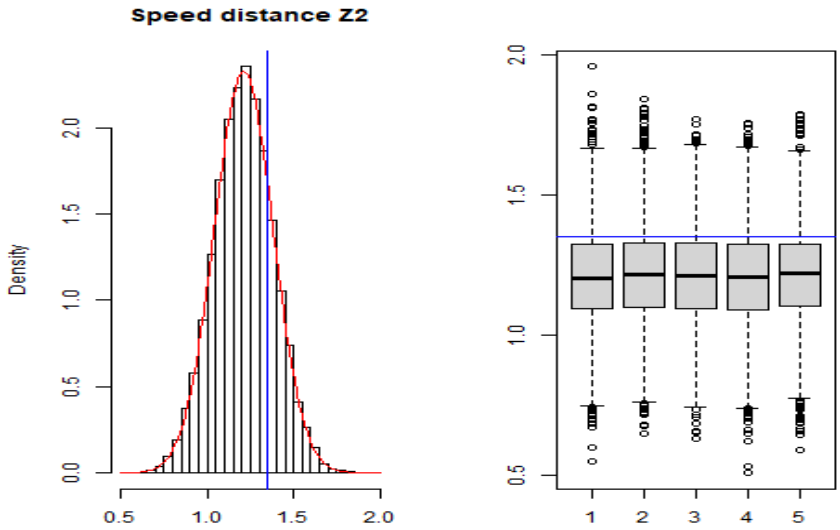

Figura B.14: Simulazione del parametro relativo alla distanza percorsa nella seconda zona di velocità. La curva rossa indica la distribuzione Normale con media e varianza del parametro, la linea blu rappresenta il valore del parametro stimato.

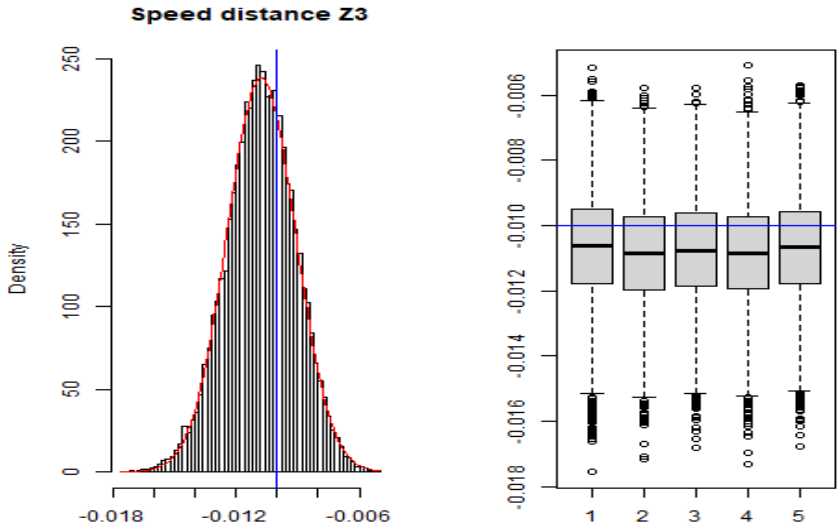

Figura B.15: Simulazione del parametro relativo alla distanza percorsa nella terza zona di velocità. La curva rossa indica la distribuzione Normale con media e varianza del parametro, la linea blu rappresenta il valore del parametro stimato.

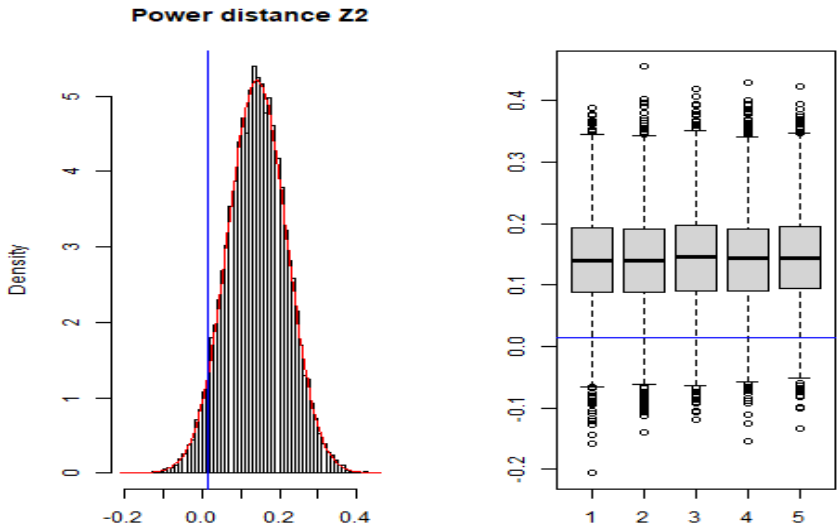

Figura B.16: Simulazione del parametro relativo alla distanza percorsa nella seconda zona di potenza. La curva rossa indica la distribuzione Normale con media e varianza del parametro, la linea blu rappresenta il valore del parametro stimato.

B.3 Simulazione con 7 covariate, 100 giocatori

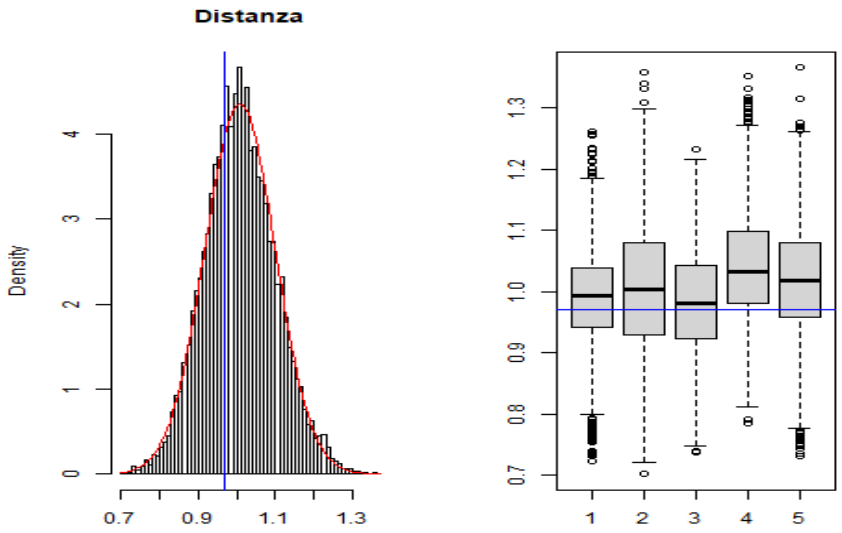

Figura B.17: Simulazione del parametro relativo alla distanza. La curva rossa indica la distribuzione Normale con media e varianza del parametro, la linea blu rappresenta il valore del parametro stimato.

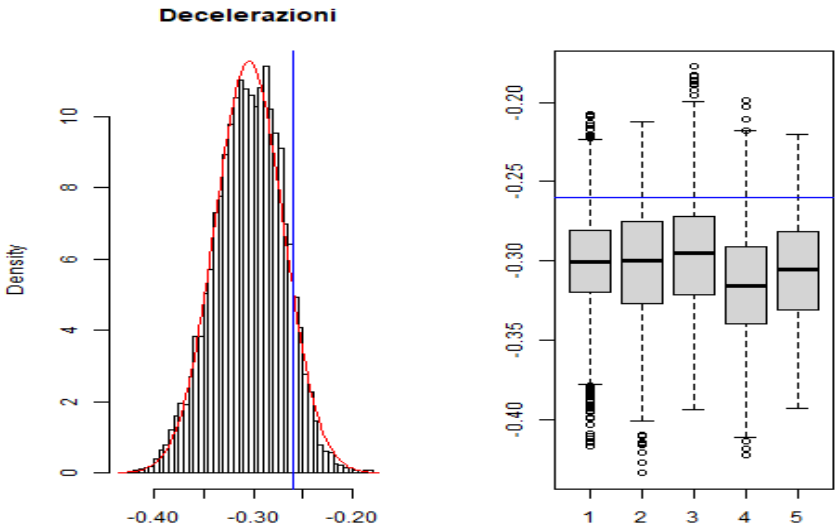

Figura B.18: Simulazione del parametro relativo alle decelerazioni. La curva rossa indica la distribuzione Normale con media e varianza del parametro, la linea blu rappresenta il valore del parametro stimato.

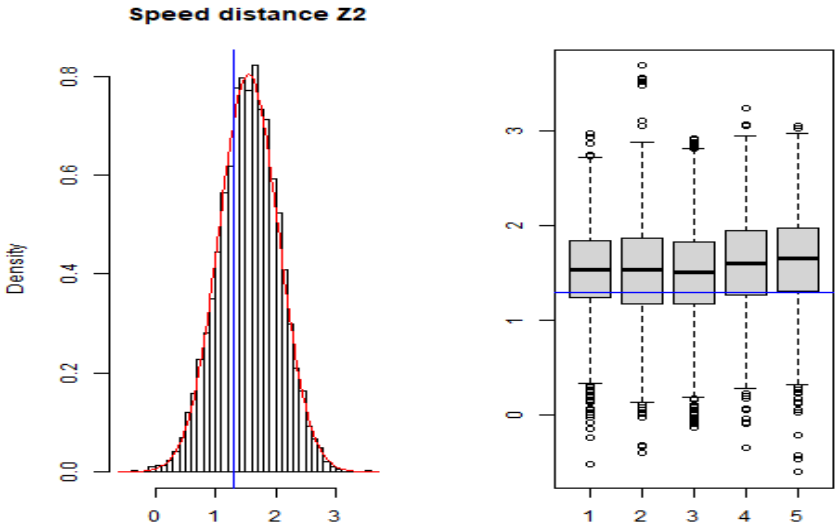

Figura B.19: Simulazione del parametro relativo alla distanza percorsa nella seconda zona di potenza. La curva rossa indica la distribuzione Normale con media e varianza del parametro, la linea blu rappresenta il valore del parametro stimato.

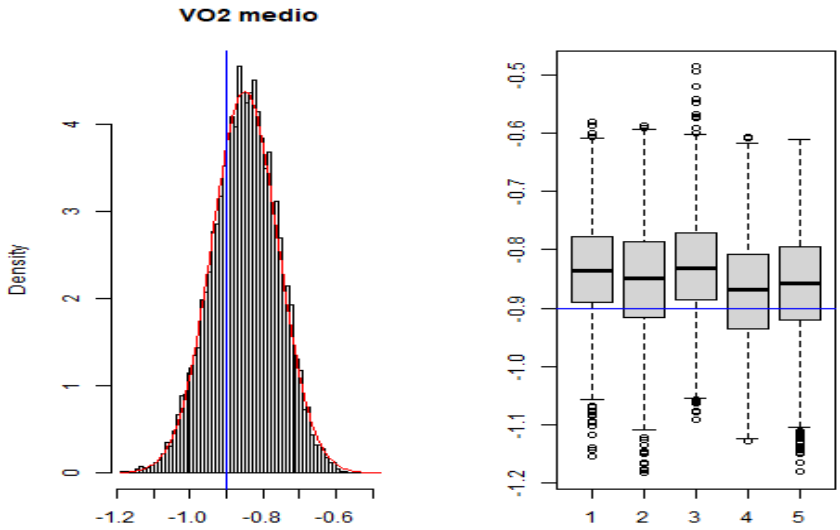

Figura B.20: Simulazione del parametro relativo al consumo medio di energia aerobica. La curva rossa indica la distribuzione Normale con media e varianza del parametro, la linea blu rappresenta il valore del parametro stimato.

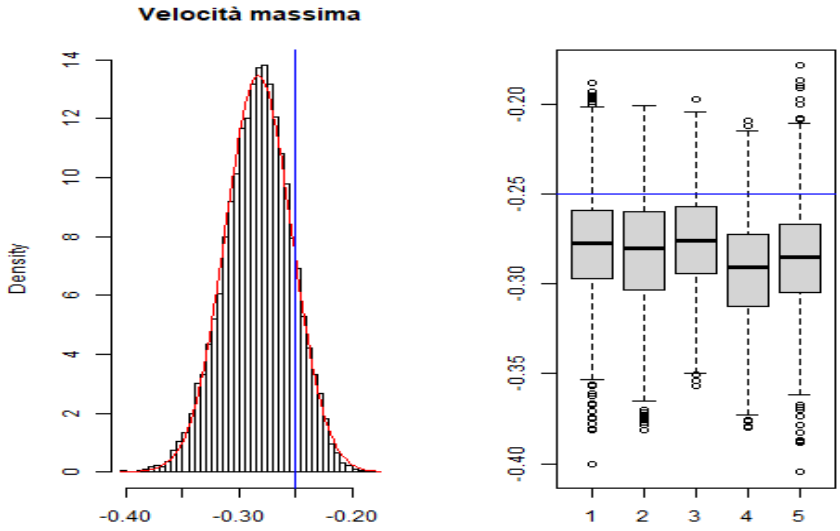

Figura B.21: Simulazione del parametro relativo alla velocità massima raggiunta. La curva rossa indica la distribuzione Normale con media e varianza del parametro, la linea blu rappresenta il valore del parametro stimato.

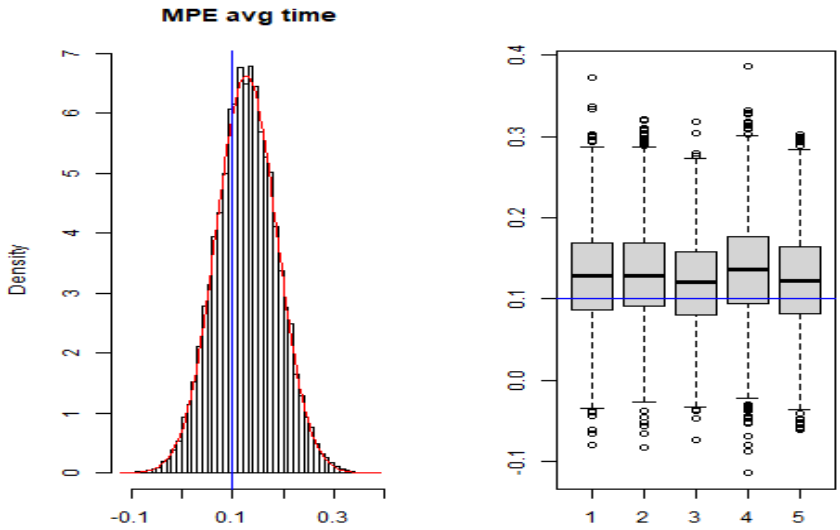

Figura B.22: Simulazione del parametro relativo alla durata media un MPE. La curva rossa indica la distribuzione Normale con media e varianza del parametro, la linea blu rappresenta il valore del parametro stimato.

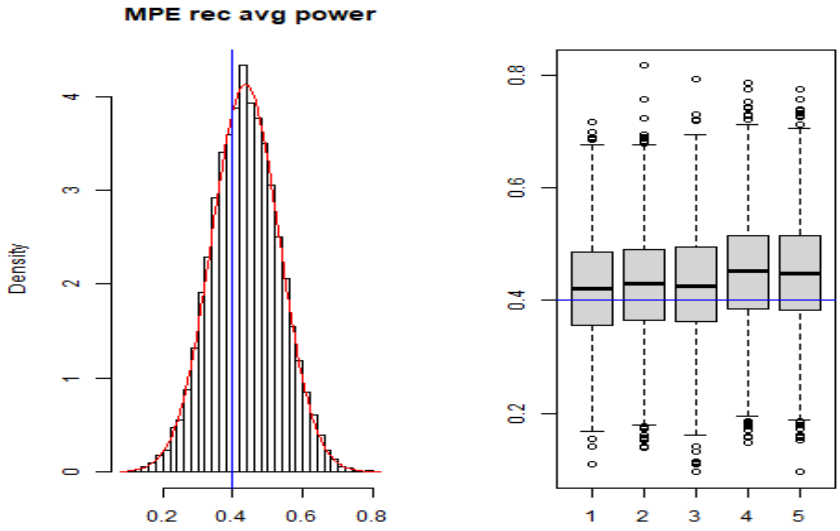

Figura B.23: Simulazione del parametro relativo alla potenza media di recupero da un MPE. La curva rossa indica la distribuzione Normale con media e varianza del parametro, la linea blu rappresenta il valore del parametro stimato.

B.4 Simulazione con 7 covariate, 500 giocatori

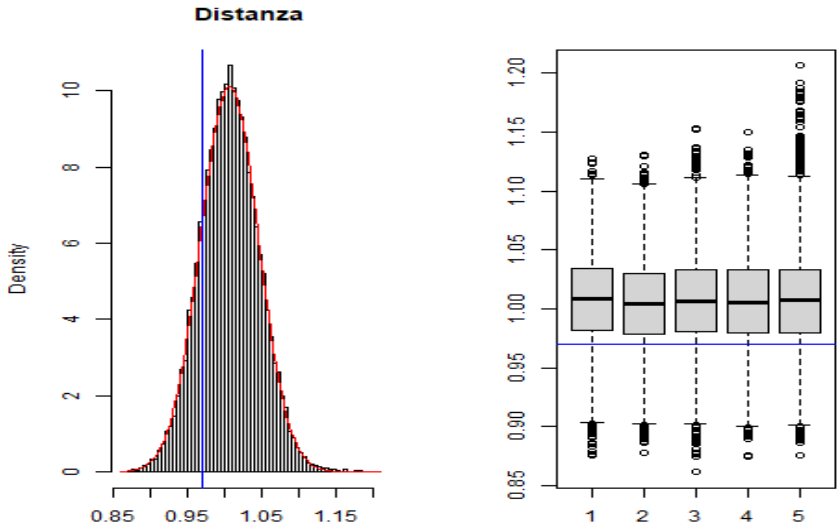

Figura B.24: Simulazione del parametro relativo alla distanza. La curva rossa indica la distribuzione Normale con media e varianza del parametro, la linea blu rappresenta il valore del parametro stimato.

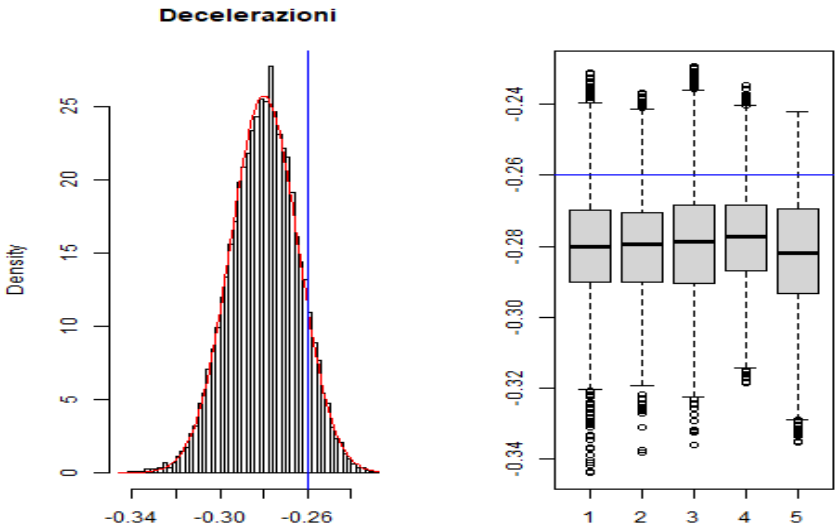

Figura B.25: Simulazione del parametro relativo alle decelerazioni. La curva rossa indica la distribuzione Normale con media e varianza del parametro, la linea blu rappresenta il valore del parametro stimato.

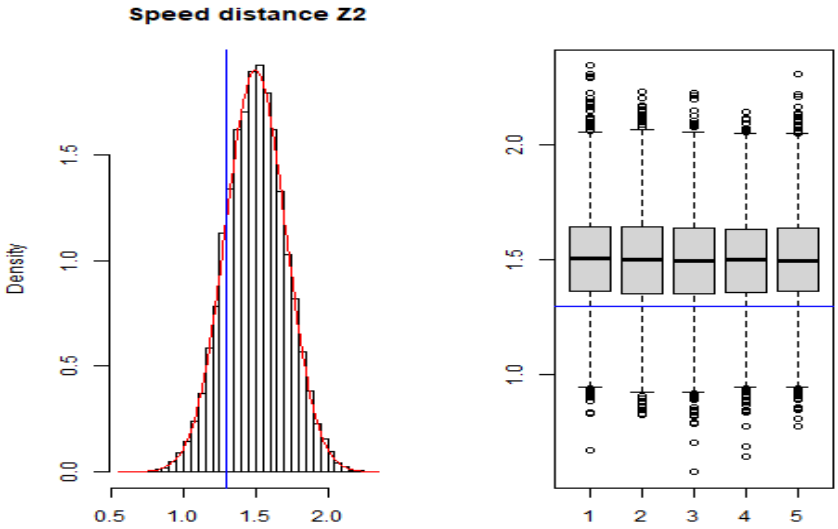

Figura B.26: Simulazione del parametro relativo alla distanza percorsa nella seconda zona di potenza. La curva rossa indica la distribuzione Normale con media e varianza del parametro, la linea blu rappresenta il valore del parametro stimato.

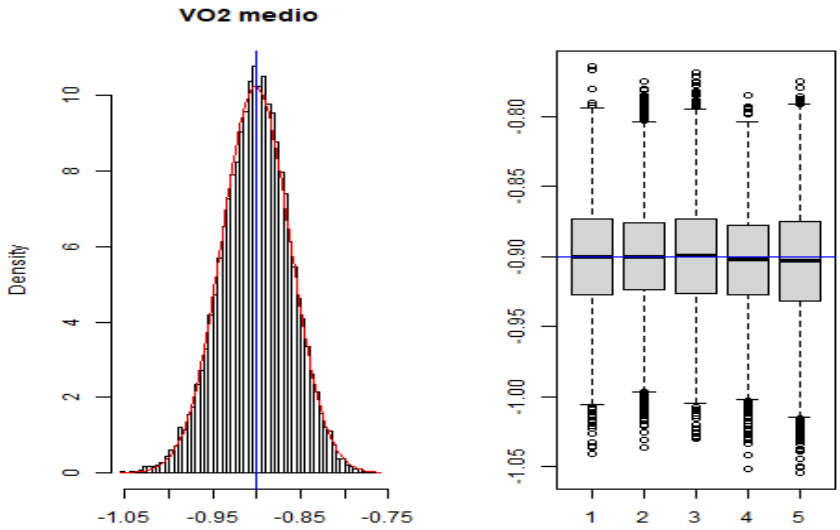

Figura B.27: Simulazione del parametro relativo al consumo medio di energia aerobica. La curva rossa indica la distribuzione Normale con media e varianza del parametro, la linea blu rappresenta il valore del parametro stimato.

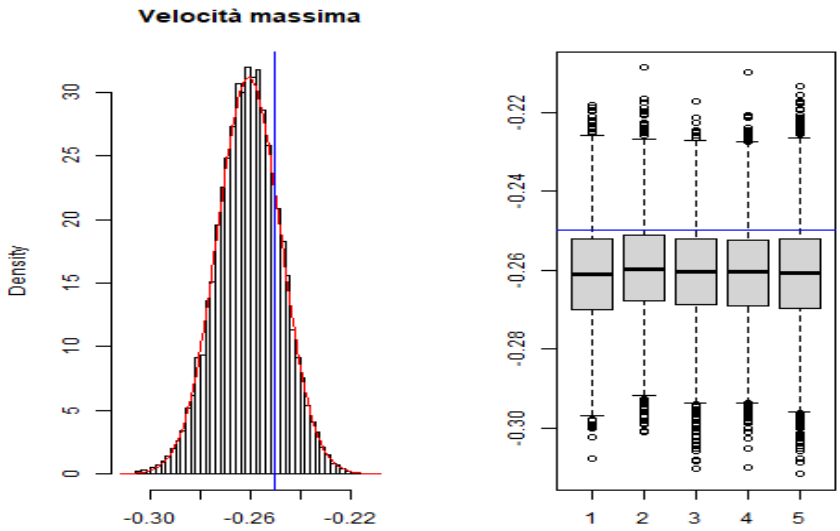

Figura B.28: Simulazione del parametro relativo alla velocità massima raggiunta. La curva rossa indica la distribuzione Normale con media e varianza del parametro, la linea blu rappresenta il valore del parametro stimato.

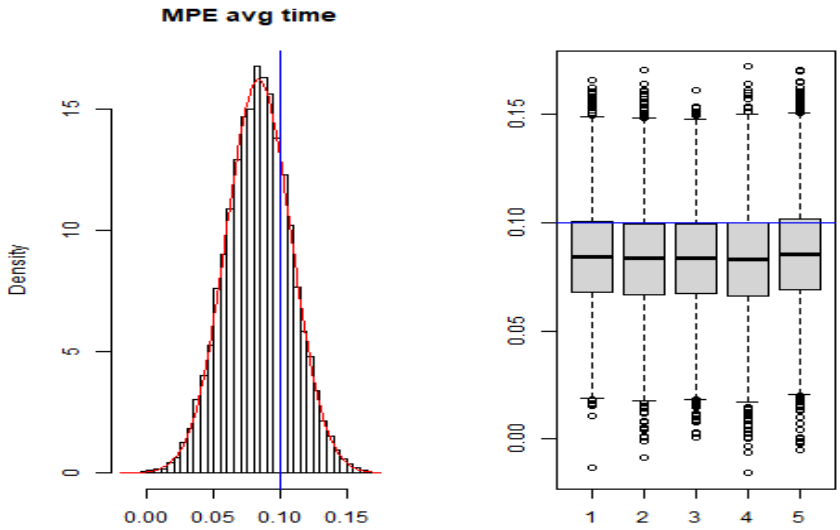

Figura B.29: Simulazione del parametro relativo alla durata media un MPE. La curva rossa indica la distribuzione Normale con media e varianza del parametro, la linea blu rappresenta il valore del parametro stimato.

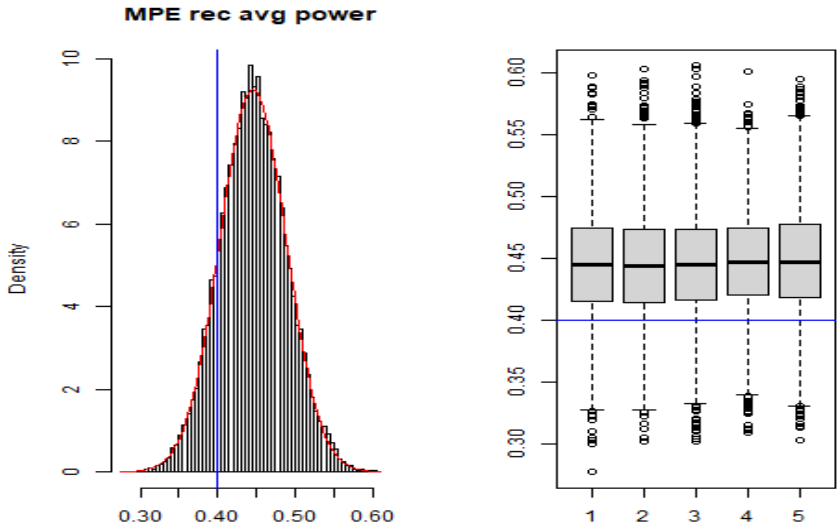

Figura B.30: Simulazione del parametro relativo alla potenza media di recupero da un MPE. La curva rossa indica la distribuzione Normale con media e varianza del parametro, la linea blu rappresenta il valore del parametro stimato.

Appendice C

Codice R utilizzato

C.1 Pulizia Dataset Sintetico

Listing C.1: Pulizia dati TEAM 1

```
#Allenamenti iniziati il 08/05/20, prima partita 20/06/20. Se
   un giocatore ha l'asterisco vicino al nome vuol dire che ha
   fatto un allenamento differenziato o esubentrato durante la
   partita, quindi non estato preso in considerazione per le
   statistiche di squadra.

TEAM1[TEAM1 =="—"] <- NA
TEAM1[TEAM1 ==""] <- NA

#A) Trasformo le variabili di durata in secondi

#a) duration..mm.ss.
duration.sec <- rep(NA, length(TEAM1$duration..mm.ss.))

for(n in 1:nrow(TEAM1))
{
  if (nchar(TEAM1$duration..mm.ss.[n]) == 5) duration.sec[n] <-
      as.integer(substr(TEAM1$duration..mm.ss.[n], 1, 2))*60 +
```

```
     as.integer(substr(TEAM1$duration..mm.ss.[n], 4, 5))

  if (nchar(TEAM1$duration..mm.ss.[n]) == 6) duration.sec[n] <-
      as.integer(substr(TEAM1$duration..mm.ss.[n], 1, 3))*60 +
     as.integer(substr(TEAM1$duration..mm.ss.[n], 5, 6))
}

#La metto in minuti
duration.min <- duration.sec/60


#b) total.time..mm.ss.
total.sec <- rep(NA, length(TEAM1$total.time..mm.ss.))

for(n in 1:nrow(TEAM1))
{
  if (nchar(TEAM1$total.time..mm.ss.[n]) == 5) total.sec[n] <-
     as.integer(substr(TEAM1$total.time..mm.ss.[n], 1, 2))*60 +
      as.integer(substr(TEAM1$total.time..mm.ss.[n], 4, 5))

  if (nchar(TEAM1$total.time..mm.ss.[n]) == 6) total.sec[n] <-
     as.integer(substr(TEAM1$total.time..mm.ss.[n], 1, 3))*60 +
      as.integer(substr(TEAM1$total.time..mm.ss.[n], 5, 6))
}

#La metto in minuti
total.min <- total.sec/60


#c)walk.time..mm.ss.
walk.sec <- rep(NA, length(TEAM1$walk.time..mm.ss.))

for(n in 1:nrow(TEAM1))
{
  if (nchar(TEAM1$walk.time..mm.ss.[n]) == 5) walk.sec[n] <- as
     .integer(substr(TEAM1$walk.time..mm.ss.[n], 1, 2))*60 + as
```

```
     .integer(substr(TEAM1$walk.time..mm.ss.[n], 4, 5))

  if (nchar(TEAM1$walk.time..mm.ss.[n]) == 6) walk.sec[n] <- as
     .integer(substr(TEAM1$walk.time..mm.ss.[n], 1, 3))*60 + as
     .integer(substr(TEAM1$walk.time..mm.ss.[n], 5, 6))
}

#La metto in minuti
walk.min <- walk.sec/60


#d)run.time..mm.ss.
run.sec <- rep(NA, length(TEAM1$run.time..mm.ss.))

for(n in 1:nrow(TEAM1))
{
  if (nchar(TEAM1$run.time..mm.ss.[n]) == 5) run.sec[n] <- as.
     integer(substr(TEAM1$run.time..mm.ss.[n], 1, 2))*60 + as.
     integer(substr(TEAM1$run.time..mm.ss.[n], 4, 5))

  if (nchar(TEAM1$run.time..mm.ss.[n]) == 6) run.sec[n] <- as.
     integer(substr(TEAM1$run.time..mm.ss.[n], 1, 3))*60 + as.
     integer(substr(TEAM1$run.time..mm.ss.[n], 5, 6))
}

#La metto in minuti
run.min <- run.sec/60


#e)time...sp.Z1..mm.ss.
time.sp.Z1.sec <- rep(NA, length(TEAM1$time...sp.Z1..mm.ss.))

for(n in 1:nrow(TEAM1))
{
  if (nchar(TEAM1$time...sp.Z1..mm.ss.[n]) == 5) time.sp.Z1.sec
     [n] <- as.integer(substr(TEAM1$time...sp.Z1..mm.ss.[n], 1,
```

```
      2))*60 + as.integer(substr(TEAM1$time...sp.Z1..mm.ss.[n],
      4, 5))

  if (nchar(TEAM1$time...sp.Z1..mm.ss.[n]) == 6) time.sp.Z1.sec
     [n] <- as.integer(substr(TEAM1$time...sp.Z1..mm.ss.[n], 1,
      3))*60 + as.integer(substr(TEAM1$time...sp.Z1..mm.ss.[n],
      5, 6))
}

#La metto in minuti
time.sp.Z1.min <- time.sp.Z1.sec/60


#f)time...sp.Z2..mm.ss.
time.sp.Z2.sec <- rep(NA, length(TEAM1$time...sp.Z2..mm.ss.))

for(n in 1:nrow(TEAM1))
{
  if (nchar(TEAM1$time...sp.Z2..mm.ss.[n]) == 5) time.sp.Z2.sec
     [n] <- as.integer(substr(TEAM1$time...sp.Z2..mm.ss.[n], 1,
      2))*60 + as.integer(substr(TEAM1$time...sp.Z2..mm.ss.[n],
      4, 5))

  if (nchar(TEAM1$time...sp.Z2..mm.ss.[n]) == 6) time.sp.Z2.sec
     [n] <- as.integer(substr(TEAM1$time...sp.Z2..mm.ss.[n], 1,
      3))*60 + as.integer(substr(TEAM1$time...sp.Z2..mm.ss.[n],
      5, 6))
}

#La metto in minuti
time.sp.Z2.min <- time.sp.Z2.sec/60


#g)time...sp.Z3..mm.ss.
time.sp.Z3.sec <- rep(NA, length(TEAM1$time...sp.Z3..mm.ss.))
```

```
for(n in 1:nrow(TEAM1))
{
  if (nchar(TEAM1$time...sp.Z3..mm.ss.[n]) == 5) time.sp.Z3.sec
     [n] <- as.integer(substr(TEAM1$time...sp.Z3..mm.ss.[n], 1,
      2))*60 + as.integer(substr(TEAM1$time...sp.Z3..mm.ss.[n],
      4, 5))

  if (nchar(TEAM1$time...sp.Z3..mm.ss.[n]) == 6) time.sp.Z3.sec
     [n] <- as.integer(substr(TEAM1$time...sp.Z3..mm.ss.[n], 1,
      3))*60 + as.integer(substr(TEAM1$time...sp.Z3..mm.ss.[n],
      5, 6))
}

#La metto in minuti
time.sp.Z3.min <- time.sp.Z3.sec/60


#h)time...sp.Z4..mm.ss.
time.sp.Z4.sec <- rep(NA, length(TEAM1$time...sp.Z4..mm.ss.))

for(n in 1:nrow(TEAM1))
{
  if (nchar(TEAM1$time...sp.Z4..mm.ss.[n]) == 5) time.sp.Z4.sec
     [n] <- as.integer(substr(TEAM1$time...sp.Z4..mm.ss.[n], 1,
      2))*60 + as.integer(substr(TEAM1$time...sp.Z4..mm.ss.[n],
      4, 5))

  if (nchar(TEAM1$time...sp.Z4..mm.ss.[n]) == 6) time.sp.Z4.sec
     [n] <- as.integer(substr(TEAM1$time...sp.Z4..mm.ss.[n], 1,
      3))*60 + as.integer(substr(TEAM1$time...sp.Z4..mm.ss.[n],
      5, 6))
}

#La metto in minuti
time.sp.Z4.min <- time.sp.Z4.sec/60
```

```
#i)time...sp.Z5..mm.ss.
time.sp.Z5.sec <- rep(NA, length(TEAM1$time...sp.Z5..mm.ss.))

for(n in 1:nrow(TEAM1))
{
  if (nchar(TEAM1$time...sp.Z5..mm.ss.[n]) == 5) time.sp.Z5.sec
     [n] <- as.integer(substr(TEAM1$time...sp.Z5..mm.ss.[n], 1,
      2))*60 + as.integer(substr(TEAM1$time...sp.Z5..mm.ss.[n],
      4, 5))

  if (nchar(TEAM1$time...sp.Z5..mm.ss.[n]) == 6) time.sp.Z5.sec
     [n] <- as.integer(substr(TEAM1$time...sp.Z5..mm.ss.[n], 1,
      3))*60 + as.integer(substr(TEAM1$time...sp.Z5..mm.ss.[n],
      5, 6))
}

#La metto in minuti
time.sp.Z5.min <- time.sp.Z5.sec/60


#B) LE VARIABILI CREATE LE AGGIUNGO AL TEAM
TEAM1 <- cbind(TEAM1, duration.min, total.min, walk.min, run.
   min, time.sp.Z1.min, time.sp.Z2.min, time.sp.Z3.min, time.sp
   .Z4.min, time.sp.Z5.min)

#Elimino le variabili non utili
TEAM1 <- TEAM1[,-c
   (3,6,8,9,10,13,14,15,16,17,21,22,49,52,55,56,57,58,61,63,
   65,67,69,73,74,75,76,77,78,79,80,81,82,83,84,85,88,89,90,
   91,92,93,94,95)]

###
str(TEAM1)
table(TEAM1$next.match)
#carattere percheci sono dei "None"
```

```
TEAM1$next.match[TEAM1$next.match =="None"] <- Inf #se no
   mettere per es. 999

TEAM1$next.match <- as.numeric(TEAM1$next.match)
```

Listing C.2: Aggregare le zone

```
## MPE

#DISTANCE: il TEAM4 ha le zone diverse (0,20,30) mentre le
   altre squadre hanno (0,10,20), quindi creo solo 2 zone
   (0,20). TEAM4 aggrego Z2 e Z3, le altre sqaudre aggrego Z1 e
    Z2

MPE.dist.Z1.1 <- rep(NA, length(TEAM1$MPE...dist.Z1))
MPE.dist.Z1.2 <- rep(NA, length(TEAM2$MPE...dist.Z1))
MPE.dist.Z1.3 <- rep(NA, length(TEAM3$MPE...dist.Z1))
MPE.dist.Z1.4 <- rep(NA, length(TEAM4$MPE...dist.Z1))
MPE.dist.Z1.5 <- rep(NA, length(TEAM5$MPE...dist.Z1))

MPE.dist.Z2.1 <- rep(NA, length(TEAM1$MPE...dist.Z1))
MPE.dist.Z2.2 <- rep(NA, length(TEAM2$MPE...dist.Z1))
MPE.dist.Z2.3 <- rep(NA, length(TEAM3$MPE...dist.Z1))
MPE.dist.Z2.4 <- rep(NA, length(TEAM4$MPE...dist.Z1))
MPE.dist.Z2.5 <- rep(NA, length(TEAM5$MPE...dist.Z1))


for(i in 1:length(TEAM1$MPE...dist.Z1)) MPE.dist.Z1.1[i] <-
   TEAM1$MPE...dist.Z1[i] + TEAM1$MPE...dist.Z2[i]

for(i in 1:length(TEAM2$MPE...dist.Z1)) MPE.dist.Z1.2[i] <-
   TEAM2$MPE...dist.Z1[i] + TEAM2$MPE...dist.Z2[i]

for(i in 1:length(TEAM3$MPE...dist.Z1)) MPE.dist.Z1.3[i] <-
   TEAM3$MPE...dist.Z1[i] + TEAM3$MPE...dist.Z2[i]
```

```
for(i in 1:length(TEAM4$MPE...dist.Z1)) MPE.dist.Z1.4[i] <-
   TEAM4$MPE...dist.Z1[i]

for(i in 1:length(TEAM5$MPE...dist.Z1)) MPE.dist.Z1.5[i] <-
   TEAM5$MPE...dist.Z1[i] + TEAM5$MPE...dist.Z2[i]

#
for(i in 1:length(TEAM1$MPE...dist.Z3)) MPE.dist.Z2.1[i] <-
   TEAM1$MPE...dist.Z3[i]

for(i in 1:length(TEAM2$MPE...dist.Z3)) MPE.dist.Z2.2[i] <-
   TEAM2$MPE...dist.Z3[i]

for(i in 1:length(TEAM3$MPE...dist.Z3)) MPE.dist.Z2.3[i] <-
   TEAM3$MPE...dist.Z3[i]

for(i in 1:length(TEAM4$MPE...dist.Z3)) MPE.dist.Z2.4[i] <-
   TEAM4$MPE...dist.Z2[i] + TEAM4$MPE...dist.Z3[i]

for(i in 1:length(TEAM5$MPE...dist.Z3)) MPE.dist.Z2.5[i] <-
   TEAM5$MPE...dist.Z3[i]

TEAM1 <- cbind(TEAM1, MPE.dist.Z1.1, MPE.dist.Z2.1)
TEAM2 <- cbind(TEAM2, MPE.dist.Z1.2, MPE.dist.Z2.2)
TEAM3 <- cbind(TEAM3, MPE.dist.Z1.3, MPE.dist.Z2.3)
TEAM4 <- cbind(TEAM4, MPE.dist.Z1.4, MPE.dist.Z2.4)
TEAM5 <- cbind(TEAM5, MPE.dist.Z1.5, MPE.dist.Z2.5)

################################
##POWER

#DISTANCE: stiamo parlando di distanza, quindi metri (es. Z1=
   114.5)
#TEAM 1 ha 3 zone e quindi 2 soglie (0,25,200)
#TEAM 2 ha 4 zone e quindi 3 soglie (0,18,25,35)
#TEAM 3 ha 3 zone e quindi 2 soglie (0,20,35)
```

```
#TEAM 4 ha 3 zone e quindi 2 soglie (0,20,30)
#TEAM 5 ha 6 zone e quindi 5 soglie (0,10,20,30,40,50)

#Direi che la scelta piu sensata sia2 soglie (0,20-25).

Power.dist.Z1.1 <- rep(NA, length(TEAM1$distance...p.Z1..m.))
Power.dist.Z1.2 <- rep(NA, length(TEAM2$distance...p.Z1..m.))
Power.dist.Z1.3 <- rep(NA, length(TEAM3$distance...p.Z1..m.))
Power.dist.Z1.4 <- rep(NA, length(TEAM4$distance...p.Z1..m.))
Power.dist.Z1.5 <- rep(NA, length(TEAM5$distance...p.Z1..m.))

Power.dist.Z2.1 <- rep(NA, length(TEAM1$distance...p.Z1..m.))
Power.dist.Z2.2 <- rep(NA, length(TEAM2$distance...p.Z1..m.))
Power.dist.Z2.3 <- rep(NA, length(TEAM3$distance...p.Z1..m.))
Power.dist.Z2.4 <- rep(NA, length(TEAM4$distance...p.Z1..m.))
Power.dist.Z2.5 <- rep(NA, length(TEAM5$distance...p.Z1..m.))


for(i in 1:length(TEAM1$distance...p.Z1..m.)) Power.dist.Z1.1[i
   ] <- TEAM1$distance...p.Z1..m.[i]

for(i in 1:length(TEAM2$distance...p.Z1..m.)) Power.dist.Z1.2[i
   ] <- TEAM2$distance...p.Z1..m.[i] + TEAM2$distance...p.Z2..m
   .[i]

for(i in 1:length(TEAM3$distance...p.Z1..m.)) Power.dist.Z1.3[i
   ] <- TEAM3$distance...p.Z1..m.[i]

for(i in 1:length(TEAM4$distance...p.Z1..m.)) Power.dist.Z1.4[i
   ] <- TEAM4$distance...p.Z1..m.[i]

for(i in 1:length(TEAM5$distance...p.Z1..m.)) Power.dist.Z1.5[i
   ] <- TEAM5$distance...p.Z1..m.[i] + TEAM5$distance...p.Z2..m
   .[i]

#
```

```
for(i in 1:length(TEAM1$distance...p.Z1..m.)) Power.dist.Z2.1[i
   ] <- TEAM1$distance...p.Z2..m.[i] + TEAM1$distance...p.Z3..m
   .[i]

for(i in 1:length(TEAM2$distance...p.Z1..m.)) Power.dist.Z2.2[i
   ] <- TEAM2$distance...p.Z3..m.[i] + TEAM2$distance...p.Z4..m
   .[i]

for(i in 1:length(TEAM3$distance...p.Z1..m.)) Power.dist.Z2.3[i
   ] <- TEAM3$distance...p.Z2..m.[i] + TEAM3$distance...p.Z3..m
   .[i]

for(i in 1:length(TEAM4$distance...p.Z1..m.)) Power.dist.Z2.4[i
   ] <- TEAM4$distance...p.Z2..m.[i] + TEAM4$distance...p.Z3..m
   .[i]

for(i in 1:length(TEAM5$distance...p.Z1..m.)) Power.dist.Z2.5[i
   ] <- TEAM5$distance...p.Z3..m.[i] + TEAM5$distance...p.Z4..m
   .[i] + TEAM5$distance...p.Z5..m.[i] + TEAM5$distance...p.Z6
   ..m.[i]

TEAM1 <- cbind(TEAM1, Power.dist.Z1.1, Power.dist.Z2.1)
TEAM2 <- cbind(TEAM2, Power.dist.Z1.2, Power.dist.Z2.2)
TEAM3 <- cbind(TEAM3, Power.dist.Z1.3, Power.dist.Z2.3)
TEAM4 <- cbind(TEAM4, Power.dist.Z1.4, Power.dist.Z2.4)
TEAM5 <- cbind(TEAM5, Power.dist.Z1.5, Power.dist.Z2.5)

#########################
##SPEED

#DISTANCE: stiamo parlando di distanza, quindi metri (es. Z1
   =114.5)
#Ho gia fatto i conti a mano, riassumo il tutto in 3 zone
   quindi 2 soglie (0, 19.8-21, 25-25.2)

Speed.dist.Z1.1 <- rep(NA, length(TEAM1$distance...sp.Z1..m.))
```

```
Speed.dist.Z1.2 <- rep(NA, length(TEAM2$distance...sp.Z1..m.))
Speed.dist.Z1.3 <- rep(NA, length(TEAM3$distance...sp.Z1..m.))
Speed.dist.Z1.4 <- rep(NA, length(TEAM4$distance...sp.Z1..m.))
Speed.dist.Z1.5 <- rep(NA, length(TEAM5$distance...sp.Z1..m.))

Speed.dist.Z2.1 <- rep(NA, length(TEAM1$distance...sp.Z1..m.))
Speed.dist.Z2.2 <- rep(NA, length(TEAM2$distance...sp.Z1..m.))
Speed.dist.Z2.3 <- rep(NA, length(TEAM3$distance...sp.Z1..m.))
Speed.dist.Z2.4 <- rep(NA, length(TEAM4$distance...sp.Z1..m.))
Speed.dist.Z2.5 <- rep(NA, length(TEAM5$distance...sp.Z1..m.))

Speed.dist.Z3.1 <- rep(NA, length(TEAM1$distance...sp.Z1..m.))
Speed.dist.Z3.2 <- rep(NA, length(TEAM2$distance...sp.Z1..m.))
Speed.dist.Z3.3 <- rep(NA, length(TEAM3$distance...sp.Z1..m.))
Speed.dist.Z3.4 <- rep(NA, length(TEAM4$distance...sp.Z1..m.))
Speed.dist.Z3.5 <- rep(NA, length(TEAM5$distance...sp.Z1..m.))


for(i in 1:length(TEAM1$distance...sp.Z1..m.)) Speed.dist.Z1.1[
   i] <- TEAM1$distance...sp.Z1..m.[i] + TEAM1$distance...sp.Z2
   ..m.[i] + TEAM1$distance...sp.Z3..m.[i]

for(i in 1:length(TEAM2$distance...sp.Z1..m.)) Speed.dist.Z1.2[
   i] <- TEAM2$distance...sp.Z1..m.[i]

for(i in 1:length(TEAM3$distance...sp.Z1..m.)) Speed.dist.Z1.3[
   i] <- TEAM3$distance...sp.Z1..m.[i] + TEAM3$distance...sp.Z2
   ..m.[i]

for(i in 1:length(TEAM4$distance...sp.Z1..m.)) Speed.dist.Z1.4[
   i] <- TEAM4$distance...sp.Z1..m.[i] + TEAM4$distance...sp.Z2
   ..m.[i] + TEAM4$distance...sp.Z3..m.[i]

for(i in 1:length(TEAM5$distance...sp.Z1..m.)) Speed.dist.Z1.5[
   i] <- TEAM5$distance...sp.Z1..m.[i] + TEAM5$distance...sp.Z2
   ..m.[i] + TEAM5$distance...sp.Z3..m.[i] + TEAM5$distance...
```

```
   sp.Z4..m.[i]

#
for(i in 1:length(TEAM1$distance...sp.Z1..m.)) Speed.dist.Z2.1[
   i] <- TEAM1$distance...sp.Z4..m.[i]

for(i in 1:length(TEAM2$distance...sp.Z1..m.)) Speed.dist.Z2.2[
   i] <- TEAM2$distance...sp.Z2..m.[i]

for(i in 1:length(TEAM3$distance...sp.Z1..m.)) Speed.dist.Z2.3[
   i] <- TEAM3$distance...sp.Z3..m.[i]

for(i in 1:length(TEAM4$distance...sp.Z1..m.)) Speed.dist.Z2.4[
   i] <- TEAM4$distance...sp.Z4..m.[i]

for(i in 1:length(TEAM5$distance...sp.Z1..m.)) Speed.dist.Z2.5[
   i] <- TEAM5$distance...sp.Z5..m.[i]

#
for(i in 1:length(TEAM1$distance...sp.Z1..m.)) Speed.dist.Z3.1[
   i] <- TEAM1$distance...sp.Z5..m.[i]

for(i in 1:length(TEAM2$distance...sp.Z1..m.)) Speed.dist.Z3.2[
   i] <- TEAM2$distance...sp.Z3..m.[i]

for(i in 1:length(TEAM3$distance...sp.Z1..m.)) Speed.dist.Z3.3[
   i] <- TEAM3$distance...sp.Z4..m.[i]

for(i in 1:length(TEAM4$distance...sp.Z1..m.)) Speed.dist.Z3.4[
   i] <- TEAM4$distance...sp.Z5..m.[i]

for(i in 1:length(TEAM5$distance...sp.Z1..m.)) Speed.dist.Z3.5[
   i] <- TEAM5$distance...sp.Z6..m.[i]


TEAM1 <- cbind(TEAM1, Speed.dist.Z1.1, Speed.dist.Z2.1, Speed.
```

```
   dist.Z3.1)
TEAM2 <- cbind(TEAM2, Speed.dist.Z1.2, Speed.dist.Z2.2, Speed.
   dist.Z3.2)
TEAM3 <- cbind(TEAM3, Speed.dist.Z1.3, Speed.dist.Z2.3, Speed.
   dist.Z3.3)
TEAM4 <- cbind(TEAM4, Speed.dist.Z1.4, Speed.dist.Z2.4, Speed.
   dist.Z3.4)
TEAM5 <- cbind(TEAM5, Speed.dist.Z1.5, Speed.dist.Z2.5, Speed.
   dist.Z3.5)


#Ora devo togliere le variabili in piu
TEAM1_fin <- TEAM1[, -c
   (28,29,30,41,42,43,44,45,46,47,48,49,56,57,58,59,60)]
TEAM2_fin <- TEAM2[, -c
   (28,29,30,41,42,43,44,45,46,47,48,49,50,51,54,55,
   56,57,58,59,60,61)]
TEAM3_fin <- TEAM3[, -c
   (6,29,30,31,42,43,44,45,46,47,48,49,50,51,52,59,60,61)]
TEAM4_fin <- TEAM4[, -c
   (6,29,30,31,42,43,44,45,46,47,48,49,50,52,53,55,56)]
TEAM5_fin <- TEAM5[, -c
   (6,29,30,31,42,43,44,45,46,47,48,49,50,51,52,53,54,55,
   56,57,58,60,61,62,63,64,65,67,68,69,70,71,72,73,74,75,
   76,77,78,79,80,85,86,87,88,89,90)]


#Ora, devo cambiare i nomi delle colonne create da me prima, ho
    fatto cosi per non sovrascrivere i dati in precedenza
colnames(TEAM1_fin)[c(44:50)] <- c("MPE.dist.Z1", "MPE.dist.Z2"
   , "Power.dist.Z1", "Power.dist.Z2", "Speed.dist.Z1", "Speed.
   dist.Z2", "Speed.dist.Z3")
colnames(TEAM2_fin)[c(40:50)] <- c("duration.min", "total.min",
    "walk.min", "run.min", "MPE.dist.Z1", "MPE.dist.Z2", "Power
   .dist.Z1", "Power.dist.Z2", "Speed.dist.Z1", "Speed.dist.Z2"
   , "Speed.dist.Z3")
```

```
colnames(TEAM3_fin)[c(40:50)] <- c("duration.min", "total.min",
    "walk.min", "run.min", "MPE.dist.Z1", "MPE.dist.Z2", "Power
   .dist.Z1", "Power.dist.Z2", "Speed.dist.Z1", "Speed.dist.Z2"
   , "Speed.dist.Z3")
colnames(TEAM4_fin)[c(40:50)] <- c("duration.min", "total.min",
    "walk.min", "run.min", "MPE.dist.Z1", "MPE.dist.Z2", "Power
   .dist.Z1", "Power.dist.Z2", "Speed.dist.Z1", "Speed.dist.Z2"
   , "Speed.dist.Z3")
colnames(TEAM5_fin)[c(40:50)] <- c("duration.min", "total.min",
    "walk.min", "run.min", "MPE.dist.Z1", "MPE.dist.Z2", "Power
   .dist.Z1", "Power.dist.Z2", "Speed.dist.Z1", "Speed.dist.Z2"
   , "Speed.dist.Z3")
```

Listing C.3: Creazione dataset giocatori infortunati TEAM 1

```
#Creo un nuovo dataset con solo i giocatori che si sono
   infortunati, prendo tutte le sessioni di allenamento e
   partite fino al primo infortunio.

#TEAM 1
#Nel TEAM1 si infortuniano i PLAYER dal 01 al 13, ma il 13 non
   esiste. Inoltre il PLAYER7 si infortuna 2 volte.

TEAM1_fin[TEAM1_fin =="PLAYER1 *"] <- "PLAYER1"
TEAM1_fin[TEAM1_fin =="PLAYER2 *"] <- "PLAYER2"
TEAM1_fin[TEAM1_fin =="PLAYER3 *"] <- "PLAYER3"
TEAM1_fin[TEAM1_fin =="PLAYER4 *"] <- "PLAYER4"
TEAM1_fin[TEAM1_fin =="PLAYER5 *"] <- "PLAYER5"
TEAM1_fin[TEAM1_fin =="PLAYER6 *"] <- "PLAYER6"
TEAM1_fin[TEAM1_fin =="PLAYER7 *"] <- "PLAYER7"
TEAM1_fin[TEAM1_fin =="PLAYER8 *"] <- "PLAYER8"
TEAM1_fin[TEAM1_fin =="PLAYER9 *"] <- "PLAYER9"
TEAM1_fin[TEAM1_fin =="PLAYER10 *"] <- "PLAYER10"
TEAM1_fin[TEAM1_fin =="PLAYER11 *"] <- "PLAYER11"
TEAM1_fin[TEAM1_fin =="PLAYER12 *"] <- "PLAYER12"

player1 <- TEAM1_fin[TEAM1_fin$athlete=="PLAYER1",]
```

```
player2 <- TEAM1_fin[TEAM1_fin$athlete=="PLAYER2",]
player3 <- TEAM1_fin[TEAM1_fin$athlete=="PLAYER3",]
player4 <- TEAM1_fin[TEAM1_fin$athlete=="PLAYER4",]
player5 <- TEAM1_fin[TEAM1_fin$athlete=="PLAYER5",]
player6 <- TEAM1_fin[TEAM1_fin$athlete=="PLAYER6",]
player7 <- TEAM1_fin[TEAM1_fin$athlete=="PLAYER7",]
player8 <- TEAM1_fin[TEAM1_fin$athlete=="PLAYER8",]
player9 <- TEAM1_fin[TEAM1_fin$athlete=="PLAYER9",]
player10 <- TEAM1_fin[TEAM1_fin$athlete=="PLAYER10",]
player11 <- TEAM1_fin[TEAM1_fin$athlete=="PLAYER11",]
player12 <- TEAM1_fin[TEAM1_fin$athlete=="PLAYER12",]

#player1 fino al giorno dell'infortunio compreso
player1$date.time
player1$date.time[77] <- "2020-05-29 19:20:05"
player1$date.time

player1.tot <- player1

player1 <- player1[39:101,]


#player2 fino al giorno dell'infortunio compreso
player2$date.time
player2$date.time[46] <- "2020-05-29 18:34:23"
player2$date.time

player2.tot <- player2

player2 <- player2[6:70,]


#player3 fino al giorno dell'infortunio compreso
player3$date.time
player3$date.time[69] <- "2020-05-29 18:34:23"
player3$date.time
```

```
player3.tot <- player3

player3 <- player3[25:94,]


#player4 fino al giorno dell'infortunio compreso
player4$date.time

player4.tot <- player4

player4 <- player4[16:93,]


#player5 fino al giorno dell'infortunio compreso
player5$date.time
player5$date.time[65] <- "2020-05-29 19:20:05"
player5$date.time

player5.tot <- player5

player5 <- player5[2:93,]


#player6, si infortuna il giorno dopo l'ultima rilevazione con
   la pettorina
player6$date.time
player6$date.time[58] <- "2020-05-29 19:20:05"
player6$date.time

player6.tot <- player6


#player7 fino al giorno dell'infortunio compreso (c'eun secondo
    infortunio il 02/08)
player7$date.time
```

```
player7.tot <- player7

player7 <- player7[62:72,]


#player8 fino al giorno dell'infortunio compreso
player8$date.time
player8$date.time[74] <- "2020-05-29 18:34:23"
player8$date.time

player8.tot <- player8


#player9 fino al giorno dell'infortunio compreso
player9$date.time

player9.tot <- player9


#player10 fino al giorno dell'infortunio compreso
player10$date.time

player10.tot <- player10

player10 <- player10[13:76,]


#player11 fino al giorno dell'infortunio compreso
player11$date.time

player11.tot <- player11

player11 <- player11[67:70,]
```

```
#player12 fino al giorno dell'infortunio compreso
player12$date.time

player12.tot <- player12

player12 <- player12[57:65,]


team1_inf <- rbind(player1, player2, player3, player4, player5,
    player6, player7, player8, player9, player10, player11,
   player12)


#Tolgo le categorie che ripetono informazioni gia presenti in
   altre
table(team1_inf$category)
team1_inf <- team1_inf[-which(team1_inf$category=="EXERCISE"),]


#Verifico se ci sono ripetizioni

#devo togliere anche il "FRIENDLY MATCH" del 16/06
   perchecompreso nel FULL TRAINING
team1_inf <- team1_inf[-which(team1_inf$date.time=="2020-06-16
   19:22:20"),]

#e il 12/06 devo tolgiere l'allenamento piu breve
team1_inf <- team1_inf[-which(team1_inf$date.time=="2020-06-12
   18:41:51"),]

time_init <- rep(NA, nrow(team1_inf))

for(i in 1:(nrow(team1_inf)))
{
  time_init[i] = as.integer(substr(team1_inf$date.time[i
     ],12,13))*60*60 + as.integer(substr(team1_inf$date.time[i
```

```
     ],15,16))*60+as.integer(substr(team1_inf$date.time[i
     ],18,19))
}

team1_inf <- cbind(team1_inf, time_init)

for(i in 1:(nrow(team1_inf)-1))
{
    #if(substr(team1_inf$date.time[i],1,19) == substr(team1_inf
       $date.time[i+1],1,19)) team1_inf <- team1_inf[-i,]

    #if( (substr(team1_inf$date.time[i],1,13) == substr(team1_
       inf$date.time[i+1],1,13) )) team1_inf <- team1_inf[-i,]

    if((substr(team1_inf$date.time[i],1,10) == substr(team1_inf
       $date.time[i+1],1,10) ) & (team1_inf$time_init[i+1] >= (
       team1_inf$time_init[i]-2*60*60))) team1_inf <- team1_inf
       [-i,]
}

#ATTENZIONE: da mandare piu volte

team1_inf <- team1_inf[, -51]

#Uniformo la variabile "category"
team1_inf[team1_inf =="FRIENDLY MATCH"] <- "OFFICIAL MATCH"
team1_inf[team1_inf =="DIFFERENZIATO"] <- "FULL TRAINING"
```

C.2 Modelli

Listing C.4: Esempio di analisi esplorativa

```
team_inf.tot <- rbind(team1_inf, team2_inf, team3_inf, team4_
   inf, team5_inf)
```

```
colnames(team_inf.tot)[6] <- "distanza"

#a) Distanza percorsa totale in metri
summary(team_inf.tot$distanza)
head(team_inf.tot$distanza)

sd(team_inf.tot$distanza)

plot(density(team_inf.tot$distanza))

hist(team_inf.tot$distanza, nclass = 50, main="Distanza
    percorsa", xlab = "Metri", probability = T) #coda un po, piu
     lunga a destra
curve(dnorm(x, mean = mean(team_inf.tot$distanza), sd = sd(team
   _inf.tot$distanza)), add = TRUE, col = "red")
legend(10000, 0.00015, legend=c("Normale"), lty = 1, col=c("red
   "), cex = 0.85)

boxplot(team_inf.tot$distanza, ylab="Metri",
```

Listing C.5: Funzioni utili

```
// Gibbs sampler
// Main functions for quantile regression model

// Authors:
//           Bernardi Mauro, University of Padova
//           Last update: December 28, 2021

// List of implemented MCMC algorithms:

// [[Rcpp::depends(RcppArmadillo)]]
#include <RcppArmadillo.h>

// [[Rcpp::depends(RcppProgress)]]
#include "progress.hpp"
#include "progress_bar.hpp"
```

```
//#include "linreg_GS.h"

#define DOUBLE_EPS 2.220446e-16
#define ARMA_64BIT_WORD 1
#define SAFE_LOG(a) (((a) <= 0.0) ? log(DOUBLE_EPS) : log(a))
#define SAFE_ZERO(a) ((a) == 0 ? DOUBLE_EPS : (a))
#define SQRT_DOUBLE_EPS sqrt(DOUBLE_EPS)

using std::pow;
using std::exp;
using std::sqrt;
using std::log;

using namespace Rcpp;
using namespace arma;



/**
 * Extend division reminder to vectors
 *
 * @param   a       Dividend
 * @param   n       Divisor
 */

arma::uvec seq_default(long double from, long double to, long
   unsigned int by) {

  int length_out;

  length_out = floor(abs((double)to-(double)from+1.0)/(double)
     by);
  arma::uvec ret(length_out);
  ret.zeros();

  ret(0) = by;
```

```
  for (int j=1; j<length_out; j++) {
    ret(j) = ret(j-1) + by;
  }
  return ret-1;
}

// funzione per generare da Polya-Gamma(a,b)
// riprende la funzione da R
// [[Rcpp::export]]
arma::vec pgdrawC(int n, arma::vec b) {
  Environment pkg = Environment::namespace_env("pgdraw");
  Function f = pkg["pgdraw"];
  arma::vec omega(n);               omega.zeros();
  omega = as<arma::vec>(f(n, b));
  return omega;
}

// [[Rcpp::export]]
arma::vec rmvstdnormC(int n) {
  arma::vec x(n);
  x.randn();
  return x;
}

// [[Rcpp::export]]
arma::vec rmvnormC(arma::vec mean, arma::mat sigma) {
  arma::vec x(sigma.n_cols);
  x.randn();
  return mean + chol(sigma, "lower") * x;
}



// [[Rcpp::export]]
arma::vec binregp_update_rnd(arma::mat X,
                             arma::vec y,
```

```
                            arma::vec omega,
                            arma::mat mPrRegP_S_INV,
                            arma::vec vPrRegP_C) {

/* ::::::::::::::::::::::::::::::::::::::::::::::
 variable declaration                              */
int p = 0, n = 0;

/* ::::::::::::::::::::::::::::::::::::::::::::::
 get dimensions                                  */
p = X.n_cols;
n = X.n_rows;

/* ::::::::::::::::::::::::::::::::::::::::::::::
 vector and matrices declaration                   */
arma::mat XWX_(p, p);                XWX_.zeros();
arma::vec XTy_(p);                   XTy_.zeros();
arma::mat mSig(p, p);                mSig.zeros();
arma::mat mQ(p, p);                  mQ.zeros();
arma::mat mR(p, p);                  mR.zeros();
arma::vec b(p);                      b.zeros();
arma::vec mu_star(p);                mu_star.zeros();
arma::mat sigma_star(p, p);          sigma_star.zeros();
arma::vec out(p);                    out.zeros();
arma::vec bb_(p);                    bb_.zeros();
arma::mat XW2_(n, p);                XW2_.zeros();
arma::mat mSig_HALF(n+p, p);         mSig_HALF.zeros();
arma::vec z(p);                      z.zeros();

/* ::::::::::::::::::::::::::::::::::::::::::::::
 Get relevant quantities                       */
XTy_ = X.t() * (y - 0.5);
bb_  = mPrRegP_S_INV * vPrRegP_C;

/* ::::::::::::::::::::::::::::::::::::::::::::::
 Perform QR update of XWX + D                  */
```

```
  XW2_ = X % repmat(sqrt(omega), 1, p);
  XWX_ = XW2_.t() * XW2_;
  mSig = XWX_ + mPrRegP_S_INV;

  /* ::::::::::::::::::::::::::::::::::::::::
   Compute the inverse of Sig     */
  sigma_star = inv_sympd(mSig);
  mu_star    = sigma_star * (XTy_ + bb_);

  /* ::::::::::::::::::::::::::::::::::::::::::::::::
   get output                                        */
  z   = rmvstdnormC(p);
  out = mu_star + chol(sigma_star, "lower") * z;

  /* ::::::::::::::::::::::::::::::::::::::::::::::::
   return output                                     */
  return out;
}


// [[Rcpp::export]]
List binregp_update_rnd2(arma::mat X,
                         arma::vec y,
                         arma::vec omega,
                         arma::mat mPrRegP_S_INV,
                         arma::vec vPrRegP_C) {

  /* ::::::::::::::::::::::::::::::::::::::::::::::
   variable declaration                             */
  int p = 0, n = 0;
  List out;

  /* ::::::::::::::::::::::::::::::::::::::::::::::
   get dimensions                                 */
  p = X.n_cols;
  n = X.n_rows;
```

```
/* ::::::::::::::::::::::::::::::::::::::::::::
 vector and matrices declaration                    */
arma::mat XWX_(p, p);                XWX_.zeros();
arma::vec XTy_(p);                   XTy_.zeros();
arma::mat mSig(p, p);                mSig.zeros();
arma::mat mQ(p, p);                  mQ.zeros();
arma::mat mR(p, p);                  mR.zeros();
arma::vec b(p);                      b.zeros();
arma::vec mu_star(p);                mu_star.zeros();
arma::mat sigma_star(p, p);          sigma_star.zeros();
//arma::vec out(p);                    out.zeros();
arma::vec bb_(p);                    bb_.zeros();
arma::mat XW2_(n, p);                XW2_.zeros();
arma::mat mSig_HALF(n+p, p);         mSig_HALF.zeros();
arma::vec z(p);                      z.zeros();

/* ::::::::::::::::::::::::::::::::::::::::::::
 Get relevant quantities                      */
XTy_ = X.t() * (y - 0.5);
bb_  = mPrRegP_S_INV * vPrRegP_C;

/* ::::::::::::::::::::::::::::::::::::::::::::
 Perform QR update of XWX + D                 */
XW2_ = X % repmat(sqrt(omega), 1, p);
XWX_ = XW2_.t() * XW2_;
mSig = XWX_ + mPrRegP_S_INV;

/* ::::::::::::::::::::::::::::::::::::::::::::
 Compute the inverse of Sig     */
sigma_star = inv_sympd(mSig);
mu_star    = sigma_star * (XTy_ + bb_);

/* ::::::::::::::::::::::::::::::::::::::::::::
 get output                                     */
z   = rmvstdnormC(p);
```

```
  out = mu_star + chol(sigma_star, "lower") * z; /* qua non
     devo trasporre anchela std dev?*/

  /* ::::::::::::::::::::::::::::::::::::::::::::
   return output                                    */
  out["regpar"]     = mu_star + chol(sigma_star, "lower") * z;
  out["mu"]         = mu_star;
  out["sigma_star"] = chol(sigma_star, "lower");

  return out;
}



// [[Rcpp::export]]
arma::vec fast_binregp_update_rnd(arma::mat X,
                                  arma::vec y,
                                  arma::vec omega,
                                  arma::mat mPrRegP_S_INV,
                                  arma::mat mPrRegP_S_HALF_INV,
                                  arma::vec vPrRegP_C) {

  /* ::::::::::::::::::::::::::::::::::::::::::::
   variable declaration                             */
  int p = 0, n = 0;
  bool iFailure;

  /* ::::::::::::::::::::::::::::::::::::::::::::
   get dimensions                                */
  p = X.n_cols;
  n = X.n_rows;

  /* ::::::::::::::::::::::::::::::::::::::::::::
   vector and matrices declaration                  */
  arma::mat XWX_(p, p);                XWX_.zeros();
  arma::vec XTy_(p);                   XTy_.zeros();
```

```
arma::mat mSig(p, p);                mSig.zeros();
arma::mat mQ(p, p);                  mQ.zeros();
arma::mat mR(p, p);                  mR.zeros();
arma::vec b(p);                      b.zeros();
arma::vec mu_star(p);                mu_star.zeros();
arma::mat sigma_star(p, p);          sigma_star.zeros();
arma::vec out(p);                    out.zeros();
arma::vec bb_(p);                    bb_.zeros();
arma::mat XW2_(n, p);                XW2_.zeros();
arma::mat mSig_HALF(n+p, p);         mSig_HALF.zeros();
arma::vec z(p);                      z.zeros();

/* ::::::::::::::::::::::::::::::::::::::::
 Get relevant quantities                    */
XTy_ = X.t() * (y - 0.5);
bb_  = mPrRegP_S_INV * vPrRegP_C;

/* ::::::::::::::::::::::::::::::::::::::::
 Perform QR update of XWX + D               */
if (n > p) {
  /* ::::::::::::::::::::::::::::::::::::::::
   compute the matrix Sigma_HALF             */
  XW2_      = X % repmat(sqrt(omega), 1, p);
  mSig_HALF = join_vert(XW2_, mPrRegP_S_HALF_INV);

  /* ::::::::::::::::::::::::::::::::::::::::
   compute the QR decomposition of Sigma_HALF *
      /
  iFailure = qr_econ(mQ, mR, mSig_HALF);

  /* ::::::::::::::::::::::::::::::::::::::::
   get output                                */
  b   = solve((trimatu(mR)).t(), (XTy_ + bb_));
  z   = rmvstdnormC(p);
  out = solve(trimatu(mR), b + z);
} else {
```

```
    /* ::::::::::::::::::::::::::::::::::::::::::::::
     compute the matrix Sigma                */
    XW2_ = X % repmat(sqrt(omega), 1, p);
    XWX_ = XW2_.t() * XW2_;
    mSig = XWX_ + mPrRegP_S_INV;
    qr_econ(mQ, mR, mSig);
    b = mQ.t() * (XTy_ + bb_);

    /* ::::::::::::::::::::::::::::::::::::::::::::
     Compute the vector of regression parameters     */
    mu_star = solve(trimatu(mR), b);  // enable fast mode

    /* ::::::::::::::::::::::::::::::::::::::::::::
     Compute the inverse of Sig     */
    sigma_star = solve(trimatu(mR), mQ.t());

    /* ::::::::::::::::::::::::::::::::::::::::::::::::::::
     get output                                          */
    z   = rmvstdnormC(p);
    out = mu_star + chol(sigma_star, "lower") * z;
  }

  /* ::::::::::::::::::::::::::::::::::::::::::::::::::::
   return output                                       */
  return out;
}

// [[Rcpp::export]]
void regp_parms_init(arma::vec& param_init, int p,
                     arma::vec vPrRegP_C, arma::mat mPrRegP_S)
                        {

  /* ::::::::::::::::::::::::::::::::::::::::::::::::::::
    variable declaration                                 */
  arma::vec regp(p);                          regp.zeros();
```

```
  /* ::::::::::::::::::::::::::::::::::::::::::::::
    Initialize parameters                                    */
  if (accu(param_init) == 0.0) {
    Rcout<<">>> Initialize regression parameters and the scale:
        both drawn from the prior." << std::endl;
    regp = rmvnormC(vPrRegP_C, mPrRegP_S);
  } else {
    Rcout<<">>> Initialize regression parameters and the scale:
        supplied values." << std::endl;
    regp = param_init.subvec(0, p-1);
  }

  /* ::::::::::::::::::::::::::::::::::::::::::::::
   get output                                               */
  param_init.subvec(0, p-1) = regp;
}

// [[Rcpp::export]]
/* This function performs the Gibbs sampler for the linear
   quantile regression model
      with independent Normal-Inverse Gamma prior.
                                    */
List blogitreg_GS_C(arma::vec vY, arma::mat mX,
                    arma::vec param_init,
                    arma::vec vPrRegP_C,
                    arma::mat mPrRegP_S,
                    int nchains,
                    int dCrep,
                    int dBurnin,
                    int thin,
                    bool display_progress = true) {

  /* ::::::::::::::::::::::::::::::::::::::::::::::
    variable declaration                                     */
  int n, p, p_;
  List out, ret;
```

```
/* ::::::::::::::::::::::::::::::::::::::::::::::
 get dimensions                                    */
Rcout<<">>>
   ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
   " << std::endl;
Rcout<<">>> Bayesian inference for logistic linear regression
    model" << std::endl;
Rcout<<">>> Independent Gaussian, inverse gamma prior
   distribuition" << std::endl;
Rcout<<">>> Multiple parallel chains with n. chains "<<
   nchains << std::endl;

/* ::::::::::::::::::::::::::::::::::::::::::::::
 get dimensions                                    */
p  = mX.n_cols - 1;
n  = mX.n_rows;
p_ = param_init.size();

/* ::::::::::::::::::::::::::::::::::::::::::::::
 vector and matrices declaration                       */

/* generic quantities      */
arma::mat mPrRegP_S_INV(p+1, p+1);
                                mPrRegP_S_INV.zeros();
arma::vec omega(n);
                                                 omega.zeros();
arma::vec vRegP(p+1);
                                               vRegP.zeros();
arma::mat mPrRegP_S_HALF_INV(p+1, p+1);
                            mPrRegP_S_HALF_INV.zeros();
arma::vec param_init_(p_);
                                          param_init_.zeros();
arma::uvec thin_idx(floor((double)dCrep / (double)thin));
         thin_idx.zeros();
```

```
/* Output quantities      */
arma::mat mRegP_STORE(dCrep, p+1);
                                mRegP_STORE.zeros();
arma::cube cRegP_STORE(dCrep, p+1, nchains, arma::fill::randu
   );
arma::mat mRegP_INIT(nchains, p+1);
                                mRegP_INIT.zeros();

/* ::::::::::::::::::::::::::::::::::::::::
 Get relevant quantities                       */
mPrRegP_S_INV      = arma::inv_sympd(mPrRegP_S);
mPrRegP_S_HALF_INV = sqrtmat_sympd(mPrRegP_S_INV);
param_init_        = param_init;
if (thin <= 0) {
  thin = 1;
} else {
  thin_idx = seq_default(1, dCrep, thin);
}

/* ::::::::::::::::::::::::::::::::::::::::::::
 Burn the Gibbs sampling iterative procedure
                      */
for (int chain = 0; chain < nchains; chain++) {

  /* ::::::::::::::::::::::::::::::::::::::::
   Initialize the chain                          */
  param_init = param_init_;

  /* ::::::::::::::::::::::::::::::::::::::::
   Starting parameters                          */
  regp_parms_init(param_init, p+1, vPrRegP_C, mPrRegP_S); /*
     ok, in base ai valori mi dice che tipo di parametri
     iniziali ho*/

  /* ::::::::::::::::::::::::::::::::::::::::
   Get initial parameters                        */
```

```
  vRegP = param_init.subvec(0, p);

  /* ::::::::::::::::::::::::::::::::::::::::::::
   Print to screen                                 */
  Rcout<<">>> ==== <<< Burn-In"           << std::endl;
  Rcout<<">>> ==== <<< Running chain  : " << chain+1 << std::
     endl;

  /* ::::::::::::::::::::::::::::::::::::::::::::
   Add the Progress bar                          */
  Progress prg_bar(dBurnin, display_progress);

  for (int iter = 0; iter < dBurnin; iter++) {

    /* ::::::::::::::::::::::::::::::::::::::::::::
     Display the Progress bar                      */
    prg_bar.increment();

    /* ::::::::::::::::::::::::::::::::::::::::::::
     full conditional (omega)                      */
    omega = pgdrawC(1, mX * vRegP); //posteriori PG

    /* ::::::::::::::::::::::::::::::::::::::::::::
     full conditional (beta)                       */
    //vRegP = fast_binregp_update_rnd(mX, vY, omega, mPrRegP_
       S_INV, mPrRegP_S_HALF_INV, vPrRegP_C);
    vRegP = binregp_update_rnd(mX, vY, omega, mPrRegP_S_INV,
       vPrRegP_C);
  }
  /* ::::::::::::::::::::::::::::::::::::::::::::
   collecting results: */
  mRegP_INIT.row(chain) = vRegP.t();
}

/* ::::::::::::::::::::::::::::::::::::::::::::
 Gibbs sampling iterative procedure              */
```

```
for (int chain = 0; chain < nchains; chain++) {

  /* ::::::::::::::::::::::::::::::::::::::::
   Initialize the chain                        */
  mRegP_STORE.zeros();

  /* ::::::::::::::::::::::::::::::::::::::::
   Get initial parameters                      */
  vRegP = mRegP_INIT.row(chain).t();

  /* ::::::::::::::::::::::::::::::::::::::::
   Print to screen                             */
  Rcout<<"
      ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::" <<
       std::endl;
  Rcout<<">>> ==== <<< Gibbs sampling algorithm" << std::endl;
  Rcout<<">>> ==== <<< Running chain  : " << chain+1 << std::
     endl;

  /* ::::::::::::::::::::::::::::::::::::::::
   Add the Progress bar                        */
  Progress prg_bar(dCrep, display_progress);

  for (int iter = 0; iter < dCrep; iter++) {

    /* ::::::::::::::::::::::::::::::::::::::::
     Display the Progress bar                  */
    prg_bar.increment();

    /* ::::::::::::::::::::::::::::::::::::::::
     full conditional (omega)                  */
    omega = pgdrawC(1, mX * vRegP); //posteriori PG

    /* ::::::::::::::::::::::::::::::::::::::::
     full conditional (beta)                   */
    //vRegP = fast_binregp_update_rnd(mX, vY, omega, mPrRegP_
```

```
        S_INV, mPrRegP_S_HALF_INV, vPrRegP_C);
      vRegP = binregp_update_rnd(mX, vY, omega, mPrRegP_S_INV,
        vPrRegP_C);

      /* ::::::::::::::::::::::::::::::::::::::::::::
       collecting results                              */
      mRegP_STORE.row(iter) = vRegP.t();
    }
    /* ::::::::::::::::::::::::::::::::::::::::::::
     collecting results: multiple chains */
    if (nchains > 1) {
      cRegP_STORE.slice(chain) = mRegP_STORE;
    }
  }

  /* ::::::::::::::::::::::::::::::::::::::::::::
   get output                                          */
  if (nchains == 1) {
    if (thin == 1) {
      out["regpar"]  = mRegP_STORE;
    } else {
      arma::mat mRegP_STORE_(thin_idx.n_elem, p+1);   mRegP_
        STORE_.zeros();
      int id = 0;
      for (int rt = 0; rt<thin_idx.n_elem; rt++) {
        id                    = thin_idx(rt);
        mRegP_STORE_.row(rt) = mRegP_STORE.row(id);
      }
      out["regpar"] = mRegP_STORE_;
    }
  } else {
    if (thin == 1) {
      out["regpar"]  = cRegP_STORE;
    } else {
      arma::cube cRegP_STORE_(thin_idx.n_elem, p+1, nchains);
           cRegP_STORE_.zeros();
```

```
      int id = 0;
      for (int rt = 0; rt<thin_idx.n_elem; rt++) {
        id                    = thin_idx(rt);
        cRegP_STORE_.row(rt) = cRegP_STORE.row(id);
      }
      out["regpar"] = cRegP_STORE_;
    }
  }

  /* ::::::::::::::::::::::::::::::::::::::::::::
   return output                                    */
  return out;
}



model.sim.mix.fin <- function(beta, N, p=1, parametri) {

  X.all <- NULL
  y.all <- NULL

  require(stats)
  for (it in 1:N) {
    resp <- 0
    y    <- NULL
    X    <- NULL
    while (resp == 0) {
      y    <- c(y, resp)
      x_   <- c(1, rnorm(p, parametri[1], parametri[2]), rnorm(
         p, parametri[3], parametri[4]), rnorm(p, parametri[5],
          parametri[6]), rexp(p, parametri[7]), rexp(p,
         parametri[8]), rexp(p, parametri[9]), rexp(p,
         parametri[10]))
      prob <- exp(x_ %*% beta) / (1.0 + exp(x_ %*% beta))
      resp <- sample(c(0, 1), 1, prob = c(1.0 - prob, prob))
      X    <- rbind(X, x_)
```

```
    }
    y           <- c(y[-1], 1)
    rownames(X) <- NULL
    y.all       <- c(y.all, y)
    X.all       <- rbind(X.all, X)
  }
  # get output
  ret   <- NULL
  ret$X <- X.all
  ret$y <- y.all
  ret$n <- c(which(ret$y == 1)[1], diff(which(ret$y == 1)))

  # return output
  return(ret)
}
```

Bibliografia

Renato Andrade, Eirik Halvorsen Wik, Alexandre Rebelo-Marques, Peter Blanch, Rodney Whiteley, João Espregueira-Mendes, and Tim J Gabbett. Is the acute: chronic workload ratio (acwr) associated with risk of time-loss injury in professional team sports? a systematic review of methodology, variables and injury risk in practical situations. *Sports medicine*, 50(9):1613–1635, 2020.

Pitre C Bourdon, Marco Cardinale, Andrew Murray, Paul Gastin, Michael Kellmann, Matthew C Varley, Tim J Gabbett, Aaron J Coutts, Darren J Burgess, Warren Gregson, et al. Monitoring athlete training loads: consensus statement. *International journal of sports physiology and performance*, 12(s2):S2–161, 2017.

Nicholas G Polson, James G Scott, and Jesse Windle. Bayesian inference for logistic models using pólya–gamma latent variables. *Journal of the American statistical Association*, 108(504):1339–1349, 2013.

Gian Nicola Bisciotti, Cristiano Eirale, Alessandro Corsini, Christophe Baudot, Gerard Saillant, and Hakim Chalabi. Return to football training and competition after lockdown caused by the covid-19 pandemic: medical recommendations. *Biology of Sport*, 37(3):313, 2020.

Jan Ekstrand, Martin Hägglund, and Markus Waldén. Injury incidence and injury patterns in professional football: the uefa injury study. *British journal of sports medicine*, 45(7):553–558, 2011.

Alessio Rossi, Luca Pappalardo, Paolo Cintia, F Marcello Iaia, Javier Fernández, and Daniel Medina. Effective injury forecasting in soccer with gps training data and machine learning. *PloS one*, 13(7):e0201264, 2018.

Laura Bowen, Aleksander Stefan Gross, Mo Gimpel, and François-Xavier Li. Accumulated workloads and the acute: chronic workload ratio relate to injury risk in elite youth football players. *British journal of sports medicine*, 51(5):452–459, 2017.

Martin Hägglund, Markus Waldén, Henrik Magnusson, Karolina Kristenson, Håkan Bengtsson, and Jan Ekstrand. Injuries affect team performance negatively in professional football: an 11-year follow-up of the uefa champions league injury study. *British journal of sports medicine*, 47(12):738–742, 2013.

Olivia A Hurley. Impact of player injuries on teams' mental states, and subsequent performances, at the rugby world cup 2015. *Frontiers in psychology*, 7:807, 2016.

Erik E Lehmann and Günther G Schulze. What does it take to be a star?-the role of performance and the media for german soccer players. *Applied Economics Quarterly*, 54(1):59, 2008.

I Fernández-Cuevas, PM Gómez-Carmona, M Sillero-Quintana, J Noya-Salces, Javier Arnaiz-Lastras, and A Pastor-Barrón. Economic costs estimation of soccer injuries in first and second spanish division professional teams. In *15th Annual Congress of the European College of Sport Sciences ECSS, 23th 26th june*, 2010.

Fabian E Ehrmann, Craig S Duncan, Doungkamol Sindhusake, William N Franzsen, and David A Greene. Gps and injury prevention in professional soccer. *The Journal of Strength & Conditioning Research*, 30(2):360–367, 2016.

Hans Selye. The general-adaptation-syndrome. *Annual review of medicine*, 2(1): 327–342, 1951.

Tim J Gabbett. The trainingâĂŤinjury prevention paradox: should athletes be training smarter and harder? *British journal of sports medicine*, 50(5):273–280, 2016.

Shane Malone, Adam Owen, Bruno Mendes, Brian Hughes, Kieran Collins, and Tim J Gabbett. High-speed running and sprinting as an injury risk factor in soccer: Can well-developed physical qualities reduce the risk? *Journal of science and medicine in sport*, 21(3):257–262, 2018.

Laura Bowen, Aleksander Stephan Gross, Mo Gimpel, Stewart Bruce-Low, and Francois-Xavier Li. Spikes in acute: chronic workload ratio (acwr) associated with a 5–7 times greater injury rate in english premier league football players: a comprehensive 3-year study. *British journal of sports medicine*, 54(12):731–738, 2020.

Jiri Kirkendall, Donald T e Dvorak. Effective injury prevention in soccer. *The physician and sportsmedicine*, 38(1):147–157, 2010.

Alejandro López-Valenciano, Francisco Ayala, José Miguel Puerta, Mark De Ste Croix, Francisco Vera-García, Sergio Hernández-Sánchez, Iñaki Ruiz-Pérez, and Gregory Myer. A preventive model for muscle injuries: a novel approach based on learning algorithms. *Medicine and science in sports and exercise*, 50(5):915, 2018.

George EP Box and George C Tiao. *Bayesian inference in statistical analysis*, volume 40. John Wiley & Sons, 2011.

Walter R Gilks, Sylvia Richardson, and David Spiegelhalter. *Markov chain Monte Carlo in practice*. CRC press, 1995.

W. K. Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109, 1970. ISSN 00063444. URL http://www.jstor.org/stable/2334940.

Stuart Geman and Donald Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on pattern analysis and machine intelligence*, (6):721–741, 1984.

Donald B Gelman, Andrew Rubin. Inference from iterative simulation using multiple sequences. *Statistical science*, 7(4):457–472, 1992.

Ringraziamenti

Volevo ringraziare innanzitutto il prof. Bernardi, la prof. Cattelan e Mattia per il supporto tecnico-tattico datomi nel scrivere questa tesi. Ringrazio anche l'azienda Exelio per avermi concesso i dati da analizzare per produrre questo elaborato.

I miei più grandi ringraziamenti vanno a mamma e papà che mi hanno permesso di poter compiere questo percorso di studi e di poter anche godere della vita da fuori sede. Ora son ca...voli, devo essere sincero. Un grande grazie va anche detto a mio fratello, Nicolò, spero che l'accento sia dalla parte giusta, alla zia, ai nonni e perchè no, anche allo zio che mi ha dato sostentamento eno(gastro)nomico.

Un immenso abbraccio va anche agli amici, sia quelli dell'infanzia, sia quelli più recenti dell'università e del coinquilinaggio che mi hanno sempre fatto sentire voluto bene, nonostante le distanze, il COVID, i cambi di facoltà e di città. Un saluto anche alla mia squadra di calcio di Padova, La Mercenaria, e a quel posto che tante volte do per scontato che esista, ma che ogni tanto mi dimentico quanto sia importante per me, Vernasso.

Infine, un grande grazie va detto a Carlotta, persona fantastica con cui ho passato gli ultimi mesi, ti auguro il meglio.

Sono le 10.40 del 3 marzo 2022, sto piangendo.