Injury Prediction for Competitive Runners

Pierina Logrono

Caitlin Rookey

Final IST 708

2023-03-24



# Introduction

The most common injuries faced by runners are overuse injuries. These injuries develop over time due to issues like ramping up mileage too quickly, improper running form, not giving one’s body adequate time to recover, having tight or weak muscles, and wearing improper footwear. According to an article in houstonmethodist.org, some ways to prevent overuse injuries are to have a plan and progress slowly, avoid the increase of speed and distance simultaneously, give muscles time to recover, and make time for cross-training.1 How does one track the distance one runs daily, how much time one spends on a workout routine, or even how fast one runs? Fortunately, thanks to technological advances, some gadgets can track all the above and more, including how much sleep a person gets overnight.[[1]](#footnote-1)

The invention of the pedometer dates back to the 1770’s when a Swiss horologist by the name of Abraham-Louis Perrelet created it. Arguably, it’s also been suggested that it was Thomas Jefferson who produced the first mechanical fitness tracker. While it is unclear as to who came up with the idea, it is obvious that the technology for the pedometer progressed over the centuries. In 1921, the polygraph was invented as the first machine to have sensors measuring galvanic skin responses, pulse rate and blood pressure. With these physiological indicators, the machine detected if a person was lying. These machines were mainly used by police.[[2]](#footnote-2)

In 1965, the first modern pedometer fitness tracker was invented by Dr.Yoshiro Hatano, a Japanese professor at the Kyushu University of Health and Welfare. The Manpo-kei, or as translated in English “the 10,000-step meter”, was developed to combat obesity. Dr. Hatano speculated that with 10,000 steps, a proper balance of caloric intake and exercise to expend calories, one can maintain a healthy body. This pedometer was a simple gadget that was worn around the waste and had the ability to calculate the number of steps the individual walked. The Manpo-kei sold quickly as people started tracking their number of steps each day.2

From there, the pedometer’s technology evolved. In the 1970’s, Accelerometers were developed with the ability to detect a moving object’s motion and were used in the development of guided missile technologies. They were also used in vehicles to detect sharp deceleration and deploy airbags in crashes to help minimize injuries to drivers and passengers. In 1982, the Polar Sports Tester PE2000 came into the market essentially as a combination of an electrocardiogram and a radio chest strap to track athletic activities. A few years later, the PE3000 was released as the first watch with the capabilities of displaying biometric information live on its display. Then in 1987, the PE3000 was upgraded to introduce Target zones for the athletes’ heart rate-based training.2

In 1996, GPS Tracking became available for civilian usage (before only available for military use) and were used in fitness trackers to give a person access to mapping of their exercise routine. In 2006, the Nokia developed their motion sensing device (Nokia 5500 Sport) that had a built-in accelerometer with the ability to detect three different planes of motion; front to back, up and down, and side-to-side. In 2008-2009, Fitbit came into the market with their clip on motion detector to track the wearer’s movements, sleep, and calorie burn. From there, Fitbit introduced a wide-spread of activity tracking devices with the capability of connecting to smartphones with features such as Swim proof and swim tracking, reminders to move, and SmartTrack Automatic Exercise Recognition.2

However, with all the advancements in technology for tracking devices, overuse injuries still exist. Is tracking one’s daily routine with one of these gadgets enough? Does sentiment also play a critical role in preventing overuse injury?

# Analysis and Models

# About the Data

This dataset was downloaded from <https://www.kaggle.com/datasets/shashwatwork/injury-prediction-for-competitive-runners>

For this analysis, we are using the daily approach as this approach gave more accurate results in the original experiment. The dimensions for this dataset: 42,766 records and 73 attributes All attributes are numeric except for Athlete ID, injury, and date.

Data format: Each day consists of 10 attributes: Number of sessions [0, 2]: training sessions athlete completed that day.

Total km [0, 25]: total distance ran by athlete.

Km.Z3.4[0, 15]: Sum of the distance ran by athlete in heart rate zones 3 and 4. Km.Z5.T1.T2[0, 10]: Sum of distance ran by athlete during anaerobic zone in high intensity track intervals.

Km. sprinting [0, 1.5]: Total distance sprinting

Strength training [0, 1]: 1 represents if athlete performed some strength training; 0 represents no strength training.

Hours.alternative [0, 3]: Hours spent doing alternative training. Alternative training can be any cross-training such as cycling or swimming.

Perceived.exertion [-0.01, 1]: This is a subjective attribute indicating how exhausted the athlete felt upon completion of the training.

Perceived.trainingSuccess [-0.01, 1]: This is a subjective attribute indicating how well the athlete thought the training went.

Perceived.recovery [-0.01, 1]: This is a subjective attribute indicating how well the athlete felt before the start of the session.

Date: The date attribute is format by the day number within the 7-year period

Athlete ID: The Athlete ID is not a unique value due to the format of the dataset, where each record is a day within the 7-year period, pertaining to each athlete. There is a total of 77 athletes, of whom 27 are women and 50 are men. Injury [0, 1]: This attribute indicates whether there was an injury event on this day, for this athlete (1), or whether there was no injury (0).

Before running models on the data, the following steps were performed: Identified variables that needed to be factorized Identified the number of injuries within the dataset vs. non-injuries Visualized all variables and distributions Created samples of the dataset to make up for the imbalance

##Load all necessary libraries

library(tm)

## Warning: package 'tm' was built under R version 4.2.2

## Loading required package: NLP

library(stringr)

## Warning: package 'stringr' was built under R version 4.2.2

library(wordcloud)

## Warning: package 'wordcloud' was built under R version 4.2.2

## Loading required package: RColorBrewer

library(stringi)

## Warning: package 'stringi' was built under R version 4.2.2

library(Matrix)

## Warning: package 'Matrix' was built under R version 4.2.2

library(tidytext)

## Warning: package 'tidytext' was built under R version 4.2.2

library(rpart)

## Warning: package 'rpart' was built under R version 4.2.2

library(rpart.plot)

## Warning: package 'rpart.plot' was built under R version 4.2.2

library(RColorBrewer)  
library(readr)  
library(ggplot2)

## Warning: package 'ggplot2' was built under R version 4.2.2

##   
## Attaching package: 'ggplot2'

## The following object is masked from 'package:NLP':  
##   
## annotate

library(cluster)  
library(factoextra)

## Warning: package 'factoextra' was built under R version 4.2.2

## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa

library(dendextend)

## Warning: package 'dendextend' was built under R version 4.2.2

##   
## ---------------------  
## Welcome to dendextend version 1.16.0  
## Type citation('dendextend') for how to cite the package.  
##   
## Type browseVignettes(package = 'dendextend') for the package vignette.  
## The github page is: https://github.com/talgalili/dendextend/  
##   
## Suggestions and bug-reports can be submitted at: https://github.com/talgalili/dendextend/issues  
## You may ask questions at stackoverflow, use the r and dendextend tags:   
## https://stackoverflow.com/questions/tagged/dendextend  
##   
## To suppress this message use: suppressPackageStartupMessages(library(dendextend))  
## ---------------------

##   
## Attaching package: 'dendextend'

## The following object is masked from 'package:rpart':  
##   
## prune

## The following object is masked from 'package:stats':  
##   
## cutree

library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library(arules)

## Warning: package 'arules' was built under R version 4.2.2

##   
## Attaching package: 'arules'

## The following object is masked from 'package:dplyr':  
##   
## recode

## The following object is masked from 'package:tm':  
##   
## inspect

## The following objects are masked from 'package:base':  
##   
## abbreviate, write

library(arulesViz)

## Warning: package 'arulesViz' was built under R version 4.2.2

library(randomForest)

## Warning: package 'randomForest' was built under R version 4.2.2

## randomForest 4.7-1.1

## Type rfNews() to see new features/changes/bug fixes.

##   
## Attaching package: 'randomForest'

## The following object is masked from 'package:dplyr':  
##   
## combine

## The following object is masked from 'package:ggplot2':  
##   
## margin

library(randomForestExplainer)

## Warning: package 'randomForestExplainer' was built under R version 4.2.2

## Registered S3 method overwritten by 'GGally':  
## method from   
## +.gg ggplot2

library(naivebayes)

## Warning: package 'naivebayes' was built under R version 4.2.2

## naivebayes 0.9.7 loaded

library(e1071)

## Warning: package 'e1071' was built under R version 4.2.2

library(caret)

## Warning: package 'caret' was built under R version 4.2.2

## Loading required package: lattice

library(class)  
library(lattice)  
library(party)

## Warning: package 'party' was built under R version 4.2.2

## Loading required package: grid

## Loading required package: mvtnorm

## Loading required package: modeltools

## Loading required package: stats4

##   
## Attaching package: 'modeltools'

## The following object is masked from 'package:arules':  
##   
## info

## Loading required package: strucchange

## Warning: package 'strucchange' was built under R version 4.2.2

## Loading required package: zoo

## Warning: package 'zoo' was built under R version 4.2.2

##   
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':  
##   
## as.Date, as.Date.numeric

## Loading required package: sandwich

## Warning: package 'sandwich' was built under R version 4.2.2

##   
## Attaching package: 'strucchange'

## The following object is masked from 'package:stringr':  
##   
## boundary

library(tidyr)

## Warning: package 'tidyr' was built under R version 4.2.2

##   
## Attaching package: 'tidyr'

## The following objects are masked from 'package:Matrix':  
##   
## expand, pack, unpack

library(purrr)

## Warning: package 'purrr' was built under R version 4.2.2

##   
## Attaching package: 'purrr'

## The following object is masked from 'package:caret':  
##   
## lift

library(corrplot)

## Warning: package 'corrplot' was built under R version 4.2.2

## corrplot 0.92 loaded

library(knitr)  
library(tidyverse)

## ── Attaching packages  
## ───────────────────────────────────────  
## tidyverse 1.3.2 ──

## ✔ tibble 3.1.8 ✔ forcats 0.5.2  
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ ggplot2::annotate() masks NLP::annotate()  
## ✖ strucchange::boundary() masks stringr::boundary()  
## ✖ randomForest::combine() masks dplyr::combine()  
## ✖ tidyr::expand() masks Matrix::expand()  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ✖ purrr::lift() masks caret::lift()  
## ✖ randomForest::margin() masks ggplot2::margin()  
## ✖ tidyr::pack() masks Matrix::pack()  
## ✖ arules::recode() masks dplyr::recode()  
## ✖ tidyr::unpack() masks Matrix::unpack()

library(gmodels)

## Warning: package 'gmodels' was built under R version 4.2.3

library(rattle)

## Warning: package 'rattle' was built under R version 4.2.2

## Loading required package: bitops  
##   
## Attaching package: 'bitops'  
##   
## The following object is masked from 'package:Matrix':  
##   
## %&%  
##   
## Rattle: A free graphical interface for data science with R.  
## Version 5.5.1 Copyright (c) 2006-2021 Togaware Pty Ltd.  
## Type 'rattle()' to shake, rattle, and roll your data.  
##   
## Attaching package: 'rattle'  
##   
## The following object is masked from 'package:randomForest':  
##   
## importance

# EDA

# Load the data

daily <- read.csv("C:/Users/benro/OneDrive/Documents/IST707/Final/day\_approach\_maskedID\_timeseries.csv")

# Check the dimensions of our dataset. Results in 42,766 records, 73 attributes.

dim(daily)

## [1] 42766 73

Note that the echo = FALSE parameter was added to the code chunk to prevent printing of the R code that generated the plot.

Checking the structure of the data. There are 42,766 observations with 73 attributes.

str(daily)

## 'data.frame': 42766 obs. of 73 variables:  
## $ nr..sessions : num 1 0 1 0 1 1 1 1 0 1 ...  
## $ total.km : num 5.8 0 0 0 0 16.4 0 5.2 0 0 ...  
## $ km.Z3.4 : num 0 0 0 0 0 10 0 0 0 0 ...  
## $ km.Z5.T1.T2 : num 0.6 0 0 0 0 0 0 0.5 0 0 ...  
## $ km.sprinting : num 1.2 0 0 0 0 0 0 1.2 0 0 ...  
## $ strength.training : num 0 0 1 0 0 1 0 0 0 1 ...  
## $ hours.alternative : num 0 0 0 0 1.08 0 1 0 0 0 ...  
## $ perceived.exertion : num 0.11 -0.01 0.1 -0.01 0.08 0.11 0.1 0.1 -0.01 0.1 ...  
## $ perceived.trainingSuccess : num 0 -0.01 0 -0.01 0 0 0 0 -0.01 0 ...  
## $ perceived.recovery : num 0.18 -0.01 0.17 -0.01 0.18 0.17 0.15 0.17 -0.01 0.17 ...  
## $ nr..sessions.1 : num 0 1 0 1 1 1 1 0 1 1 ...  
## $ total.km.1 : num 0 0 0 0 16.4 0 5.2 0 0 17.6 ...  
## $ km.Z3.4.1 : num 0 0 0 0 10 0 0 0 0 7.2 ...  
## $ km.Z5.T1.T2.1 : num 0 0 0 0 0 0 0.5 0 0 0 ...  
## $ km.sprinting.1 : num 0 0 0 0 0 0 1.2 0 0 0 ...  
## $ strength.training.1 : num 0 1 0 0 1 0 0 0 1 0 ...  
## $ hours.alternative.1 : num 0 0 0 1.08 0 1 0 0 0 0 ...  
## $ perceived.exertion.1 : num -0.01 0.1 -0.01 0.08 0.11 0.1 0.1 -0.01 0.1 0.11 ...  
## $ perceived.trainingSuccess.1: num -0.01 0 -0.01 0 0 0 0 -0.01 0 0 ...  
## $ perceived.recovery.1 : num -0.01 0.17 -0.01 0.18 0.17 0.15 0.17 -0.01 0.17 0.17 ...  
## $ nr..sessions.2 : num 1 0 1 1 1 1 0 1 1 1 ...  
## $ total.km.2 : num 0 0 0 16.4 0 5.2 0 0 17.6 0 ...  
## $ km.Z3.4.2 : num 0 0 0 10 0 0 0 0 7.2 0 ...  
## $ km.Z5.T1.T2.2 : num 0 0 0 0 0 0.5 0 0 0 0 ...  
## $ km.sprinting.2 : num 0 0 0 0 0 1.2 0 0 0 0 ...  
## $ strength.training.2 : num 1 0 0 1 0 0 0 1 0 0 ...  
## $ hours.alternative.2 : num 0 0 1.08 0 1 0 0 0 0 1 ...  
## $ perceived.exertion.2 : num 0.1 -0.01 0.08 0.11 0.1 0.1 -0.01 0.1 0.11 0.08 ...  
## $ perceived.trainingSuccess.2: num 0 -0.01 0 0 0 0 -0.01 0 0 0 ...  
## $ perceived.recovery.2 : num 0.17 -0.01 0.18 0.17 0.15 0.17 -0.01 0.17 0.17 0.17 ...  
## $ nr..sessions.3 : num 0 1 1 1 1 0 1 1 1 1 ...  
## $ total.km.3 : num 0 0 16.4 0 5.2 0 0 17.6 0 10.5 ...  
## $ km.Z3.4.3 : num 0 0 10 0 0 0 0 7.2 0 6.5 ...  
## $ km.Z5.T1.T2.3 : num 0 0 0 0 0.5 0 0 0 0 0 ...  
## $ km.sprinting.3 : num 0 0 0 0 1.2 0 0 0 0 0 ...  
## $ strength.training.3 : num 0 0 1 0 0 0 1 0 0 0 ...  
## $ hours.alternative.3 : num 0 1.08 0 1 0 0 0 0 1 0 ...  
## $ perceived.exertion.3 : num -0.01 0.08 0.11 0.1 0.1 -0.01 0.1 0.11 0.08 0.09 ...  
## $ perceived.trainingSuccess.3: num -0.01 0 0 0 0 -0.01 0 0 0 0 ...  
## $ perceived.recovery.3 : num -0.01 0.18 0.17 0.15 0.17 -0.01 0.17 0.17 0.17 0.18 ...  
## $ nr..sessions.4 : num 1 1 1 1 0 1 1 1 1 0 ...  
## $ total.km.4 : num 0 16.4 0 5.2 0 0 17.6 0 10.5 0 ...  
## $ km.Z3.4.4 : num 0 10 0 0 0 0 7.2 0 6.5 0 ...  
## $ km.Z5.T1.T2.4 : num 0 0 0 0.5 0 0 0 0 0 0 ...  
## $ km.sprinting.4 : num 0 0 0 1.2 0 0 0 0 0 0 ...  
## $ strength.training.4 : num 0 1 0 0 0 1 0 0 0 0 ...  
## $ hours.alternative.4 : num 1.08 0 1 0 0 0 0 1 0 0 ...  
## $ perceived.exertion.4 : num 0.08 0.11 0.1 0.1 -0.01 0.1 0.11 0.08 0.09 -0.01 ...  
## $ perceived.trainingSuccess.4: num 0 0 0 0 -0.01 0 0 0 0 -0.01 ...  
## $ perceived.recovery.4 : num 0.18 0.17 0.15 0.17 -0.01 0.17 0.17 0.17 0.18 -0.01 ...  
## $ nr..sessions.5 : num 1 1 1 0 1 1 1 1 0 1 ...  
## $ total.km.5 : num 16.4 0 5.2 0 0 17.6 0 10.5 0 4.8 ...  
## $ km.Z3.4.5 : num 10 0 0 0 0 7.2 0 6.5 0 0.6 ...  
## $ km.Z5.T1.T2.5 : num 0 0 0.5 0 0 0 0 0 0 0 ...  
## $ km.sprinting.5 : num 0 0 1.2 0 0 0 0 0 0 1.2 ...  
## $ strength.training.5 : num 1 0 0 0 1 0 0 0 0 0 ...  
## $ hours.alternative.5 : num 0 1 0 0 0 0 1 0 0 0 ...  
## $ perceived.exertion.5 : num 0.11 0.1 0.1 -0.01 0.1 0.11 0.08 0.09 -0.01 0.11 ...  
## $ perceived.trainingSuccess.5: num 0 0 0 -0.01 0 0 0 0 -0.01 0 ...  
## $ perceived.recovery.5 : num 0.17 0.15 0.17 -0.01 0.17 0.17 0.17 0.18 -0.01 0.18 ...  
## $ nr..sessions.6 : num 1 1 0 1 1 1 1 0 1 0 ...  
## $ total.km.6 : num 0 5.2 0 0 17.6 0 10.5 0 4.8 0 ...  
## $ km.Z3.4.6 : num 0 0 0 0 7.2 0 6.5 0 0.6 0 ...  
## $ km.Z5.T1.T2.6 : num 0 0.5 0 0 0 0 0 0 0 0 ...  
## $ km.sprinting.6 : num 0 1.2 0 0 0 0 0 0 1.2 0 ...  
## $ strength.training.6 : num 0 0 0 1 0 0 0 0 0 0 ...  
## $ hours.alternative.6 : num 1 0 0 0 0 1 0 0 0 0 ...  
## $ perceived.exertion.6 : num 0.1 0.1 -0.01 0.1 0.11 0.08 0.09 -0.01 0.11 -0.01 ...  
## $ perceived.trainingSuccess.6: num 0 0 -0.01 0 0 0 0 -0.01 0 -0.01 ...  
## $ perceived.recovery.6 : num 0.15 0.17 -0.01 0.17 0.17 0.17 0.18 -0.01 0.18 -0.01 ...  
## $ Athlete.ID : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ injury : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ Date : int 0 1 2 3 4 5 6 7 8 9 ...

Checking for missing values in the dataset. There are no missing values.

sum(is.na(daily))

## [1] 0

How many Total injuries events in the 7 years in the team? Binary variable with 0 represents not injured, 1 represents injured. This column should be converted into a factor in the cleaning process.

table(daily$injury)

##   
## 0 1   
## 42183 583

Looking at summary of columns to see what should be factorized and/or discretized.

summary(daily)

## nr..sessions total.km km.Z3.4 km.Z5.T1.T2   
## Min. :0.0000 Min. : 0.000 Min. : 0.0000 Min. : 0.0000   
## 1st Qu.:0.0000 1st Qu.: 0.000 1st Qu.: 0.0000 1st Qu.: 0.0000   
## Median :1.0000 Median : 6.000 Median : 0.0000 Median : 0.0000   
## Mean :0.8296 Mean : 7.038 Mean : 0.6914 Mean : 0.5799   
## 3rd Qu.:1.0000 3rd Qu.:12.000 3rd Qu.: 0.0000 3rd Qu.: 0.0000   
## Max. :2.0000 Max. :55.900 Max. :42.2000 Max. :48.0000   
## km.sprinting strength.training hours.alternative perceived.exertion  
## Min. : 0.00000 Min. :0.0000 Min. : 0.0000 Min. :-0.0100   
## 1st Qu.: 0.00000 1st Qu.:0.0000 1st Qu.: 0.0000 1st Qu.:-0.0100   
## Median : 0.00000 Median :0.0000 Median : 0.0000 Median : 0.1600   
## Mean : 0.07302 Mean :0.1162 Mean : 0.1635 Mean : 0.2478   
## 3rd Qu.: 0.00000 3rd Qu.:0.0000 3rd Qu.: 0.0000 3rd Qu.: 0.4400   
## Max. :40.00000 Max. :2.0000 Max. :10.2200 Max. : 1.0000   
## perceived.trainingSuccess perceived.recovery nr..sessions.1 total.km.1   
## Min. :-0.0100 Min. :-0.0100 Min. :0.000 Min. : 0.000   
## 1st Qu.:-0.0100 1st Qu.:-0.0100 1st Qu.:0.000 1st Qu.: 0.000   
## Median : 0.2600 Median : 0.1600 Median :1.000 Median : 6.000   
## Mean : 0.3498 Mean : 0.1959 Mean :0.829 Mean : 7.024   
## 3rd Qu.: 0.7200 3rd Qu.: 0.3000 3rd Qu.:1.000 3rd Qu.:12.000   
## Max. : 1.0000 Max. : 1.0000 Max. :2.000 Max. :55.900   
## km.Z3.4.1 km.Z5.T1.T2.1 km.sprinting.1 strength.training.1  
## Min. : 0.0000 Min. : 0.0000 Min. : 0.00000 Min. :0.0000   
## 1st Qu.: 0.0000 1st Qu.: 0.0000 1st Qu.: 0.00000 1st Qu.:0.0000   
## Median : 0.0000 Median : 0.0000 Median : 0.00000 Median :0.0000   
## Mean : 0.6894 Mean : 0.5749 Mean : 0.07261 Mean :0.1171   
## 3rd Qu.: 0.0000 3rd Qu.: 0.0000 3rd Qu.: 0.00000 3rd Qu.:0.0000   
## Max. :42.2000 Max. :48.0000 Max. :40.00000 Max. :2.0000   
## hours.alternative.1 perceived.exertion.1 perceived.trainingSuccess.1  
## Min. : 0.0000 Min. :-0.0100 Min. :-0.0100   
## 1st Qu.: 0.0000 1st Qu.:-0.0100 1st Qu.:-0.0100   
## Median : 0.0000 Median : 0.1600 Median : 0.2500   
## Mean : 0.1643 Mean : 0.2472 Mean : 0.3493   
## 3rd Qu.: 0.0000 3rd Qu.: 0.4400 3rd Qu.: 0.7200   
## Max. :10.2200 Max. : 1.0000 Max. : 1.0000   
## perceived.recovery.1 nr..sessions.2 total.km.2 km.Z3.4.2   
## Min. :-0.0100 Min. :0.0000 Min. : 0.000 Min. : 0.0000   
## 1st Qu.:-0.0100 1st Qu.:0.0000 1st Qu.: 0.000 1st Qu.: 0.0000   
## Median : 0.1600 Median :1.0000 Median : 6.000 Median : 0.0000   
## Mean : 0.1958 Mean :0.8296 Mean : 7.039 Mean : 0.6952   
## 3rd Qu.: 0.3000 3rd Qu.:1.0000 3rd Qu.:12.000 3rd Qu.: 0.0000   
## Max. : 1.0000 Max. :2.0000 Max. :55.900 Max. :42.2000   
## km.Z5.T1.T2.2 km.sprinting.2 strength.training.2 hours.alternative.2  
## Min. : 0.0000 Min. : 0.00000 Min. :0.0000 Min. : 0.0000   
## 1st Qu.: 0.0000 1st Qu.: 0.00000 1st Qu.:0.0000 1st Qu.: 0.0000   
## Median : 0.0000 Median : 0.00000 Median :0.0000 Median : 0.0000   
## Mean : 0.5784 Mean : 0.07314 Mean :0.1165 Mean : 0.1634   
## 3rd Qu.: 0.0000 3rd Qu.: 0.00000 3rd Qu.:0.0000 3rd Qu.: 0.0000   
## Max. :48.0000 Max. :40.00000 Max. :2.0000 Max. :10.2200   
## perceived.exertion.2 perceived.trainingSuccess.2 perceived.recovery.2  
## Min. :-0.0100 Min. :-0.0100 Min. :-0.0100   
## 1st Qu.:-0.0100 1st Qu.:-0.0100 1st Qu.:-0.0100   
## Median : 0.1600 Median : 0.2600 Median : 0.1600   
## Mean : 0.2477 Mean : 0.3494 Mean : 0.1961   
## 3rd Qu.: 0.4400 3rd Qu.: 0.7200 3rd Qu.: 0.3000   
## Max. : 1.0000 Max. : 1.0000 Max. : 1.0000   
## nr..sessions.3 total.km.3 km.Z3.4.3 km.Z5.T1.T2.3   
## Min. :0.0000 Min. : 0.00 Min. : 0.0000 Min. : 0.0000   
## 1st Qu.:0.0000 1st Qu.: 0.00 1st Qu.: 0.0000 1st Qu.: 0.0000   
## Median :1.0000 Median : 6.00 Median : 0.0000 Median : 0.0000   
## Mean :0.8293 Mean : 7.04 Mean : 0.6926 Mean : 0.5782   
## 3rd Qu.:1.0000 3rd Qu.:12.00 3rd Qu.: 0.0000 3rd Qu.: 0.0000   
## Max. :2.0000 Max. :57.00 Max. :42.2000 Max. :48.0000   
## km.sprinting.3 strength.training.3 hours.alternative.3  
## Min. : 0.00000 Min. :0.0000 Min. : 0.000   
## 1st Qu.: 0.00000 1st Qu.:0.0000 1st Qu.: 0.000   
## Median : 0.00000 Median :0.0000 Median : 0.000   
## Mean : 0.07269 Mean :0.1163 Mean : 0.163   
## 3rd Qu.: 0.00000 3rd Qu.:0.0000 3rd Qu.: 0.000   
## Max. :40.00000 Max. :2.0000 Max. :10.220   
## perceived.exertion.3 perceived.trainingSuccess.3 perceived.recovery.3  
## Min. :-0.0100 Min. :-0.0100 Min. :-0.0100   
## 1st Qu.:-0.0100 1st Qu.:-0.0100 1st Qu.:-0.0100   
## Median : 0.1600 Median : 0.2600 Median : 0.1600   
## Mean : 0.2477 Mean : 0.3496 Mean : 0.1962   
## 3rd Qu.: 0.4400 3rd Qu.: 0.7200 3rd Qu.: 0.3000   
## Max. : 1.0000 Max. : 1.0000 Max. : 1.0000   
## nr..sessions.4 total.km.4 km.Z3.4.4 km.Z5.T1.T2.4   
## Min. :0.0000 Min. : 0.000 Min. : 0.0000 Min. : 0.0000   
## 1st Qu.:0.0000 1st Qu.: 0.000 1st Qu.: 0.0000 1st Qu.: 0.0000   
## Median :1.0000 Median : 6.000 Median : 0.0000 Median : 0.0000   
## Mean :0.8299 Mean : 7.043 Mean : 0.6955 Mean : 0.5778   
## 3rd Qu.:1.0000 3rd Qu.:12.000 3rd Qu.: 0.0000 3rd Qu.: 0.0000   
## Max. :2.0000 Max. :55.900 Max. :42.2000 Max. :48.0000   
## km.sprinting.4 strength.training.4 hours.alternative.4  
## Min. : 0.00000 Min. :0.0000 Min. : 0.0000   
## 1st Qu.: 0.00000 1st Qu.:0.0000 1st Qu.: 0.0000   
## Median : 0.00000 Median :0.0000 Median : 0.0000   
## Mean : 0.07217 Mean :0.1162 Mean : 0.1627   
## 3rd Qu.: 0.00000 3rd Qu.:0.0000 3rd Qu.: 0.0000   
## Max. :40.00000 Max. :2.0000 Max. :10.0000   
## perceived.exertion.4 perceived.trainingSuccess.4 perceived.recovery.4  
## Min. :-0.0100 Min. :-0.0100 Min. :-0.0100   
## 1st Qu.:-0.0100 1st Qu.:-0.0100 1st Qu.:-0.0100   
## Median : 0.1600 Median : 0.2600 Median : 0.1700   
## Mean : 0.2476 Mean : 0.3498 Mean : 0.1963   
## 3rd Qu.: 0.4400 3rd Qu.: 0.7200 3rd Qu.: 0.3000   
## Max. : 1.0000 Max. : 1.0000 Max. : 1.0000   
## nr..sessions.5 total.km.5 km.Z3.4.5 km.Z5.T1.T2.5   
## Min. :0.0000 Min. : 0.000 Min. : 0.0000 Min. : 0.0000   
## 1st Qu.:0.0000 1st Qu.: 0.000 1st Qu.: 0.0000 1st Qu.: 0.0000   
## Median :1.0000 Median : 6.000 Median : 0.0000 Median : 0.0000   
## Mean :0.8288 Mean : 7.055 Mean : 0.6974 Mean : 0.5801   
## 3rd Qu.:1.0000 3rd Qu.:12.000 3rd Qu.: 0.0000 3rd Qu.: 0.0000   
## Max. :2.0000 Max. :55.900 Max. :42.2000 Max. :48.0000   
## km.sprinting.5 strength.training.5 hours.alternative.5 perceived.exertion.5  
## Min. : 0.0000 Min. :0.0000 Min. : 0.0000 Min. :-0.0100   
## 1st Qu.: 0.0000 1st Qu.:0.0000 1st Qu.: 0.0000 1st Qu.:-0.0100   
## Median : 0.0000 Median :0.0000 Median : 0.0000 Median : 0.1600   
## Mean : 0.0723 Mean :0.1159 Mean : 0.1613 Mean : 0.2477   
## 3rd Qu.: 0.0000 3rd Qu.:0.0000 3rd Qu.: 0.0000 3rd Qu.: 0.4400   
## Max. :40.0000 Max. :2.0000 Max. :10.0000 Max. : 1.0000   
## perceived.trainingSuccess.5 perceived.recovery.5 nr..sessions.6   
## Min. :-0.0100 Min. :-0.0100 Min. :0.0000   
## 1st Qu.:-0.0100 1st Qu.:-0.0100 1st Qu.:0.0000   
## Median : 0.2600 Median : 0.1700 Median :1.0000   
## Mean : 0.3497 Mean : 0.1963 Mean :0.8288   
## 3rd Qu.: 0.7200 3rd Qu.: 0.3000 3rd Qu.:1.0000   
## Max. : 1.0000 Max. : 1.0000 Max. :2.0000   
## total.km.6 km.Z3.4.6 km.Z5.T1.T2.6 km.sprinting.6   
## Min. : 0.000 Min. : 0.000 Min. : 0.0000 Min. : 0.0000   
## 1st Qu.: 0.000 1st Qu.: 0.000 1st Qu.: 0.0000 1st Qu.: 0.0000   
## Median : 6.000 Median : 0.000 Median : 0.0000 Median : 0.0000   
## Mean : 7.051 Mean : 0.695 Mean : 0.5804 Mean : 0.0726   
## 3rd Qu.:12.000 3rd Qu.: 0.000 3rd Qu.: 0.0000 3rd Qu.: 0.0000   
## Max. :55.900 Max. :42.200 Max. :48.0000 Max. :40.0000   
## strength.training.6 hours.alternative.6 perceived.exertion.6  
## Min. :0.0000 Min. : 0.0000 Min. :-0.0100   
## 1st Qu.:0.0000 1st Qu.: 0.0000 1st Qu.:-0.0100   
## Median :0.0000 Median : 0.0000 Median : 0.1600   
## Mean :0.1161 Mean : 0.1623 Mean : 0.2475   
## 3rd Qu.:0.0000 3rd Qu.: 0.0000 3rd Qu.: 0.4400   
## Max. :2.0000 Max. :20.0000 Max. : 1.0000   
## perceived.trainingSuccess.6 perceived.recovery.6 Athlete.ID   
## Min. :-0.0100 Min. :-0.0100 Min. : 0.00   
## 1st Qu.:-0.0100 1st Qu.:-0.0100 1st Qu.:20.00   
## Median : 0.2600 Median : 0.1700 Median :34.00   
## Mean : 0.3495 Mean : 0.1962 Mean :34.55   
## 3rd Qu.: 0.7200 3rd Qu.: 0.3000 3rd Qu.:50.00   
## Max. : 1.0000 Max. : 1.0000 Max. :73.00   
## injury Date   
## Min. :0.00000 Min. : 0   
## 1st Qu.:0.00000 1st Qu.: 436   
## Median :0.00000 Median :1256   
## Mean :0.01363 Mean :1228   
## 3rd Qu.:0.00000 3rd Qu.:1913   
## Max. :1.00000 Max. :2673

The variables for kilometer could be binned to reduce the number of records. We will divide them into percentiles below:

#Total Km each day  
daily\_linear<-daily  
#day 1  
Percentile\_00 = min(daily\_linear$total.km)  
Percentile\_33 = quantile(daily\_linear$total.km, 0.33333)  
Percentile\_67 = quantile(daily\_linear$total.km, 0.66667)  
Percentile\_100 = max(daily\_linear$total.km)  
  
Daily\_bindA = rbind(Percentile\_00, Percentile\_33, Percentile\_67, Percentile\_100)  
dimnames(Daily\_bindA)[[2]] = "Value"  
  
#day 2  
Percentile\_00.1 = min(daily\_linear$total.km.1)  
Percentile\_33.1 = quantile(daily\_linear$total.km.1, 0.33333)  
Percentile\_67.1 = quantile(daily\_linear$total.km.1, 0.66667)  
Percentile\_100.1 = max(daily\_linear$total.km.1)  
  
Daily\_bindA1 = rbind(Percentile\_00, Percentile\_33, Percentile\_67, Percentile\_100)  
dimnames(Daily\_bindA1)[[2]] = "Value"  
  
#Day3  
Percentile\_00.2 = min(daily\_linear$total.km.2)  
Percentile\_33.2 = quantile(daily\_linear$total.km.2, 0.33333)  
Percentile\_67.2 = quantile(daily\_linear$total.km.2, 0.66667)  
Percentile\_100.2 = max(daily\_linear$total.km.2)  
  
Daily\_bindA2 = rbind(Percentile\_00, Percentile\_33, Percentile\_67, Percentile\_100)  
dimnames(Daily\_bindA2)[[2]] = "Value"  
  
#day4  
Percentile\_00.3 = min(daily\_linear$total.km.3)  
Percentile\_33.3 = quantile(daily\_linear$total.km.3, 0.33333)  
Percentile\_67.3 = quantile(daily\_linear$total.km.3, 0.66667)  
Percentile\_100.3 = max(daily\_linear$total.km.3)  
  
Daily\_bindA3 = rbind(Percentile\_00, Percentile\_33, Percentile\_67, Percentile\_100)  
dimnames(Daily\_bindA3)[[2]] = "Value"  
  
#day 5  
Percentile\_00.4 = min(daily\_linear$total.km.4)  
Percentile\_33.4 = quantile(daily\_linear$total.km.4, 0.33333)  
Percentile\_67.4 = quantile(daily\_linear$total.km.4, 0.66667)  
Percentile\_100.4 = max(daily\_linear$total.km.4)  
  
Daily\_bindA4 = rbind(Percentile\_00, Percentile\_33, Percentile\_67, Percentile\_100)  
dimnames(Daily\_bindA4)[[2]] = "Value"  
  
#day 6  
Percentile\_00.5 = min(daily\_linear$total.km.5)  
Percentile\_33.5 = quantile(daily\_linear$total.km.5, 0.33333)  
Percentile\_67.5 = quantile(daily\_linear$total.km.5, 0.66667)  
Percentile\_100.5 = max(daily\_linear$total.km.5)  
  
Daily\_bindA5 = rbind(Percentile\_00, Percentile\_33, Percentile\_67, Percentile\_100)  
dimnames(Daily\_bindA5)[[2]] = "Value"  
  
#day 7  
Percentile\_00.6 = min(daily\_linear$total.km.6)  
Percentile\_33.6 = quantile(daily\_linear$total.km.6, 0.33333)  
Percentile\_67.6 = quantile(daily\_linear$total.km.6, 0.66667)  
Percentile\_100.6 = max(daily\_linear$total.km.6)  
  
Daily\_bindA6 = rbind(Percentile\_00, Percentile\_33, Percentile\_67, Percentile\_100)  
dimnames(Daily\_bindA6)[[2]] = "Value"  
  
Percentiles.Daily <- cbind(Daily\_bindA, Daily\_bindA1, Daily\_bindA2, Daily\_bindA3, Daily\_bindA4, Daily\_bindA5, Daily\_bindA6)  
colnames(Percentiles.Daily)<-c("total.km day 1","total.km day 2","total.km day 3","total.km day 4","total.km day 5","total.km day 6","total.km day 7")  
  
kable(t(Percentiles.Daily), digits = 0, format = "markdown", padding =2, format.args = list(big.mark =","))

|  | Percentile\_00 | Percentile\_33 | Percentile\_67 | Percentile\_100 |
| --- | --- | --- | --- | --- |
| total.km day 1 | 0 | 0 | 10 | 56 |
| total.km day 2 | 0 | 0 | 10 | 56 |
| total.km day 3 | 0 | 0 | 10 | 56 |
| total.km day 4 | 0 | 0 | 10 | 56 |
| total.km day 5 | 0 | 0 | 10 | 56 |
| total.km day 6 | 0 | 0 | 10 | 56 |
| total.km day 7 | 0 | 0 | 10 | 56 |

# Injury to non-injury

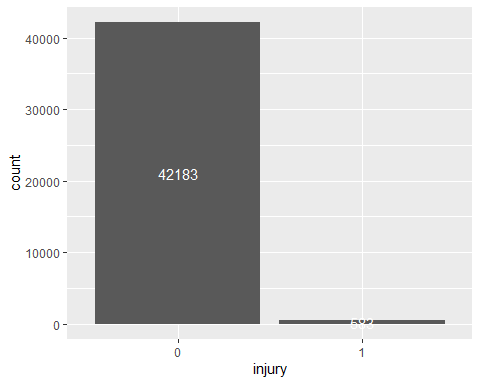
Converting the injury variable to factor and Plotting the injuries. This shows us how unbalanced the dataset is.

daily$injury <- as.factor(daily$injury)  
injuryplot<- ggplot(daily) +   
 aes(x=injury)+geom\_histogram(stat="count") +  
 stat\_count(geom = "text",  
 color = "white",  
 aes(label = ..count..),  
 position= position\_stack(vjust= 0.5))

## Warning in geom\_histogram(stat = "count"): Ignoring unknown parameters:  
## `binwidth`, `bins`, and `pad`

injuryplot

## Warning: The dot-dot notation (`..count..`) was deprecated in ggplot2 3.4.0.  
## ℹ Please use `after\_stat(count)` instead.



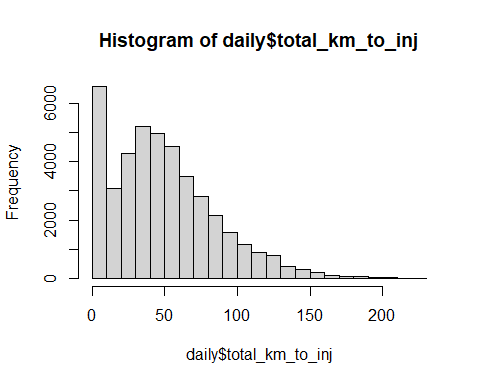
#Total km ran leading up to the next “event” or injury. Adding all total kilometers ran 7 days before possible injury event

daily$total\_km\_to\_inj <- daily$total.km + daily$total.km.1 + daily$total.km.2 + daily$total.km.3 + daily$total.km.4+ daily$total.km.5 + daily$total.km.6  
summary(daily$total\_km\_to\_inj)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.00 22.70 44.60 49.29 69.90 220.30

#Histogram of calculated total\_km\_to\_inj Histogram of Total km 7 days before “event” to view distribution of total km Athletes ran 7 days leading to “injury” or no injury Note: There’s more than 5000 rows in which athletes run a total of 0 km before possible injury.

hist(daily$total\_km\_to\_inj)

 # Subjective subset Creating a subset with all subjective “perceived” data to look at the different distributions. The subjective data consist of how the athletes felt pre- workout (perceived recovery), how exerted the athletes felt (perceived exertion), and how successful the athletes felt they did post-workout. There is 7 days for each variable from Day 0 (day before “injury” or “no injury”) to Day 6 (7 days before “injury” or “no injury” ).

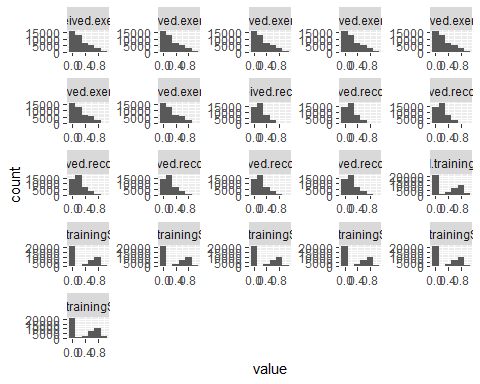
subjectivesubset <- daily %>% select(starts\_with(c("perceived.")))

# Histograms of all subjective variables

This shows that there was indications where: -athletes felt exerted after their workouts -athletes felt they did not fully recover from the previous day -athletes did not feel they were successful post-workout

subjectivesubset %>%  
 keep(is.numeric) %>%  
 gather() %>%  
 ggplot(aes(value)) +  
 facet\_wrap(~ key, scales = "free") +  
 geom\_histogram() +  
 stat\_bin(binwidth = .20)

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



# Subset all variables of KM ran by athletes Subsetting all data of distance ran by athletes in km. This subset consist of Total KM, Total distance sprinted, total distance in different heart rate zones; Z3-Z4, and Z5 in long and short track intervals (T1 and T2). There is 7 days for each variable from Day 0 (day before “injury” or “no injury”) to Day 6 (7 days before “injury” or “no injury” ).

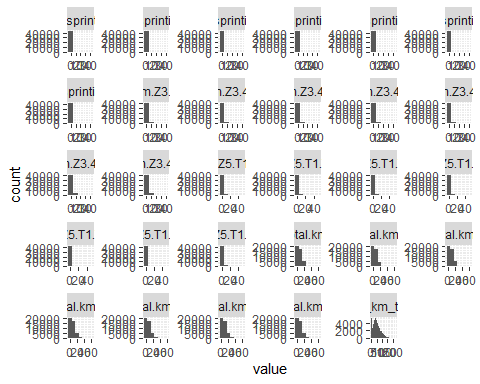
kmsubset <- daily %>% select(contains(c("km")))

# Histograms of all distance (km) data

Once again, this shows how unbalance the data is. For instance, for km.Z5.T1.T2, it looks as if no athletes ran during anaerobic max heart rate zone. However, when simply calling the head() function, some km are visible aside from all zeros. This also has to do with the athletes not running at max heart rate zone at ALL times.

kmsubset %>%  
 keep(is.numeric) %>%  
 gather() %>%  
 ggplot(aes(value)) +  
 facet\_wrap(~ key, scales = "free") +  
 geom\_histogram() +  
 stat\_bin(binwidth = 10)

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



# Run head() function for km ran during heart rate zone 5 for Day 0 to ensure data exist besides 0

head(kmsubset$km.Z5.T1.T2, 200)

## [1] 0.6 0.0 0.0 0.0 0.0 0.0 0.0 0.5 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
## [19] 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.3  
## [37] 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
## [55] 4.2 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.5 0.0 0.0 1.5 0.0 0.0 0.0 0.5 0.0  
## [73] 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.1 0.0 0.0 0.0 0.0 2.7  
## [91] 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 2.0 0.0 1.6 0.0 2.4 0.0 0.0 0.0 1.6  
## [109] 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.4 0.0 0.5 0.0 0.0 0.5 0.0 0.0 0.6  
## [127] 0.0 0.0 0.0 0.0 0.8 0.0 0.0 0.4 2.2 0.0 3.2 0.0 1.8 0.0 0.0 0.0 0.0 0.0  
## [145] 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
## [163] 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
## [181] 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
## [199] 0.0 0.0

# Strength and Alternative training

Subsetting all Strength, and alternative training (hrs) performed by athletes. There is 7 days for each variable from Day 0 (day before “injury” or “no injury”) to Day 6 (7 days before “injury” or “no injury” ).

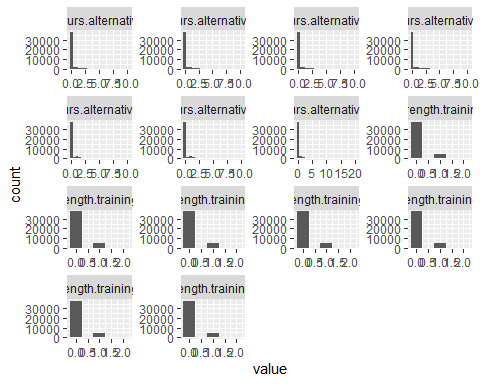
strength\_alt\_subset <- daily %>% select(contains(c("strength", "alternative")))

# Histograms of all Strength and Alternative training

This shows that athletes are also involved in other training activities aside from running

strength\_alt\_subset %>%  
 keep(is.numeric) %>%  
 gather() %>%  
 ggplot(aes(value)) +  
 facet\_wrap(~ key, scales = "free") +  
 geom\_histogram() +  
 stat\_bin(binwidth = .50)

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

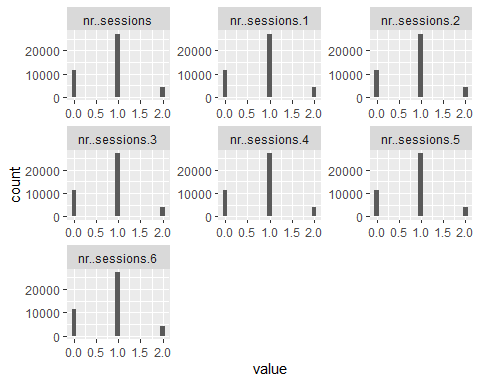
 # Number of sessions Subsetting number of sessions the athletes performed for all days leading to “injury” or “no injury”

nr\_sessions\_subset <- daily %>% select(contains(c("sessions")))

# Histograms of all number of sessions data for each day leading to “injury” or “no injury”

nr\_sessions\_subset %>%  
 keep(is.numeric) %>%  
 gather() %>%  
 ggplot(aes(value)) +  
 facet\_wrap(~ key, scales = "free") +  
 geom\_histogram() +  
 stat\_bin(binwidth = .10)

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



# Breaks for km sprinting

Out of curiosity, buckets are created to see how many km were logged in the day leading to “event”(possible injury)

daily$sprintkm\_0\_disc <- cut(daily$km.sprinting, breaks = c(0.0, 0.3, 0.4, 0.5, 0.7, 0.9, 1.0, 1.2, 1.4, 1.6, 1.8, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5, 5.0, 5.5, 6.0, 6.5, 7.0, 7.5, 8.0, 8.5, 9.0, 9.5, 10.0, 10.5, 11.0, 12.0, 13.0, 14.0, 15.0, 16.0, 17.0, 18.0, 40.0),  
 labels = c("0.0-0.29","0.3-0.39","0.4-0.49","0.5-0.69","0.7-0.89","0.9-0.99","1.0-1.19", "1.2-1.39","1.4-1.59","1.6-1.79","1.8-1.99","2.0-2.49","2.5-2.9","3.0-3.49","3.5-3.9","4.0-4.49","4.5-4.49",  
"5.0-5.49","5.5-5.49", "6.0-6.49", "6.5-6.99", "7.0-7.49", "7.5-7.99", "8.0-8.49", "9.0-9.49", "9.5-9.99", "10.0-10.49", "10.5-10.99", "11.0-11.99","12.0-12.99", "13.0-13.99","14.0-14.99","15.0-15.99","16.0-16.99","17.0-17.99","18.0-39.9", "40.0+"), right=FALSE)

View in table

table(daily$sprintkm\_0\_disc)

##   
## 0.0-0.29 0.3-0.39 0.4-0.49 0.5-0.69 0.7-0.89 0.9-0.99 1.0-1.19   
## 39789 503 392 631 309 61 433   
## 1.2-1.39 1.4-1.59 1.6-1.79 1.8-1.99 2.0-2.49 2.5-2.9 3.0-3.49   
## 162 110 44 21 117 30 58   
## 3.5-3.9 4.0-4.49 4.5-4.49 5.0-5.49 5.5-5.49 6.0-6.49 6.5-6.99   
## 12 22 8 13 5 8 1   
## 7.0-7.49 7.5-7.99 8.0-8.49 9.0-9.49 9.5-9.99 10.0-10.49 10.5-10.99   
## 3 3 5 4 4 2 1   
## 11.0-11.99 12.0-12.99 13.0-13.99 14.0-14.99 15.0-15.99 16.0-16.99 17.0-17.99   
## 1 3 2 2 1 4 0   
## 18.0-39.9 40.0+   
## 0 1

**Visualizing the distance in km logged the day leading to “event”**

Here is visible that there was some sprinting that took place. However, there’s also more indication of imbalance of the data as there was more “0s” logged the day before.

ggplot(daily) + aes(x=sprintkm\_0\_disc) +   
  geom\_bar(fill = "orange", col="black") +   
  theme(axis.text.x=   
          element\_text(angle=90, hjust=1)) +   
  ggtitle("Sprinting distance (km) 1 day before event")

**Chart

Description automatically generated**

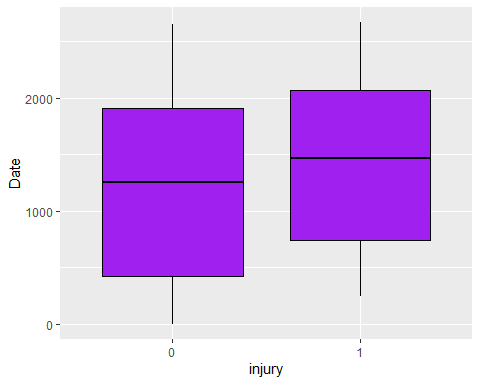
# How Many days are in this dataset and when did most injuries take place?

In comparison to no injury events, most injury events occurred in the later years well into the experiment.

summary(daily$Date)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0 436 1256 1228 1913 2673

Date\_inj<- ggplot(daily) +   
 aes(x=injury, y=Date)+geom\_boxplot(fill="purple", col="black")  
Date\_inj

 #Visualize Variables

Create subsets of each day for further review

day1 <- daily[, c(1:10, 71:73)]  
day2 <- daily[, c(11:20, 71:73)]  
day3 <- daily[, c(21:30, 71:73)]  
day4 <- daily[, c(31:40, 71:73)]  
day5 <- daily[, c(41:50, 71:73)]  
day6 <- daily[, c(51:60, 71:73)]  
day7 <- daily[, c(61:70, 71:73)]

Create subsets for each injured for the full dataset and the daily sets

injuredDaily <- subset(daily, injury == "1")  
str(injuredDaily)

## 'data.frame': 583 obs. of 75 variables:  
## $ nr..sessions : num 1 0 0 1 1 1 1 1 1 2 ...  
## $ total.km : num 0 0 0 7.5 3.5 6.9 4.9 16 15 24 ...  
## $ km.Z3.4 : num 0 0 0 4 0 3.9 0 0 0 0 ...  
## $ km.Z5.T1.T2 : num 0 0 0 0 0 0 0.4 0 0 6 ...  
## $ km.sprinting : num 0 0 0 0 0.5 0 0 0 0 0 ...  
## $ strength.training : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ hours.alternative : num 1 0 0 0 0 0 0 0 0 0 ...  
## $ perceived.exertion : num 0.08 -0.01 -0.01 0.1 0.08 0.09 0.09 0.07 0.1 0.77 ...  
## $ perceived.trainingSuccess : num 0 -0.01 -0.01 0 0 0 0 0.14 0.87 0.78 ...  
## $ perceived.recovery : num 0.18 -0.01 -0.01 0.18 0.18 0.17 0.18 0.31 0.18 0.16 ...  
## $ nr..sessions.1 : num 1 1 0 1 2 1 0 1 1 1 ...  
## $ total.km.1 : num 11.5 0 0 12.5 3.9 6.3 0 19 10 16 ...  
## $ km.Z3.4.1 : num 5 0 0 4 0 1.6 0 9 0 0 ...  
## $ km.Z5.T1.T2.1 : num 0 0 0 0 0.4 0 0 0 0 0 ...  
## $ km.sprinting.1 : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ strength.training.1 : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ hours.alternative.1 : num 0 1 0 0 0.67 0 0 0 0 0 ...  
## $ perceived.exertion.1 : num 0.1 0.08 -0.01 0.09 0.09 0.1 -0.01 0.48 0.62 0.22 ...  
## $ perceived.trainingSuccess.1: num 0 0 -0.01 0 0 0 -0.01 0.76 0.7 0.65 ...  
## $ perceived.recovery.1 : num 0.18 0.16 -0.01 0.18 0.18 0.17 -0.01 0.29 0.25 0.27 ...  
## $ nr..sessions.2 : num 0 1 0 1 2 1 1 1 0 1 ...  
## $ total.km.2 : num 0 13.3 0 0 4.4 0 12 22 0 11 ...  
## $ km.Z3.4.2 : num 0 6 0 0 0 0 0 0 0 0 ...  
## $ km.Z5.T1.T2.2 : num 0 1.8 0 0 0 0 0 0 0 0 ...  
## $ km.sprinting.2 : num 0 0 0 0 0.9 0 0 0 0 0 ...  
## $ strength.training.2 : num 0 0 0 1 1 1 0 0 0 0 ...  
## $ hours.alternative.2 : num 0 0 0 0.75 0 0 0 0 0 0 ...  
## $ perceived.exertion.2 : num -0.01 0.1 -0.01 0.09 0.1 0.1 0.09 0.1 -0.01 0.33 ...  
## $ perceived.trainingSuccess.2: num -0.01 0 -0.01 0 0 0 0 0.84 -0.01 0.72 ...  
## $ perceived.recovery.2 : num -0.01 0.17 -0.01 0.17 0.17 0.16 0.17 0.23 -0.01 0.15 ...  
## $ nr..sessions.3 : num 1 1 1 1 0 1 0 0 1 1 ...  
## $ total.km.3 : num 4 0 7 5.7 0 4 0 0 15 9 ...  
## $ km.Z3.4.3 : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ km.Z5.T1.T2.3 : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ km.sprinting.3 : num 0 0 0 0.7 0 0 0 0 0 0 ...  
## $ strength.training.3 : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ hours.alternative.3 : num 0 0.75 0 0 0 0 0 0 0 0 ...  
## $ perceived.exertion.3 : num 0.1 0.08 0.08 0.09 -0.01 0.08 -0.01 -0.01 0.08 0.09 ...  
## $ perceived.trainingSuccess.3: num 0 0 0 0 -0.01 0 -0.01 -0.01 0.81 0.85 ...  
## $ perceived.recovery.3 : num 0.17 0.16 0.18 0.17 -0.01 0.17 -0.01 -0.01 0.2 0.5 ...  
## $ nr..sessions.4 : num 1 1 0 1 2 1 0 1 1 1 ...  
## $ total.km.4 : num 0 5.1 0 10.5 5.2 6.8 0 19 30 18 ...  
## $ km.Z3.4.4 : num 0 0 0 0 0 0 0 0 0 9 ...  
## $ km.Z5.T1.T2.4 : num 0 0 0 0 1.2 0.8 0 0 0 0 ...  
## $ km.sprinting.4 : num 0 1.1 0 0 0 0 0 0 0 0 ...  
## $ strength.training.4 : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ hours.alternative.4 : num 1.25 0 0 0 0.67 0 0 0 0 0 ...  
## $ perceived.exertion.4 : num 0.09 0.09 -0.01 0.09 0.11 0.11 -0.01 0.09 0.16 0.36 ...  
## $ perceived.trainingSuccess.4: num 0 0 -0.01 0 0 0 -0.01 0.82 0.66 0.36 ...  
## $ perceived.recovery.4 : num 0.17 0.17 -0.01 0.18 0.18 0.18 -0.01 0.16 0.24 0.31 ...  
## $ nr..sessions.5 : num 1 1 0 1 1 1 1 1 1 0 ...  
## $ total.km.5 : num 4.9 15.2 0 11 7.5 4.5 6.5 19 24 0 ...  
## $ km.Z3.4.5 : num 0 6.4 0 6 0 4.5 1.5 11 0 0 ...  
## $ km.Z5.T1.T2.5 : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ km.sprinting.5 : num 0.9 0 0 0 0 0 0 0 0 0 ...  
## $ strength.training.5 : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ hours.alternative.5 : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ perceived.exertion.5 : num 0.1 0.09 -0.01 0.1 0.09 0.09 0.09 0.3 0.13 -0.01 ...  
## $ perceived.trainingSuccess.5: num 0 0 -0.01 0 0 0 0 0.86 0.82 -0.01 ...  
## $ perceived.recovery.5 : num 0.17 0.17 -0.01 0.18 0.17 0.15 0.18 0.14 0.15 -0.01 ...  
## $ nr..sessions.6 : num 1 1 0 1 1 1 0 1 1 0 ...  
## $ total.km.6 : num 14.5 0 0 0 4.9 0 0 22 20 0 ...  
## $ km.Z3.4.6 : num 4.5 0 0 0 0 0 0 0 0 0 ...  
## $ km.Z5.T1.T2.6 : num 0 0 0 0 0.9 0 0 0 10 0 ...  
## $ km.sprinting.6 : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ strength.training.6 : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ hours.alternative.6 : num 0 1 0 1.5 0 1 0 0 0 0 ...  
## $ perceived.exertion.6 : num 0.08 0.08 -0.01 0.08 0.1 0.08 -0.01 0.17 0.59 -0.01 ...  
## $ perceived.trainingSuccess.6: num 0 0 -0.01 0 0 0 -0.01 0.65 0.53 -0.01 ...  
## $ perceived.recovery.6 : num 0.17 0.17 -0.01 0.17 0.18 0.17 -0.01 0.32 0.35 -0.01 ...  
## $ Athlete.ID : int 0 0 0 0 0 0 0 1 1 2 ...  
## $ injury : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...  
## $ Date : int 324 353 386 413 461 490 528 672 765 680 ...  
## $ total\_km\_to\_inj : num 34.9 33.6 7 47.2 29.4 28.5 23.4 117 114 78 ...  
## $ sprintkm\_0\_disc : Factor w/ 37 levels "0.0-0.29","0.3-0.39",..: 1 1 1 1 4 1 1 1 1 1 ...

injured\_d1 <- subset(day1, injury == "1")  
injured\_d2 <- subset(day2, injury == "1")  
injured\_d3 <- subset(day3, injury == "1")  
injured\_d4 <- subset(day4, injury == "1")  
injured\_d5 <- subset(day5, injury == "1")  
injured\_d6 <- subset(day6, injury == "1")  
injured\_d7 <- subset(day7, injury == "1")

Remove Athlete ID and Date variable

Aid <- as.character(daily[["Athlete.ID"]])  
daily$Athlete.ID <- NULL  
  
date <- as.character(daily[["Date"]])  
daily$Date <- NULL

#Association Rule Mining

We will use the Apriori algorithm to generate association rules to help us understand which variables contribute to injury.

dailyARM <- daily  
  
dailyARM<-lapply(dailyARM, function(x){as.factor(x)})  
dailyARM = as.data.frame(dailyARM)

dailytrans = as(dailyARM, "transactions")  
  
str(dailytrans)

## Formal class 'transactions' [package "arules"] with 3 slots  
## ..@ data :Formal class 'ngCMatrix' [package "Matrix"] with 5 slots  
## .. .. ..@ i : int [1:3121917] 1 61 387 567 707 773 776 996 1087 1207 ...  
## .. .. ..@ p : int [1:42767] 0 73 146 219 292 365 438 511 584 657 ...  
## .. .. ..@ Dim : int [1:2] 10807 42766  
## .. .. ..@ Dimnames:List of 2  
## .. .. .. ..$ : NULL  
## .. .. .. ..$ : NULL  
## .. .. ..@ factors : list()  
## ..@ itemInfo :'data.frame': 10807 obs. of 3 variables:  
## .. ..$ labels : chr [1:10807] "nr..sessions=0" "nr..sessions=1" "nr..sessions=2" "total.km=0" ...  
## .. ..$ variables: Factor w/ 73 levels "hours.alternative",..: 30 30 30 66 66 66 66 66 66 66 ...  
## .. ..$ levels : Factor w/ 2007 levels "-0.01","0","0.0-0.29",..: 2 108 1008 2 13 23 33 44 55 66 ...  
## ..@ itemsetInfo:'data.frame': 42766 obs. of 1 variable:  
## .. ..$ transactionID: chr [1:42766] "1" "2" "3" "4" ...

Interesting Rules are not found when running the Apriori, even after utilizing discretized sprint km column. All rules point to the 0s in the data set.

rules <- apriori(dailytrans, parameter = list(supp=.80, conf = .50, minlen=3))

## Apriori  
##   
## Parameter specification:  
## confidence minval smax arem aval originalSupport maxtime support minlen  
## 0.5 0.1 1 none FALSE TRUE 5 0.8 3  
## maxlen target ext  
## 10 rules TRUE  
##   
## Algorithmic control:  
## filter tree heap memopt load sort verbose  
## 0.1 TRUE TRUE FALSE TRUE 2 TRUE  
##   
## Absolute minimum support count: 34212   
##   
## set item appearances ...[0 item(s)] done [0.00s].  
## set transactions ...[10805 item(s), 42766 transaction(s)] done [5.34s].  
## sorting and recoding items ... [37 item(s)] done [0.07s].  
## creating transaction tree ... done [0.13s].  
## checking subsets of size 1 2 3 4 5 done [0.77s].  
## writing ... [1182 rule(s)] done [0.00s].  
## creating S4 object ... done [0.02s].

rules <- sort(rules, decreasing = TRUE, by="lift")  
arules::inspect(rules[1:50])

## lhs rhs support confidence coverage lift count  
## [1] {km.sprinting.2=0,   
## km.sprinting.5=0,   
## injury=0,   
## sprintkm\_0\_disc=0.0-0.29} => {km.sprinting=0} 0.8193659 0.9967856 0.8220081 1.078084 35041  
## [2] {km.sprinting.2=0,   
## km.sprinting.5=0,   
## sprintkm\_0\_disc=0.0-0.29} => {km.sprinting=0} 0.8299584 0.9967705 0.8326474 1.078068 35494  
## [3] {km.Z5.T1.T2=0,   
## injury=0,   
## sprintkm\_0\_disc=0.0-0.29} => {km.sprinting=0} 0.8052425 0.9967582 0.8078614 1.078055 34437  
## [4] {km.Z5.T1.T2=0,   
## sprintkm\_0\_disc=0.0-0.29} => {km.sprinting=0} 0.8149231 0.9967396 0.8175887 1.078035 34851  
## [5] {km.sprinting.4=0,   
## km.sprinting.5=0,   
## sprintkm\_0\_disc=0.0-0.29} => {km.sprinting=0} 0.8240658 0.9966065 0.8268718 1.077891 35242  
## [6] {km.sprinting.4=0,   
## km.sprinting.5=0,   
## injury=0,   
## sprintkm\_0\_disc=0.0-0.29} => {km.sprinting=0} 0.8134032 0.9965907 0.8161858 1.077874 34786  
## [7] {km.sprinting.2=0,   
## km.sprinting.3=0,   
## sprintkm\_0\_disc=0.0-0.29} => {km.sprinting=0} 0.8216808 0.9964272 0.8246270 1.077697 35140  
## [8] {km.sprinting.2=0,   
## km.sprinting.3=0,   
## injury=0,   
## sprintkm\_0\_disc=0.0-0.29} => {km.sprinting=0} 0.8112987 0.9964102 0.8142216 1.077678 34696  
## [9] {km.sprinting.3=0,   
## km.sprinting.4=0,   
## sprintkm\_0\_disc=0.0-0.29} => {km.sprinting=0} 0.8205818 0.9962527 0.8236683 1.077508 35093  
## [10] {km.sprinting.3=0,   
## km.sprinting.5=0,   
## sprintkm\_0\_disc=0.0-0.29} => {km.sprinting=0} 0.8300285 0.9962113 0.8331852 1.077463 35497  
## [11] {km.sprinting.3=0,   
## km.sprinting.4=0,   
## injury=0,   
## sprintkm\_0\_disc=0.0-0.29} => {km.sprinting=0} 0.8100360 0.9962041 0.8131226 1.077455 34642  
## [12] {km.sprinting.5=0,   
## km.sprinting.6=0,   
## sprintkm\_0\_disc=0.0-0.29} => {km.sprinting=0} 0.8206519 0.9961964 0.8237852 1.077447 35096  
## [13] {km.sprinting.3=0,   
## km.sprinting.5=0,   
## injury=0,   
## sprintkm\_0\_disc=0.0-0.29} => {km.sprinting=0} 0.8194126 0.9961907 0.8225459 1.077441 35043  
## [14] {km.sprinting.5=0,   
## km.sprinting.6=0,   
## injury=0,   
## sprintkm\_0\_disc=0.0-0.29} => {km.sprinting=0} 0.8105037 0.9961776 0.8136136 1.077427 34662  
## [15] {km.sprinting.2=0,   
## km.sprinting.4=0,   
## sprintkm\_0\_disc=0.0-0.29} => {km.sprinting=0} 0.8288828 0.9960661 0.8321564 1.077306 35448  
## [16] {km.sprinting.2=0,   
## km.sprinting.4=0,   
## injury=0,   
## sprintkm\_0\_disc=0.0-0.29} => {km.sprinting=0} 0.8183136 0.9960438 0.8215639 1.077282 34996  
## [17] {km.sprinting.2=0,   
## km.sprinting.6=0,   
## sprintkm\_0\_disc=0.0-0.29} => {km.sprinting=0} 0.8232708 0.9958422 0.8267081 1.077064 35208  
## [18] {km.sprinting.2=0,   
## km.sprinting.6=0,   
## injury=0,   
## sprintkm\_0\_disc=0.0-0.29} => {km.sprinting=0} 0.8130992 0.9958189 0.8165131 1.077039 34773  
## [19] {km.sprinting.1=0,   
## km.sprinting.2=0,   
## sprintkm\_0\_disc=0.0-0.29} => {km.sprinting=0} 0.8173549 0.9956987 0.8208858 1.076909 34955  
## [20] {km.sprinting.1=0,   
## km.sprinting.5=0,   
## sprintkm\_0\_disc=0.0-0.29} => {km.sprinting=0} 0.8237619 0.9956758 0.8273395 1.076884 35229  
## [21] {km.sprinting.4=0,   
## km.sprinting.6=0,   
## sprintkm\_0\_disc=0.0-0.29} => {km.sprinting=0} 0.8236683 0.9956753 0.8272459 1.076883 35225  
## [22] {km.sprinting.1=0,   
## km.sprinting.2=0,   
## injury=0,   
## sprintkm\_0\_disc=0.0-0.29} => {km.sprinting=0} 0.8069494 0.9956722 0.8104569 1.076880 34510  
## [23] {km.sprinting.1=0,   
## km.sprinting.5=0,   
## injury=0,   
## sprintkm\_0\_disc=0.0-0.29} => {km.sprinting=0} 0.8130992 0.9956478 0.8166534 1.076854 34773  
## [24] {km.sprinting.4=0,   
## km.sprinting.6=0,   
## injury=0,   
## sprintkm\_0\_disc=0.0-0.29} => {km.sprinting=0} 0.8134266 0.9956211 0.8170042 1.076825 34787  
## [25] {km.sprinting.2=0,   
## sprintkm\_0\_disc=0.0-0.29} => {km.sprinting=0} 0.8713230 0.9953788 0.8753683 1.076563 37263  
## [26] {km.sprinting.3=0,   
## km.sprinting.6=0,   
## sprintkm\_0\_disc=0.0-0.29} => {km.sprinting=0} 0.8209559 0.9953788 0.8247673 1.076563 35109  
## [27] {km.sprinting.1=0,   
## km.sprinting.4=0,   
## sprintkm\_0\_disc=0.0-0.29} => {km.sprinting=0} 0.8202778 0.9953750 0.8240892 1.076559 35080  
## [28] {km.sprinting.2=0,   
## injury=0,   
## sprintkm\_0\_disc=0.0-0.29} => {km.sprinting=0} 0.8600991 0.9953726 0.8640976 1.076556 36783  
## [29] {km.sprinting.5=0,   
## sprintkm\_0\_disc=0.0-0.29} => {km.sprinting=0} 0.8731001 0.9953616 0.8771688 1.076544 37339  
## [30] {km.sprinting.5=0,   
## injury=0,   
## sprintkm\_0\_disc=0.0-0.29} => {km.sprinting=0} 0.8616892 0.9953542 0.8657111 1.076536 36851  
## [31] {km.sprinting.3=0,   
## km.sprinting.6=0,   
## injury=0,   
## sprintkm\_0\_disc=0.0-0.29} => {km.sprinting=0} 0.8108077 0.9953498 0.8145957 1.076531 34675  
## [32] {km.sprinting.1=0,   
## km.sprinting.4=0,   
## injury=0,   
## sprintkm\_0\_disc=0.0-0.29} => {km.sprinting=0} 0.8097320 0.9953150 0.8135435 1.076494 34629  
## [33] {km.Z5.T1.T2.5=0,   
## sprintkm\_0\_disc=0.0-0.29} => {km.sprinting=0} 0.8131927 0.9952779 0.8170509 1.076454 34777  
## [34] {km.Z5.T1.T2.5=0,   
## injury=0,   
## sprintkm\_0\_disc=0.0-0.29} => {km.sprinting=0} 0.8030679 0.9952763 0.8068793 1.076452 34344  
## [35] {km.sprinting.1=0,   
## km.sprinting.3=0,   
## sprintkm\_0\_disc=0.0-0.29} => {km.sprinting=0} 0.8221718 0.9952165 0.8261236 1.076387 35161  
## [36] {km.sprinting.1=0,   
## km.sprinting.3=0,   
## injury=0,   
## sprintkm\_0\_disc=0.0-0.29} => {km.sprinting=0} 0.8115793 0.9951829 0.8155076 1.076351 34708  
## [37] {km.Z5.T1.T2.2=0,   
## sprintkm\_0\_disc=0.0-0.29} => {km.sprinting=0} 0.8073002 0.9951002 0.8112753 1.076261 34525  
## [38] {km.sprinting.4=0,   
## sprintkm\_0\_disc=0.0-0.29} => {km.sprinting=0} 0.8691484 0.9950742 0.8734509 1.076233 37170  
## [39] {km.sprinting.4=0,   
## injury=0,   
## sprintkm\_0\_disc=0.0-0.29} => {km.sprinting=0} 0.8578076 0.9950363 0.8620867 1.076192 36685  
## [40] {km.sprinting.3=0,   
## sprintkm\_0\_disc=0.0-0.29} => {km.sprinting=0} 0.8684469 0.9949103 0.8728897 1.076056 37140  
## [41] {km.sprinting.3=0,   
## injury=0,   
## sprintkm\_0\_disc=0.0-0.29} => {km.sprinting=0} 0.8572464 0.9948711 0.8616658 1.076014 36661  
## [42] {km.sprinting.1=0,   
## km.sprinting.6=0,   
## sprintkm\_0\_disc=0.0-0.29} => {km.sprinting=0} 0.8203012 0.9948388 0.8245569 1.075979 35081  
## [43] {km.sprinting.1=0,   
## km.sprinting.6=0,   
## injury=0,   
## sprintkm\_0\_disc=0.0-0.29} => {km.sprinting=0} 0.8101295 0.9948029 0.8143619 1.075940 34646  
## [44] {km.sprinting.6=0,   
## sprintkm\_0\_disc=0.0-0.29} => {km.sprinting=0} 0.8644718 0.9945658 0.8691952 1.075683 36970  
## [45] {km.Z5.T1.T2.3=0,   
## sprintkm\_0\_disc=0.0-0.29} => {km.sprinting=0} 0.8101062 0.9945457 0.8145489 1.075662 34645  
## [46] {km.sprinting.6=0,   
## injury=0,   
## sprintkm\_0\_disc=0.0-0.29} => {km.sprinting=0} 0.8536922 0.9945247 0.8583922 1.075639 36509  
## [47] {km.Z3.4=0,   
## sprintkm\_0\_disc=0.0-0.29} => {km.sprinting=0} 0.8226395 0.9944597 0.8272226 1.075569 35181  
## [48] {km.Z3.4=0,   
## injury=0,   
## sprintkm\_0\_disc=0.0-0.29} => {km.sprinting=0} 0.8119067 0.9944438 0.8164430 1.075552 34722  
## [49] {km.Z3.4.2=0,   
## sprintkm\_0\_disc=0.0-0.29} => {km.sprinting=0} 0.8229201 0.9943211 0.8276201 1.075419 35193  
## [50] {km.Z3.4.2=0,   
## injury=0,   
## sprintkm\_0\_disc=0.0-0.29} => {km.sprinting=0} 0.8126549 0.9942782 0.8173315 1.075372 34754

Subset of rules with “injury” on right-hand side was created to see if there is interesting rules point to injuries and non-injuries. Unfortunately, similar results were formulated

rules\_subset <- subset(rules, (rhs %oin% c("injury=1", "injury=0")))  
arules::inspect(rules\_subset[1:10])

## lhs rhs support confidence coverage lift count  
## [1] {strength.training.1=0,   
## sprintkm\_0\_disc=0.0-0.29} => {injury=0} 0.8141982 0.9881658 0.8239489 1.001823 34820  
## [2] {km.sprinting=0,   
## strength.training.1=0} => {injury=0} 0.8093579 0.9881523 0.8190619 1.001809 34613  
## [3] {km.sprinting=0,   
## strength.training.1=0,   
## sprintkm\_0\_disc=0.0-0.29} => {injury=0} 0.8093579 0.9881523 0.8190619 1.001809 34613  
## [4] {km.Z5.T1.T2=0,   
## km.sprinting=0} => {injury=0} 0.8052425 0.9881209 0.8149231 1.001777 34437  
## [5] {km.Z5.T1.T2=0,   
## km.sprinting=0,   
## sprintkm\_0\_disc=0.0-0.29} => {injury=0} 0.8052425 0.9881209 0.8149231 1.001777 34437  
## [6] {km.Z5.T1.T2=0,   
## sprintkm\_0\_disc=0.0-0.29} => {injury=0} 0.8078614 0.9881024 0.8175887 1.001759 34549  
## [7] {strength.training.1=0,   
## km.sprinting.6=0} => {injury=0} 0.8156012 0.9879620 0.8255390 1.001616 34880  
## [8] {km.sprinting.6=0,   
## strength.training.6=0} => {injury=0} 0.8050788 0.9879200 0.8149231 1.001574 34430  
## [9] {strength.training.1=0,   
## km.sprinting.2=0} => {injury=0} 0.8098489 0.9878776 0.8197867 1.001531 34634  
## [10] {strength.training.1=0,   
## km.sprinting.4=0} => {injury=0} 0.8120703 0.9877979 0.8221017 1.001450 34729

Even with lowering the support and confidence we found no rules indicating injury.

#set rhs  
dailyrhs <- apriori(dailytrans, parameter = list(support=.01, confidence=.02, maxlen=3), appearance = list(rhs="injury=1"))

## Apriori  
##   
## Parameter specification:  
## confidence minval smax arem aval originalSupport maxtime support minlen  
## 0.02 0.1 1 none FALSE TRUE 5 0.01 1  
## maxlen target ext  
## 3 rules TRUE  
##   
## Algorithmic control:  
## filter tree heap memopt load sort verbose  
## 0.1 TRUE TRUE FALSE TRUE 2 TRUE  
##   
## Absolute minimum support count: 427   
##   
## set item appearances ...[1 item(s)] done [0.00s].  
## set transactions ...[10805 item(s), 42766 transaction(s)] done [4.91s].  
## sorting and recoding items ... [651 item(s)] done [0.16s].  
## creating transaction tree ... done [0.09s].  
## checking subsets of size 1 2 3

## Warning in apriori(dailytrans, parameter = list(support = 0.01, confidence  
## = 0.02, : Mining stopped (maxlen reached). Only patterns up to a length of 3  
## returned!

## done [6.46s].  
## writing ... [0 rule(s)] done [0.03s].  
## creating S4 object ... done [0.03s].

rulesrhs <- sort(dailyrhs, decreasing = TRUE, by="lift")  
rulesrhs

## set of 0 rules

Injured only

#creating a subset to run ARM on only the injured  
injuredDailyARM <- injuredDaily  
injuredDailyARM<-lapply(injuredDailyARM, function(x){as.factor(x)})  
injuredDailyARM = as.data.frame(injuredDailyARM)

#turn data into transactions  
injureddailytrans = as(injuredDailyARM, "transactions")  
  
str(injureddailytrans)

## Formal class 'transactions' [package "arules"] with 3 slots  
## ..@ data :Formal class 'ngCMatrix' [package "Matrix"] with 5 slots  
## .. .. ..@ i : int [1:43725] 1 3 162 198 240 263 274 296 382 478 ...  
## .. .. ..@ p : int [1:584] 0 75 150 225 300 375 450 525 600 675 ...  
## .. .. ..@ Dim : int [1:2] 4677 583  
## .. .. ..@ Dimnames:List of 2  
## .. .. .. ..$ : NULL  
## .. .. .. ..$ : NULL  
## .. .. ..@ factors : list()  
## ..@ itemInfo :'data.frame': 4677 obs. of 3 variables:  
## .. ..$ labels : chr [1:4677] "nr..sessions=0" "nr..sessions=1" "nr..sessions=2" "total.km=0" ...  
## .. ..$ variables: Factor w/ 75 levels "Athlete.ID","Date",..: 32 32 32 68 68 68 68 68 68 68 ...  
## .. ..$ levels : Factor w/ 1282 levels "-0.01","0","0.0-0.29",..: 2 108 495 2 23 108 117 495 506 508 ...  
## ..@ itemsetInfo:'data.frame': 583 obs. of 1 variable:  
## .. ..$ transactionID: chr [1:583] "1" "2" "3" "4" ...

injuredrules <- apriori(injureddailytrans, parameter = list(supp=.1, conf = .45, minlen=3))

## Apriori  
##   
## Parameter specification:  
## confidence minval smax arem aval originalSupport maxtime support minlen  
## 0.45 0.1 1 none FALSE TRUE 5 0.1 3  
## maxlen target ext  
## 10 rules TRUE  
##   
## Algorithmic control:  
## filter tree heap memopt load sort verbose  
## 0.1 TRUE TRUE FALSE TRUE 2 TRUE  
##   
## Absolute minimum support count: 58   
##   
## set item appearances ...[0 item(s)] done [0.00s].  
## set transactions ...[4655 item(s), 583 transaction(s)] done [0.06s].  
## sorting and recoding items ... [99 item(s)] done [0.00s].  
## creating transaction tree ... done [0.00s].  
## checking subsets of size 1 2 3 4 5

## Warning in apriori(injureddailytrans, parameter = list(supp = 0.1, conf =  
## 0.45, : Mining stopped (time limit reached). Only patterns up to a length of 5  
## returned!

## done [7.81s].  
## writing ... [24928170 rule(s)] done [9.23s].  
## creating S4 object ... done [41.76s].

injuredrules <- sort(injuredrules, decreasing = TRUE, by="lift")  
arules::inspect(injuredrules[1:10])

## lhs rhs support confidence coverage lift count  
## [1] {perceived.trainingSuccess=0,   
## nr..sessions.3=1} => {perceived.trainingSuccess.3=0} 0.1097770 1 0.1097770 7.024096 64  
## [2] {nr..sessions.3=1,   
## perceived.trainingSuccess.4=0} => {perceived.trainingSuccess.3=0} 0.1063465 1 0.1063465 7.024096 62  
## [3] {nr..sessions.3=1,   
## perceived.trainingSuccess.6=0} => {perceived.trainingSuccess.3=0} 0.1132075 1 0.1132075 7.024096 66  
## [4] {nr..sessions.1=1,   
## perceived.trainingSuccess.6=0} => {perceived.trainingSuccess.1=0} 0.1063465 1 0.1063465 7.024096 62  
## [5] {nr..sessions=1,   
## perceived.trainingSuccess=0,   
## nr..sessions.3=1} => {perceived.trainingSuccess.3=0} 0.1012007 1 0.1012007 7.024096 59  
## [6] {perceived.trainingSuccess=0,   
## nr..sessions.3=1,   
## strength.training.6=0} => {perceived.trainingSuccess.3=0} 0.1046312 1 0.1046312 7.024096 61  
## [7] {perceived.trainingSuccess=0,   
## nr..sessions.3=1,   
## strength.training.3=0} => {perceived.trainingSuccess.3=0} 0.1029160 1 0.1029160 7.024096 60  
## [8] {perceived.trainingSuccess=0,   
## nr..sessions.3=1,   
## hours.alternative.6=0} => {perceived.trainingSuccess.3=0} 0.1012007 1 0.1012007 7.024096 59  
## [9] {perceived.trainingSuccess=0,   
## hours.alternative.1=0,   
## nr..sessions.3=1} => {perceived.trainingSuccess.3=0} 0.1012007 1 0.1012007 7.024096 59  
## [10] {perceived.trainingSuccess=0,   
## nr..sessions.3=1,   
## strength.training.5=0} => {perceived.trainingSuccess.3=0} 0.1012007 1 0.1012007 7.024096 59

#Subsetting to see the consequence (RHS) of certain training. Possible that number of sessions a day and sprinting had an impact.  
  
rules\_injsubset <- subset(injuredrules, (rhs %oin% c("injury=1", "injury=0")))  
arules::inspect(rules\_injsubset[1:10])

## lhs rhs support confidence coverage lift count  
## [1] {nr..sessions.3=2,   
## km.sprinting.4=0} => {injury=1} 0.1012007 1 0.1012007 1 59  
## [2] {nr..sessions.1=2,   
## km.Z3.4.5=0} => {injury=1} 0.1029160 1 0.1029160 1 60  
## [3] {nr..sessions.1=2,   
## strength.training.4=0} => {injury=1} 0.1012007 1 0.1012007 1 59  
## [4] {strength.training=0,   
## nr..sessions.1=2} => {injury=1} 0.1012007 1 0.1012007 1 59  
## [5] {km.sprinting=0,   
## nr..sessions.1=2} => {injury=1} 0.1012007 1 0.1012007 1 59  
## [6] {nr..sessions.1=2,   
## sprintkm\_0\_disc=0.0-0.29} => {injury=1} 0.1012007 1 0.1012007 1 59  
## [7] {nr..sessions.1=2,   
## km.sprinting.2=0} => {injury=1} 0.1046312 1 0.1046312 1 61  
## [8] {nr..sessions.1=2,   
## km.sprinting.4=0} => {injury=1} 0.1012007 1 0.1012007 1 59  
## [9] {nr..sessions.1=2,   
## km.sprinting.5=0} => {injury=1} 0.1029160 1 0.1029160 1 60  
## [10] {strength.training.1=0,   
## nr..sessions.4=2} => {injury=1} 0.1046312 1 0.1046312 1 61

# Prepare data for predicitive models

Make a copy of the dataset which we will work off of.

dailyPM <- daily

Subset by KM variable types. We removed all of the non-kilometer variables so we could focus on time-on-feet data for certain models. This would also allow for faster processing time on the more time consuming models.

dailyKM <- dailyPM[,c(2, 3,4,5,12,13,14,15,22,23,24,25,32,33,34,35,42,43,44,45,52,53,54,55,62,63,64,65,71)]

Set seed and randomize the full and clean daily dataset for reference use.

set.seed(123)  
randomize <- runif(nrow(daily))  
dailyr<-daily[order(randomize), ]

Set seed and randomize for the kilometer subset created to run models.

set.seed(123)  
randomize <- runif(nrow(dailyKM))  
dailyKMr<-dailyKM[order(randomize), ]

Create test and training set for full daily dataset.

dailyr$injury <- as.factor(daily$injury)  
  
dailytrain <- dailyr[1:34213, ]  
dailytest <- dailyr[34214:42766, ]  
dailytraintarget <- dailyr[1:34213, 71]  
dailytesttarget <- dailyr[34214:42766, 71]

Create test and training set for kilometer only dataset.

dailyKM$injury <- as.factor(dailyKM$injury)  
  
dailytrainKM <- dailyKMr[1:34213, ]  
dailytestKM <- dailyKMr[34214:42766, ]  
dailytraintargetKM <- dailyKMr[1:34213, 29]  
dailytesttargetKM <- dailyKMr[34214:42766, 29]  
  
range(dailytestKM$total.km)

## [1] 0.0 52.5

range(dailytestKM$km.sprinting)

## [1] 0.0 15.9

str(dailytesttargetKM)

## Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...

str(dailyKM)

## 'data.frame': 42766 obs. of 29 variables:  
## $ total.km : num 5.8 0 0 0 0 16.4 0 5.2 0 0 ...  
## $ km.Z3.4 : num 0 0 0 0 0 10 0 0 0 0 ...  
## $ km.Z5.T1.T2 : num 0.6 0 0 0 0 0 0 0.5 0 0 ...  
## $ km.sprinting : num 1.2 0 0 0 0 0 0 1.2 0 0 ...  
## $ total.km.1 : num 0 0 0 0 16.4 0 5.2 0 0 17.6 ...  
## $ km.Z3.4.1 : num 0 0 0 0 10 0 0 0 0 7.2 ...  
## $ km.Z5.T1.T2.1 : num 0 0 0 0 0 0 0.5 0 0 0 ...  
## $ km.sprinting.1: num 0 0 0 0 0 0 1.2 0 0 0 ...  
## $ total.km.2 : num 0 0 0 16.4 0 5.2 0 0 17.6 0 ...  
## $ km.Z3.4.2 : num 0 0 0 10 0 0 0 0 7.2 0 ...  
## $ km.Z5.T1.T2.2 : num 0 0 0 0 0 0.5 0 0 0 0 ...  
## $ km.sprinting.2: num 0 0 0 0 0 1.2 0 0 0 0 ...  
## $ total.km.3 : num 0 0 16.4 0 5.2 0 0 17.6 0 10.5 ...  
## $ km.Z3.4.3 : num 0 0 10 0 0 0 0 7.2 0 6.5 ...  
## $ km.Z5.T1.T2.3 : num 0 0 0 0 0.5 0 0 0 0 0 ...  
## $ km.sprinting.3: num 0 0 0 0 1.2 0 0 0 0 0 ...  
## $ total.km.4 : num 0 16.4 0 5.2 0 0 17.6 0 10.5 0 ...  
## $ km.Z3.4.4 : num 0 10 0 0 0 0 7.2 0 6.5 0 ...  
## $ km.Z5.T1.T2.4 : num 0 0 0 0.5 0 0 0 0 0 0 ...  
## $ km.sprinting.4: num 0 0 0 1.2 0 0 0 0 0 0 ...  
## $ total.km.5 : num 16.4 0 5.2 0 0 17.6 0 10.5 0 4.8 ...  
## $ km.Z3.4.5 : num 10 0 0 0 0 7.2 0 6.5 0 0.6 ...  
## $ km.Z5.T1.T2.5 : num 0 0 0.5 0 0 0 0 0 0 0 ...  
## $ km.sprinting.5: num 0 0 1.2 0 0 0 0 0 0 1.2 ...  
## $ total.km.6 : num 0 5.2 0 0 17.6 0 10.5 0 4.8 0 ...  
## $ km.Z3.4.6 : num 0 0 0 0 7.2 0 6.5 0 0.6 0 ...  
## $ km.Z5.T1.T2.6 : num 0 0.5 0 0 0 0 0 0 0 0 ...  
## $ km.sprinting.6: num 0 1.2 0 0 0 0 0 0 1.2 0 ...  
## $ injury : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...

Normalize This proved unsuccessful with the full daily dataset because of the wide range of variable type (hours, binary, Km)

min\_max\_func <- function(x) {  
 a <- (x - min(x))  
 b <- (max(x) - min(x))  
 return(a/b)  
}  
  
  
dailytrain\_n <- as.data.frame(lapply(dailytrain[,1:70], min\_max\_func))  
dailytrain\_n

dailytest\_n <- as.data.frame(lapply(dailytest[,1:70], min\_max\_func))  
dailytest\_n

head(dailytrain)

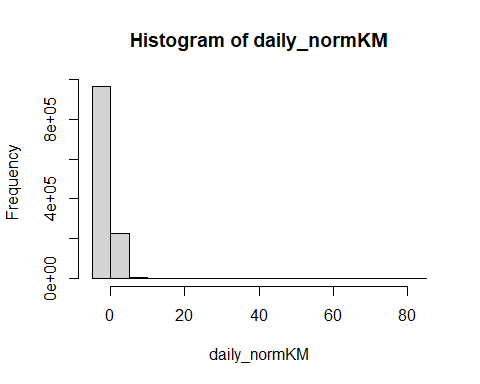
## nr..sessions total.km km.Z3.4 km.Z5.T1.T2 km.sprinting strength.training  
## 19296 1 13.0 0 7.0 0 0  
## 15829 1 0.0 0 0.0 0 0  
## 3934 1 15.5 0 1.5 0 0  
## 9036 2 37.0 5 0.0 0 0  
## 39632 1 0.0 0 0.0 0 1  
## 13550 2 6.0 0 0.0 0 1  
## hours.alternative perceived.exertion perceived.trainingSuccess  
## 19296 0 0.39 0.82  
## 15829 2 0.46 0.49  
## 3934 0 0.15 0.00  
## 9036 0 0.12 0.00  
## 39632 0 0.30 0.93  
## 13550 0 0.56 0.64  
## perceived.recovery nr..sessions.1 total.km.1 km.Z3.4.1 km.Z5.T1.T2.1  
## 19296 0.27 1 10.0 0 0  
## 15829 0.44 1 0.0 0 0  
## 3934 0.14 1 16.0 0 0  
## 9036 0.14 2 26.0 0 0  
## 39632 0.21 1 16.7 0 8  
## 13550 0.59 0 0.0 0 0  
## km.sprinting.1 strength.training.1 hours.alternative.1  
## 19296 0 0 0  
## 15829 0 0 8  
## 3934 1 0 0  
## 9036 0 0 0  
## 39632 0 0 0  
## 13550 0 0 0  
## perceived.exertion.1 perceived.trainingSuccess.1 perceived.recovery.1  
## 19296 0.26 0.80 0.24  
## 15829 0.48 0.50 0.46  
## 3934 0.14 0.00 0.13  
## 9036 0.10 0.00 0.14  
## 39632 0.89 1.00 0.27  
## 13550 -0.01 -0.01 -0.01  
## nr..sessions.2 total.km.2 km.Z3.4.2 km.Z5.T1.T2.2 km.sprinting.2  
## 19296 1 16.4 0 6.4 0  
## 15829 1 0.0 0 0.0 0  
## 3934 1 3.0 0 0.0 0  
## 9036 1 10.0 0 0.0 0  
## 39632 1 13.5 0 0.0 0  
## 13550 0 0.0 0 0.0 0  
## strength.training.2 hours.alternative.2 perceived.exertion.2  
## 19296 0 0 0.34  
## 15829 0 4 0.48  
## 3934 0 0 0.15  
## 9036 0 0 0.09  
## 39632 0 0 0.35  
## 13550 0 0 -0.01  
## perceived.trainingSuccess.2 perceived.recovery.2 nr..sessions.3  
## 19296 0.83 0.25 1  
## 15829 0.49 0.42 1  
## 3934 0.00 0.14 2  
## 9036 0.00 0.12 1  
## 39632 0.83 0.13 1  
## 13550 -0.01 -0.01 1  
## total.km.3 km.Z3.4.3 km.Z5.T1.T2.3 km.sprinting.3 strength.training.3  
## 19296 14.0 0 0 0.0 0  
## 15829 0.0 0 0 0.0 0  
## 3934 16.0 0 0 0.0 0  
## 9036 22.5 0 10 0.5 0  
## 39632 0.0 0 0 0.0 0  
## 13550 9.0 0 0 0.0 0  
## hours.alternative.3 perceived.exertion.3 perceived.trainingSuccess.3  
## 19296 0 0.34 0.70  
## 15829 8 0.49 0.46  
## 3934 0 0.14 0.00  
## 9036 0 0.09 0.00  
## 39632 1 0.20 1.00  
## 13550 0 0.21 0.66  
## perceived.recovery.3 nr..sessions.4 total.km.4 km.Z3.4.4 km.Z5.T1.T2.4  
## 19296 0.31 1 17.7 0 6  
## 15829 0.45 0 0.0 0 0  
## 3934 0.14 1 4.0 1 0  
## 9036 0.14 1 30.5 0 0  
## 39632 0.14 1 17.8 8 0  
## 13550 0.58 1 11.5 0 0  
## km.sprinting.4 strength.training.4 hours.alternative.4  
## 19296 0 0 0  
## 15829 0 0 0  
## 3934 0 0 0  
## 9036 0 0 0  
## 39632 0 0 0  
## 13550 0 0 0  
## perceived.exertion.4 perceived.trainingSuccess.4 perceived.recovery.4  
## 19296 0.36 0.76 0.37  
## 15829 -0.01 -0.01 -0.01  
## 3934 0.12 0.00 0.14  
## 9036 0.14 0.00 0.12  
## 39632 0.44 1.00 0.13  
## 13550 0.26 0.63 0.61  
## nr..sessions.5 total.km.5 km.Z3.4.5 km.Z5.T1.T2.5 km.sprinting.5  
## 19296 0 0.0 0 0.0 0.0  
## 15829 1 0.0 0 0.0 0.0  
## 3934 1 10.0 0 1.5 0.5  
## 9036 2 26.2 0 0.0 0.0  
## 39632 1 0.0 0 0.0 0.0  
## 13550 1 13.5 0 0.0 0.0  
## strength.training.5 hours.alternative.5 perceived.exertion.5  
## 19296 0 0 -0.01  
## 15829 0 3 0.46  
## 3934 0 0 0.15  
## 9036 0 0 0.09  
## 39632 1 0 0.28  
## 13550 0 0 0.35  
## perceived.trainingSuccess.5 perceived.recovery.5 nr..sessions.6  
## 19296 -0.01 -0.01 1  
## 15829 0.43 0.46 0  
## 3934 0.00 0.15 2  
## 9036 0.00 0.14 1  
## 39632 1.00 0.14 1  
## 13550 0.79 0.27 1  
## total.km.6 km.Z3.4.6 km.Z5.T1.T2.6 km.sprinting.6 strength.training.6  
## 19296 15.7 8.7 0.0 0.0 0  
## 15829 0.0 0.0 0.0 0.0 0  
## 3934 14.5 0.0 0.0 1.5 1  
## 9036 23.2 13.2 0.0 0.0 0  
## 39632 15.1 0.0 5.2 0.0 0  
## 13550 10.5 0.0 4.0 0.0 0  
## hours.alternative.6 perceived.exertion.6 perceived.trainingSuccess.6  
## 19296 0 0.38 0.76  
## 15829 0 -0.01 -0.01  
## 3934 0 0.13 0.00  
## 9036 0 0.13 0.00  
## 39632 0 0.69 0.86  
## 13550 0 0.69 0.13  
## perceived.recovery.6 injury total\_km\_to\_inj sprintkm\_0\_disc  
## 19296 0.25 0 86.8 0.0-0.29  
## 15829 -0.01 0 0.0 0.0-0.29  
## 3934 0.13 0 79.0 0.0-0.29  
## 9036 0.12 0 175.4 0.0-0.29  
## 39632 0.19 0 63.1 0.0-0.29  
## 13550 0.63 0 50.5 0.0-0.29

Normalize the kilometer dataset.

daily\_normKM <- scale(dailyKM[,1:28])  
head(daily\_normKM)

## total.km km.Z3.4 km.Z5.T1.T2 km.sprinting total.km.1 km.Z3.4.1  
## [1,] -0.1656833 -0.2983103 0.01107636 2.3309818 -0.9399385 -0.2983322  
## [2,] -0.9417882 -0.2983103 -0.32006077 -0.1510215 -0.9399385 -0.2983322  
## [3,] -0.9417882 -0.2983103 -0.32006077 -0.1510215 -0.9399385 -0.2983322  
## [4,] -0.9417882 -0.2983103 -0.32006077 -0.1510215 -0.9399385 -0.2983322  
## [5,] -0.9417882 -0.2983103 -0.32006077 -0.1510215 1.2546165 4.0288623  
## [6,] 1.2527154 4.0163917 -0.32006077 -0.1510215 -0.9399385 -0.2983322  
## km.Z5.T1.T2.1 km.sprinting.1 total.km.2 km.Z3.4.2 km.Z5.T1.T2.2  
## [1,] -0.3184204 -0.1499709 -0.9422259 -0.2985399 -0.31987621  
## [2,] -0.3184204 -0.1499709 -0.9422259 -0.2985399 -0.31987621  
## [3,] -0.3184204 -0.1499709 -0.9422259 -0.2985399 -0.31987621  
## [4,] -0.3184204 -0.1499709 1.2529994 3.9957953 -0.31987621  
## [5,] -0.3184204 -0.1499709 -0.9422259 -0.2985399 -0.31987621  
## [6,] -0.3184204 -0.1499709 -0.2461789 -0.2985399 -0.04337644  
## km.sprinting.2 total.km.3 km.Z3.4.3 km.Z5.T1.T2.3 km.sprinting.3  
## [1,] -0.1505977 -0.9430837 -0.299184 -0.31930305 -0.1508219  
## [2,] -0.1505977 -0.9430837 -0.299184 -0.31930305 -0.1508219  
## [3,] -0.1505977 1.2538926 4.020470 -0.31930305 -0.1508219  
## [4,] -0.1505977 -0.9430837 -0.299184 -0.31930305 -0.1508219  
## [5,] -0.1505977 -0.2464814 -0.299184 -0.04317593 2.3389030  
## [6,] 2.3204049 -0.9430837 -0.299184 -0.31930305 -0.1508219  
## total.km.4 km.Z3.4.4 km.Z5.T1.T2.4 km.sprinting.4 total.km.5 km.Z3.4.5  
## [1,] -0.9438606 -0.2991491 -0.31966074 -0.1504282 1.2509603 3.9967033  
## [2,] 1.2539897 4.0023866 -0.31966074 -0.1504282 -0.9443371 -0.2996205  
## [3,] -0.9438606 -0.2991491 -0.31966074 -0.1504282 -0.2482672 -0.2996205  
## [4,] -0.2469812 -0.2991491 -0.04303635 2.3509018 -0.9443371 -0.2996205  
## [5,] -0.9438606 -0.2991491 -0.31966074 -0.1504282 -0.9443371 -0.2996205  
## [6,] -0.9438606 -0.2991491 -0.31966074 -0.1504282 1.4115918 2.7937326  
## km.Z5.T1.T2.5 km.sprinting.5 total.km.6 km.Z3.4.6 km.Z5.T1.T2.6  
## [1,] -0.32020940 -0.1484432 -0.9439297 -0.2990868 -0.31983154  
## [2,] -0.32020940 -0.1484432 -0.2478281 -0.2990868 -0.04427933  
## [3,] -0.04420153 2.3152497 -0.9439297 -0.2990868 -0.31983154  
## [4,] -0.32020940 -0.1484432 -0.9439297 -0.2990868 -0.31983154  
## [5,] -0.32020940 -0.1484432 1.4121065 2.7992281 -0.31983154  
## [6,] -0.32020940 -0.1484432 -0.9439297 -0.2990868 -0.31983154  
## km.sprinting.6  
## [1,] -0.1500857  
## [2,] 2.3308382  
## [3,] -0.1500857  
## [4,] -0.1500857  
## [5,] -0.1500857  
## [6,] -0.1500857

hist(daily\_normKM)



# Decision Tree

Data prep

set.seed(123)  
noinjuries <- daily %>% filter(injury == "0")  
injuries <- daily %>% filter(injury == "1")  
sample <- sample(nrow(noinjuries), 583, replace = FALSE)  
  
samplenoinjuries <- noinjuries[sample,]  
  
#new dataframe to use for training   
dailysample <- rbind(samplenoinjuries, injuries)  
  
  
# Create the test and train dataset  
trainlist <- createDataPartition(y=dailysample$injury, p=.7, list=FALSE)  
train\_daily <- dailysample[trainlist,]  
test\_daily <- dailysample[-trainlist,]

# Decision Tree 1

DT1 <- train(injury~.,  
 method="rpart",   
 data=train\_daily,  
 tuneLength = 10)  
   
  
print(DT1)

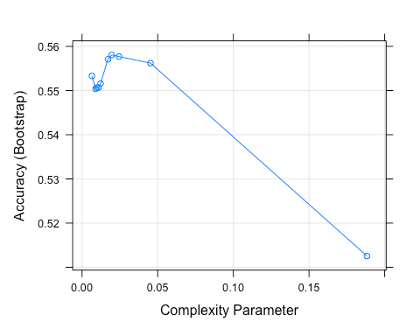
## CART   
##   
## 818 samples  
 ## 72 predictor  
 ## 2 classes: '0', '1'   
##   
## No pre-processing  
 ## Resampling: Bootstrapped (25 reps)   
## Summary of sample sizes: 818, 818, 818, 818, 818, 818, ...   
## Resampling results across tuning parameters:  
 ##   
## cp Accuracy Kappa   
## 0.006519967 0.5532992 0.10700487  
 ## 0.008964955 0.5503937 0.10168411  
 ## 0.009779951 0.5506683 0.10279844  
 ## 0.011002445 0.5506667 0.10241945  
 ## 0.012224939 0.5515883 0.10413055  
 ## 0.017114914 0.5570707 0.11597581  
 ## 0.019559902 0.5580677 0.11882420  
 ## 0.024449878 0.5576946 0.11743520  
 ## 0.045232274 0.5562130 0.11337211  
 ## 0.188264059 0.5125388 0.03692476  
 ##   
## Accuracy was used to select the optimal model using the largest value.  
 ## The final value used for the model was cp = 0.0195599.

testlabels <- test\_daily$injury  
 test\_daily <- test\_daily %>% select(-injury)  
 DT\_Pred <- predict(DT1, test\_daily, type = 'raw')  
 confusionMatrix(DT\_Pred, testlabels, dnn = c("Prediction", "Actual"))

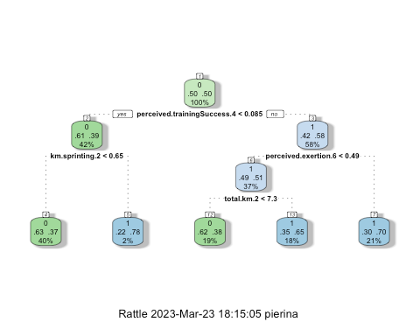
## Confusion Matrix and Statistics  
 ##   
## Actual  
 ## Prediction 0 1  
 ## 0 116 92  
 ## 1 58 82  
 ##   
## Accuracy : 0.569   
## 95% CI : (0.5151, 0.6217)  
 ## No Information Rate : 0.5   
## P-Value [Acc > NIR] : 0.005823   
##   
## Kappa : 0.1379   
##   
## Mcnemar's Test P-Value : 0.007051   
##   
## Sensitivity : 0.6667   
## Specificity : 0.4713   
## Pos Pred Value : 0.5577   
## Neg Pred Value : 0.5857   
## Prevalence : 0.5000   
## Detection Rate : 0.3333   
## Detection Prevalence : 0.5977   
## Balanced Accuracy : 0.5690   
##   
## 'Positive' Class : 0   
##

# Plot the decision tree

plot(DT1)



fancyRpartPlot(DT1$finalModel)



# Decision Tree 2

# Training the C5.0 model with smaller sample as well

DT2 <-C5.0(train\_daily %>% select (-injury), train\_daily$injury)  
 DT2

##   
## Call:  
 ## C5.0.default(x = train\_daily %>% select(-injury), y = train\_daily$injury)  
 ##   
## Classification Tree  
 ## Number of samples: 818   
## Number of predictors: 72   
##   
## Tree size: 125   
##   
## Non-standard options: attempt to group attributes

summary(DT2)

##   
## Call:  
 ## C5.0.default(x = train\_daily %>% select(-injury), y = train\_daily$injury)  
 ##   
##   
## C5.0 [Release 2.07 GPL Edition] Thu Mar 23 18:15:06 2023  
 ## -------------------------------  
 ##   
## Class specified by attribute `outcome'  
 ##   
## Read 818 cases (73 attributes) from undefined.data  
 ##   
## Decision tree:  
 ##   
## nr..sessions.4 <= 0:  
 ## :...km.sprinting.2 > 0.5:  
 ## : :...perceived.trainingSuccess.5 <= 0.78: 1 (8)  
 ## : : perceived.trainingSuccess.5 > 0.78: 0 (2)  
 ## : km.sprinting.2 <= 0.5:  
 ## : :...perceived.exertion.2 > 0.67:  
 ## : :...perceived.trainingSuccess.6 <= 0.77: 1 (12)  
 ## : : perceived.trainingSuccess.6 > 0.77: 0 (3)  
 ## : perceived.exertion.2 <= 0.67:  
 ## : :...km.Z3.4.6 > 3:  
 ## : :...total.km.6 <= 11.5: 0 (4/1)  
 ## : : total.km.6 > 11.5: 1 (6)  
 ## : km.Z3.4.6 <= 3:  
 ## : :...km.sprinting.5 > 0: 0 (9)  
 ## : km.sprinting.5 <= 0:  
 ## : :...km.Z5.T1.T2 > 0.5:  
 ## : :...nr..sessions.5 <= 0: 0 (3)  
 ## : : nr..sessions.5 > 0: 1 (6)  
 ## : km.Z5.T1.T2 <= 0.5:  
 ## : :...strength.training.1 > 0: 0 (7)  
 ## : strength.training.1 <= 0:  
 ## : :...km.Z5.T1.T2.6 > 1.2: 0 (11)  
 ## : km.Z5.T1.T2.6 <= 1.2:  
 ## : :...perceived.recovery.6 > 0.32:  
 ## : :...perceived.recovery > 0.42: 0 (4)  
 ## : : perceived.recovery <= 0.42:  
 ## : : :...nr..sessions.5 <= 0: 0 (4/1)  
 ## : : nr..sessions.5 > 0: 1 (10/1)  
 ## : perceived.recovery.6 <= 0.32:  
 ## : :...km.Z3.4.1 > 0.5: 0 (5)  
 ## : km.Z3.4.1 <= 0.5:  
 ## : :...strength.training.2 > 0:  
 ## : :...total.km.3 <= 10: 0 (5/1)  
 ## : : total.km.3 > 10: 1 (3)  
 ## : strength.training.2 <= 0:  
 ## : :...nr..sessions > 1:  
 ## : :...total.km <= 2.5: 0 (3)  
 ## : : total.km > 2.5: 1 (3)  
 ## : nr..sessions <= 1:  
 ## : :...hours.alternative.2 > 0.25: [S1]  
 ## : hours.alternative.2 <= 0.25:  
 ## : :...km.Z5.T1.T2.2 > 0.8: [S2]  
 ## : km.Z5.T1.T2.2 <= 0.8: [S3]  
 ## nr..sessions.4 > 0:  
 ## :...sprintkm\_0\_disc in {0.3-0.39,0.7-0.89,0.9-0.99,1.2-1.39,1.8-1.99,2.0-2.49,  
 ## : 2.5-2.9,3.0-3.49,3.5-3.9,4.0-4.49,4.5-4.49,5.0-5.49,  
 ## : 5.5-5.49,6.0-6.49,6.5-6.99,7.0-7.49,7.5-7.99,8.0-8.49,  
 ## : 9.0-9.49,9.5-9.99,10.0-10.49,10.5-10.99,11.0-11.99,  
 ## : 12.0-12.99,13.0-13.99,14.0-14.99,15.0-15.99,16.0-16.99,  
 ## : 17.0-17.99,18.0-39.9,40.0+}: 1 (16/1)  
 ## sprintkm\_0\_disc in {1.4-1.59,1.6-1.79}: 0 (4/2)  
 ## sprintkm\_0\_disc = 0.4-0.49:  
 ## :...perceived.recovery.4 <= 0.11: 0 (2)  
 ## : perceived.recovery.4 > 0.11: 1 (5)  
 ## sprintkm\_0\_disc = 0.5-0.69:  
 ## :...km.sprinting.4 > 0.8: 0 (2)  
 ## : km.sprinting.4 <= 0.8:  
 ## : :...total.km.6 <= 26: 1 (14/1)  
 ## : total.km.6 > 26: 0 (2)  
 ## sprintkm\_0\_disc = 1.0-1.19:  
 ## :...nr..sessions > 1: 1 (3)  
 ## : nr..sessions <= 1:  
 ## : :...km.Z5.T1.T2.5 <= 0.6: 0 (6)  
 ## : km.Z5.T1.T2.5 > 0.6: 1 (2)  
 ## sprintkm\_0\_disc = 0.0-0.29:  
 ## :...perceived.exertion.6 > 0.48:  
 ## :...km.Z3.4 > 0: 1 (10)  
 ## : km.Z3.4 <= 0:  
 ## : :...km.Z3.4.1 > 6:  
 ## : :...perceived.recovery.1 <= 0.36: 1 (3/1)  
 ## : : perceived.recovery.1 > 0.36: 0 (6)  
 ## : km.Z3.4.1 <= 6:  
 ## : :...hours.alternative.4 > 1.17: 1 (11)  
 ## : hours.alternative.4 <= 1.17:  
 ## : :...hours.alternative.1 > 1.17: 1 (7)  
 ## : hours.alternative.1 <= 1.17:  
 ## : :...strength.training.3 > 0:  
 ## : :...perceived.trainingSuccess.5 <= 0.37: 1 (8/1)  
 ## : : perceived.trainingSuccess.5 > 0.37: 0 (7)  
 ## : strength.training.3 <= 0:  
 ## : :...nr..sessions.2 <= 0: 1 (21/2)  
 ## : nr..sessions.2 > 0:  
 ## : :...nr..sessions.5 <= 0:  
 ## : :...km.Z3.4.2 <= 2: 0 (9/1)  
 ## : : km.Z3.4.2 > 2: 1 (3)  
 ## : nr..sessions.5 > 0: [S4]  
 ## perceived.exertion.6 <= 0.48:  
 ## :...strength.training.6 > 0:  
 ## :...hours.alternative.1 > 1.33: 0 (3)  
 ## : hours.alternative.1 <= 1.33:  
 ## : :...perceived.exertion > 0.54: 1 (8)  
 ## : perceived.exertion <= 0.54:  
 ## : :...strength.training > 0: 0 (3)  
 ## : strength.training <= 0:  
 ## : :...nr..sessions.2 > 1: 0 (6/1)  
 ## : nr..sessions.2 <= 1:  
 ## : :...km.Z3.4 > 5.8: 0 (2)  
 ## : km.Z3.4 <= 5.8:  
 ## : :...km.sprinting.4 > 1.7: 0 (2)  
 ## : km.sprinting.4 <= 1.7:  
 ## : :...km.sprinting.2 <= 0: 1 (22/2)  
 ## : km.sprinting.2 > 0:  
 ## : :...perceived.exertion.1 <= 0.1: 1 (2)  
 ## : perceived.exertion.1 > 0.1: 0 (2)  
 ## strength.training.6 <= 0:  
 ## :...strength.training.3 > 0:  
 ## :...km.Z5.T1.T2.6 > 1.6: 0 (3)  
 ## : km.Z5.T1.T2.6 <= 1.6:  
 ## : :...km.Z3.4.5 > 2.6: 1 (3)  
 ## : km.Z3.4.5 <= 2.6:  
 ## : :...nr..sessions.4 > 1: 1 (2)  
 ## : nr..sessions.4 <= 1:  
 ## : :...hours.alternative.4 > 0.08: 0 (3)  
 ## : hours.alternative.4 <= 0.08:  
 ## : :...hours.alternative.3 > 1.33: 0 (3/1)  
 ## : hours.alternative.3 <= 1.33:  
 ## : :...nr..sessions <= 0: 0 (5/1)  
 ## : nr..sessions > 0:  
 ## : :...km.Z3.4.6 <= 4.2: 1 (27/3)  
 ## : km.Z3.4.6 > 4.2: 0 (2)  
 ## strength.training.3 <= 0:  
 ## :...nr..sessions.5 > 1:  
 ## :...hours.alternative.3 > 0.87: 1 (2)  
 ## : hours.alternative.3 <= 0.87:  
 ## : :...perceived.recovery <= 0.11: 1 (4/1)  
 ## : perceived.recovery > 0.11: 0 (16)  
 ## nr..sessions.5 <= 1:  
 ## :...nr..sessions.4 > 1:  
 ## :...km.Z3.4.2 > 3.5: 1 (2)  
 ## : km.Z3.4.2 <= 3.5:  
 ## : :...total.km.3 <= 13.2: 0 (16)  
 ## : total.km.3 > 13.2:  
 ## : :...strength.training.2 <= 0: 1 (3)  
 ## : strength.training.2 > 0: 0 (3)  
 ## nr..sessions.4 <= 1:  
 ## :...strength.training.4 > 0:  
 ## :...km.Z3.4.1 > 2.2: 1 (2)  
 ## : km.Z3.4.1 <= 2.2:  
 ## : :...perceived.exertion.5 > 0.83: 1 (3)  
 ## : perceived.exertion.5 <= 0.83:  
 ## : :...total.km.2 <= 6.1: 0 (17)  
 ## : total.km.2 > 6.1:  
 ## : :...nr..sessions.1 <= 0: 1 (2)  
 ## : nr..sessions.1 > 0:  
 ## : :...km.Z5.T1.T2.5 > 0.8: 0 (4)  
 ## : km.Z5.T1.T2.5 <= 0.8: [S5]  
 ## strength.training.4 <= 0:  
 ## :...strength.training.2 > 0:  
 ## :...km.sprinting.4 > 0: 0 (2)  
 ## : km.sprinting.4 <= 0:  
 ## : :...hours.alternative.1 > 0.92: 0 (3)  
 ## : hours.alternative.1 <= 0.92:  
 ## : :...km.Z5.T1.T2.1 <= 4.8: [S6]  
 ## : km.Z5.T1.T2.1 > 4.8:  
 ## : :...km.Z5.T1.T2.1 <= 6.5: 0 (4)  
 ## : km.Z5.T1.T2.1 > 6.5: 1 (2)  
 ## strength.training.2 <= 0:  
 ## :...strength.training.1 > 0:  
 ## :...km.Z3.4 > 3: 0 (2)  
 ## : km.Z3.4 <= 3:  
 ## : :...nr..sessions.3 <= 0: 0 (4/1)  
 ## : nr..sessions.3 > 0:  
 ## : :...nr..sessions.2 <= 0: 1 (4)  
 ## : nr..sessions.2 > 0:  
 ## : :...total.km.5 > 6: 1 (5)  
 ## : total.km.5 <= 6: [S7]  
 ## strength.training.1 <= 0:  
 ## :...nr..sessions.6 > 1:  
 ## :...km.Z3.4.1 <= 4.5: 0 (8)  
 ## : km.Z3.4.1 > 4.5: 1 (2)  
 ## nr..sessions.6 <= 1:  
 ## :...nr..sessions.1 <= 0:  
 ## :...km.sprinting.3 > 0.8: 1 (2)  
 ## : km.sprinting.3 <= 0.8: [S8]  
 ## nr..sessions.1 > 0:  
 ## :...nr..sessions.5 <= 0:  
 ## :...km.Z3.4.3 > 2.5: 0 (3)  
 ## : km.Z3.4.3 <= 2.5: [S9]  
 ## nr..sessions.5 > 0:  
 ## :...total.km.2 > 19: 1 (6)  
 ## total.km.2 <= 19: [S10]  
 ##   
## SubTree [S1]  
 ##   
## nr..sessions.2 <= 1: 1 (3)  
 ## nr..sessions.2 > 1: 0 (2)  
 ##   
## SubTree [S2]  
 ##   
## total.km.2 <= 11.6: 0 (2)  
 ## total.km.2 > 11.6: 1 (2)  
 ##   
## SubTree [S3]  
 ##   
## hours.alternative.6 <= 0.58: 0 (69/8)  
 ## hours.alternative.6 > 0.58:  
 ## :...perceived.exertion.5 <= 0.2: 1 (2)  
 ## perceived.exertion.5 > 0.2: 0 (2)  
 ##   
## SubTree [S4]  
 ##   
## perceived.trainingSuccess.5 > 0.89: 0 (4)  
 ## perceived.trainingSuccess.5 <= 0.89:  
 ## :...strength.training.2 > 0: 1 (11/1)  
 ## strength.training.2 <= 0:  
 ## :...perceived.trainingSuccess.1 > 0.78: 1 (18/1)  
 ## perceived.trainingSuccess.1 <= 0.78:  
 ## :...total\_km\_to\_inj <= 35.6: 1 (7)  
 ## total\_km\_to\_inj > 35.6:  
 ## :...perceived.exertion.2 <= 0.62: 0 (22/5)  
 ## perceived.exertion.2 > 0.62: 1 (11)  
 ##   
## SubTree [S5]  
 ##   
## perceived.trainingSuccess <= 0.57: 0 (2)  
 ## perceived.trainingSuccess > 0.57: 1 (3)  
 ##   
## SubTree [S6]  
 ##   
## perceived.exertion.1 <= 0.63: 1 (17)  
 ## perceived.exertion.1 > 0.63: 0 (3/1)  
 ##   
## SubTree [S7]  
 ##   
## perceived.exertion.4 <= 0.69: 0 (3)  
 ## perceived.exertion.4 > 0.69: 1 (2)  
 ##   
## SubTree [S8]  
 ##   
## perceived.exertion > 0.62: 1 (4)  
 ## perceived.exertion <= 0.62:  
 ## :...perceived.exertion.6 > 0.24: 0 (7)  
 ## perceived.exertion.6 <= 0.24:  
 ## :...km.Z3.4.4 > 0.5: 0 (4)  
 ## km.Z3.4.4 <= 0.5:  
 ## :...perceived.recovery.6 > 0.06: 1 (7/1)  
 ## perceived.recovery.6 <= 0.06:  
 ## :...total.km.4 > 10.8: 1 (5)  
 ## total.km.4 <= 10.8:  
 ## :...perceived.exertion.2 <= 0.18: 0 (10/2)  
 ## perceived.exertion.2 > 0.18: 1 (4/1)  
 ##   
## SubTree [S9]  
 ##   
## km.Z5.T1.T2.4 > 0.5: 0 (7/1)  
 ## km.Z5.T1.T2.4 <= 0.5:  
 ## :...km.Z5.T1.T2.6 > 3: 1 (4)  
 ## km.Z5.T1.T2.6 <= 3:  
 ## :...km.Z5.T1.T2.1 > 0.4: 0 (2)  
 ## km.Z5.T1.T2.1 <= 0.4:  
 ## :...perceived.trainingSuccess.2 > 0.43: 1 (10)  
 ## perceived.trainingSuccess.2 <= 0.43:  
 ## :...km.Z3.4.4 > 1.5: 1 (3)  
 ## km.Z3.4.4 <= 1.5:  
 ## :...km.Z3.4.6 <= 1: 0 (10/1)  
 ## km.Z3.4.6 > 1: 1 (3/1)  
 ##   
## SubTree [S10]  
 ##   
## km.sprinting.6 > 0.2: 0 (8)  
 ## km.sprinting.6 <= 0.2:  
 ## :...km.sprinting.1 > 0.3: 1 (4)  
 ## km.sprinting.1 <= 0.3:  
 ## :...nr..sessions > 1: 0 (6)  
 ## nr..sessions <= 1:  
 ## :...strength.training > 0:  
 ## :...hours.alternative.4 <= 0.5: 1 (6/1)  
 ## : hours.alternative.4 > 0.5: 0 (2)  
 ## strength.training <= 0:  
 ## :...nr..sessions.2 <= 0:  
 ## :...nr..sessions <= 0: 1 (5/1)  
 ## : nr..sessions > 0:  
 ## : :...perceived.exertion.3 <= 0.11: 1 (3)  
 ## : perceived.exertion.3 > 0.11: 0 (7)  
 ## nr..sessions.2 > 0:  
 ## :...hours.alternative.1 > 2.5: 1 (2)  
 ## hours.alternative.1 <= 2.5:  
 ## :...hours.alternative <= 0.83: 0 (40/9)  
 ## hours.alternative > 0.83: 1 (2)  
 ##   
##   
## Evaluation on training data (818 cases):  
 ##   
## Decision Tree   
## ----------------   
## Size Errors   
##   
## 125 57( 7.0%) <<  
 ##   
##   
## (a) (b) <-classified as  
 ## ---- ----  
 ## 389 20 (a): class 0  
 ## 37 372 (b): class 1  
 ##   
##   
## Attribute usage:  
 ##   
## 100.00% nr..sessions.4  
 ## 76.77% sprintkm\_0\_disc  
 ## 69.93% perceived.exertion.6  
 ## 59.29% strength.training.3  
 ## 51.83% nr..sessions.5  
 ## 50.61% strength.training.6  
 ## 49.88% strength.training.2  
 ## 41.93% strength.training.1  
 ## 35.70% km.Z3.4.1  
 ## 33.01% strength.training.4  
 ## 30.68% hours.alternative.1  
 ## 27.75% perceived.exertion.2  
 ## 26.89% nr..sessions.2  
 ## 26.41% km.sprinting.2  
 ## 25.67% km.Z5.T1.T2.6  
 ## 25.43% km.Z3.4  
 ## 25.31% km.Z3.4.6  
 ## 25.18% nr..sessions  
 ## 22.86% nr..sessions.1  
 ## 22.86% hours.alternative.4  
 ## 22.74% nr..sessions.6  
 ## 18.95% km.sprinting.5  
 ## 17.85% km.Z5.T1.T2  
 ## 17.73% perceived.recovery.6  
 ## 15.04% total.km.2  
 ## 12.96% strength.training  
 ## 11.98% perceived.trainingSuccess.5  
 ## 10.76% perceived.exertion  
 ## 10.39% km.sprinting.6  
 ## 10.02% hours.alternative.2  
 ## 9.41% km.sprinting.1  
 ## 9.41% km.Z5.T1.T2.2  
 ## 9.41% km.sprinting.4  
 ## 8.92% hours.alternative.6  
 ## 7.21% hours.alternative.3  
 ## 7.09% perceived.trainingSuccess.1  
 ## 6.60% km.Z5.T1.T2.1  
 ## 5.62% km.Z3.4.4  
 ## 5.50% km.Z3.4.5  
 ## 5.26% km.sprinting.3  
 ## 5.13% hours.alternative  
 ## 5.13% km.Z3.4.3  
 ## 4.89% total\_km\_to\_inj  
 ## 4.77% km.Z5.T1.T2.4  
 ## 4.65% perceived.recovery  
 ## 4.40% km.Z3.4.2  
 ## 4.28% perceived.exertion.5  
 ## 3.67% total.km.3  
 ## 3.18% perceived.trainingSuccess.2  
 ## 3.18% total.km.6  
 ## 2.93% perceived.exertion.1  
 ## 2.32% total.km.4  
 ## 2.20% nr..sessions.3  
 ## 2.08% km.Z5.T1.T2.5  
 ## 1.83% perceived.trainingSuccess.6  
 ## 1.22% perceived.exertion.3  
 ## 1.22% total.km.5  
 ## 1.10% perceived.recovery.1  
 ## 0.86% perceived.recovery.4  
 ## 0.73% total.km  
 ## 0.61% perceived.trainingSuccess  
 ## 0.61% perceived.exertion.4  
 ##   
##   
## Time: 0.0 secs

# Making Predictions for Decision Tree 2

DT\_Pred2 <- predict(DT2, test\_daily)  
 CrossTable(testlabels, DT\_Pred2,   
 prop.chisq = FALSE, prop.c = FALSE, prop.r = FALSE,   
 dnn = c('actual default', 'predicted default'))

##   
##   
## Cell Contents  
 ## |-------------------------|  
 ## | N |  
 ## | N / Table Total |  
 ## |-------------------------|  
 ##   
##   
## Total Observations in Table: 348   
##   
##   
## | predicted default   
## actual default | 0 | 1 | Row Total |   
## ---------------|-----------|-----------|-----------|  
 ## 0 | 102 | 72 | 174 |   
## | 0.293 | 0.207 | |   
## ---------------|-----------|-----------|-----------|  
 ## 1 | 86 | 88 | 174 |   
## | 0.247 | 0.253 | |   
## ---------------|-----------|-----------|-----------|  
 ## Column Total | 188 | 160 | 348 |   
## ---------------|-----------|-----------|-----------|  
 ##   
##

# Improving the Model

*# Keep only total km and sprinting variables*  
dailytrainKMsubset<- subset(train\_daily, select= c(total.km, km.sprinting, total.km.1, km.sprinting.1, total.km.2, km.sprinting.2,total.km.3, km.sprinting.3,total.km.4, km.sprinting.4,total.km.5, km.sprinting.5,total.km.6, km.sprinting.6, injury))  
 dailytestKMsubset <- subset(test\_daily, select= c(total.km, km.sprinting, total.km.1, km.sprinting.1, total.km.2, km.sprinting.2,total.km.3, km.sprinting.3,total.km.4, km.sprinting.4,total.km.5, km.sprinting.5,total.km.6, km.sprinting.6))

# Decision Tree 3

# Training the boosted model with new data subset

DT2\_boosted <-C5.0(dailytrainKMsubset %>% select(-injury), dailytrainKMsubset$injury, trials = 5)  
 DT2

##   
## Call:  
 ## C5.0.default(x = train\_daily %>% select(-injury), y = train\_daily$injury)  
 ##   
## Classification Tree  
 ## Number of samples: 818   
## Number of predictors: 72   
##   
## Tree size: 125   
##   
## Non-standard options: attempt to group attributes

summary(DT2)

##   
## Call:  
 ## C5.0.default(x = train\_daily %>% select(-injury), y = train\_daily$injury)  
 ##   
##   
## C5.0 [Release 2.07 GPL Edition] Thu Mar 23 18:15:06 2023  
 ## -------------------------------  
 ##   
## Class specified by attribute `outcome'  
 ##   
## Read 818 cases (73 attributes) from undefined.data  
 ##   
## Decision tree:  
 ##   
## nr..sessions.4 <= 0:  
 ## :...km.sprinting.2 > 0.5:  
 ## : :...perceived.trainingSuccess.5 <= 0.78: 1 (8)  
 ## : : perceived.trainingSuccess.5 > 0.78: 0 (2)  
 ## : km.sprinting.2 <= 0.5:  
 ## : :...perceived.exertion.2 > 0.67:  
 ## : :...perceived.trainingSuccess.6 <= 0.77: 1 (12)  
 ## : : perceived.trainingSuccess.6 > 0.77: 0 (3)  
 ## : perceived.exertion.2 <= 0.67:  
 ## : :...km.Z3.4.6 > 3:  
 ## : :...total.km.6 <= 11.5: 0 (4/1)  
 ## : : total.km.6 > 11.5: 1 (6)  
 ## : km.Z3.4.6 <= 3:  
 ## : :...km.sprinting.5 > 0: 0 (9)  
 ## : km.sprinting.5 <= 0:  
 ## : :...km.Z5.T1.T2 > 0.5:  
 ## : :...nr..sessions.5 <= 0: 0 (3)  
 ## : : nr..sessions.5 > 0: 1 (6)  
 ## : km.Z5.T1.T2 <= 0.5:  
 ## : :...strength.training.1 > 0: 0 (7)  
 ## : strength.training.1 <= 0:  
 ## : :...km.Z5.T1.T2.6 > 1.2: 0 (11)  
 ## : km.Z5.T1.T2.6 <= 1.2:  
 ## : :...perceived.recovery.6 > 0.32:  
 ## : :...perceived.recovery > 0.42: 0 (4)  
 ## : : perceived.recovery <= 0.42:  
 ## : : :...nr..sessions.5 <= 0: 0 (4/1)  
 ## : : nr..sessions.5 > 0: 1 (10/1)  
 ## : perceived.recovery.6 <= 0.32:  
 ## : :...km.Z3.4.1 > 0.5: 0 (5)  
 ## : km.Z3.4.1 <= 0.5:  
 ## : :...strength.training.2 > 0:  
 ## : :...total.km.3 <= 10: 0 (5/1)  
 ## : : total.km.3 > 10: 1 (3)  
 ## : strength.training.2 <= 0:  
 ## : :...nr..sessions > 1:  
 ## : :...total.km <= 2.5: 0 (3)  
 ## : : total.km > 2.5: 1 (3)  
 ## : nr..sessions <= 1:  
 ## : :...hours.alternative.2 > 0.25: [S1]  
 ## : hours.alternative.2 <= 0.25:  
 ## : :...km.Z5.T1.T2.2 > 0.8: [S2]  
 ## : km.Z5.T1.T2.2 <= 0.8: [S3]  
 ## nr..sessions.4 > 0:  
 ## :...sprintkm\_0\_disc in {0.3-0.39,0.7-0.89,0.9-0.99,1.2-1.39,1.8-1.99,2.0-2.49,  
 ## : 2.5-2.9,3.0-3.49,3.5-3.9,4.0-4.49,4.5-4.49,5.0-5.49,  
 ## : 5.5-5.49,6.0-6.49,6.5-6.99,7.0-7.49,7.5-7.99,8.0-8.49,  
 ## : 9.0-9.49,9.5-9.99,10.0-10.49,10.5-10.99,11.0-11.99,  
 ## : 12.0-12.99,13.0-13.99,14.0-14.99,15.0-15.99,16.0-16.99,  
 ## : 17.0-17.99,18.0-39.9,40.0+}: 1 (16/1)  
 ## sprintkm\_0\_disc in {1.4-1.59,1.6-1.79}: 0 (4/2)  
 ## sprintkm\_0\_disc = 0.4-0.49:  
 ## :...perceived.recovery.4 <= 0.11: 0 (2)  
 ## : perceived.recovery.4 > 0.11: 1 (5)  
 ## sprintkm\_0\_disc = 0.5-0.69:  
 ## :...km.sprinting.4 > 0.8: 0 (2)  
 ## : km.sprinting.4 <= 0.8:  
 ## : :...total.km.6 <= 26: 1 (14/1)  
 ## : total.km.6 > 26: 0 (2)  
 ## sprintkm\_0\_disc = 1.0-1.19:  
 ## :...nr..sessions > 1: 1 (3)  
 ## : nr..sessions <= 1:  
 ## : :...km.Z5.T1.T2.5 <= 0.6: 0 (6)  
 ## : km.Z5.T1.T2.5 > 0.6: 1 (2)  
 ## sprintkm\_0\_disc = 0.0-0.29:  
 ## :...perceived.exertion.6 > 0.48:  
 ## :...km.Z3.4 > 0: 1 (10)  
 ## : km.Z3.4 <= 0:  
 ## : :...km.Z3.4.1 > 6:  
 ## : :...perceived.recovery.1 <= 0.36: 1 (3/1)  
 ## : : perceived.recovery.1 > 0.36: 0 (6)  
 ## : km.Z3.4.1 <= 6:  
 ## : :...hours.alternative.4 > 1.17: 1 (11)  
 ## : hours.alternative.4 <= 1.17:  
 ## : :...hours.alternative.1 > 1.17: 1 (7)  
 ## : hours.alternative.1 <= 1.17:  
 ## : :...strength.training.3 > 0:  
 ## : :...perceived.trainingSuccess.5 <= 0.37: 1 (8/1)  
 ## : : perceived.trainingSuccess.5 > 0.37: 0 (7)  
 ## : strength.training.3 <= 0:  
 ## : :...nr..sessions.2 <= 0: 1 (21/2)  
 ## : nr..sessions.2 > 0:  
 ## : :...nr..sessions.5 <= 0:  
 ## : :...km.Z3.4.2 <= 2: 0 (9/1)  
 ## : : km.Z3.4.2 > 2: 1 (3)  
 ## : nr..sessions.5 > 0: [S4]  
 ## perceived.exertion.6 <= 0.48:  
 ## :...strength.training.6 > 0:  
 ## :...hours.alternative.1 > 1.33: 0 (3)  
 ## : hours.alternative.1 <= 1.33:  
 ## : :...perceived.exertion > 0.54: 1 (8)  
 ## : perceived.exertion <= 0.54:  
 ## : :...strength.training > 0: 0 (3)  
 ## : strength.training <= 0:  
 ## : :...nr..sessions.2 > 1: 0 (6/1)  
 ## : nr..sessions.2 <= 1:  
 ## : :...km.Z3.4 > 5.8: 0 (2)  
 ## : km.Z3.4 <= 5.8:  
 ## : :...km.sprinting.4 > 1.7: 0 (2)  
 ## : km.sprinting.4 <= 1.7:  
 ## : :...km.sprinting.2 <= 0: 1 (22/2)  
 ## : km.sprinting.2 > 0:  
 ## : :...perceived.exertion.1 <= 0.1: 1 (2)  
 ## : perceived.exertion.1 > 0.1: 0 (2)  
 ## strength.training.6 <= 0:  
 ## :...strength.training.3 > 0:  
 ## :...km.Z5.T1.T2.6 > 1.6: 0 (3)  
 ## : km.Z5.T1.T2.6 <= 1.6:  
 ## : :...km.Z3.4.5 > 2.6: 1 (3)  
 ## : km.Z3.4.5 <= 2.6:  
 ## : :...nr..sessions.4 > 1: 1 (2)  
 ## : nr..sessions.4 <= 1:  
 ## : :...hours.alternative.4 > 0.08: 0 (3)  
 ## : hours.alternative.4 <= 0.08:  
 ## : :...hours.alternative.3 > 1.33: 0 (3/1)  
 ## : hours.alternative.3 <= 1.33:  
 ## : :...nr..sessions <= 0: 0 (5/1)  
 ## : nr..sessions > 0:  
 ## : :...km.Z3.4.6 <= 4.2: 1 (27/3)  
 ## : km.Z3.4.6 > 4.2: 0 (2)  
 ## strength.training.3 <= 0:  
 ## :...nr..sessions.5 > 1:  
 ## :...hours.alternative.3 > 0.87: 1 (2)  
 ## : hours.alternative.3 <= 0.87:  
 ## : :...perceived.recovery <= 0.11: 1 (4/1)  
 ## : perceived.recovery > 0.11: 0 (16)  
 ## nr..sessions.5 <= 1:  
 ## :...nr..sessions.4 > 1:  
 ## :...km.Z3.4.2 > 3.5: 1 (2)  
 ## : km.Z3.4.2 <= 3.5:  
 ## : :...total.km.3 <= 13.2: 0 (16)  
 ## : total.km.3 > 13.2:  
 ## : :...strength.training.2 <= 0: 1 (3)  
 ## : strength.training.2 > 0: 0 (3)  
 ## nr..sessions.4 <= 1:  
 ## :...strength.training.4 > 0:  
 ## :...km.Z3.4.1 > 2.2: 1 (2)  
 ## : km.Z3.4.1 <= 2.2:  
 ## : :...perceived.exertion.5 > 0.83: 1 (3)  
 ## : perceived.exertion.5 <= 0.83:  
 ## : :...total.km.2 <= 6.1: 0 (17)  
 ## : total.km.2 > 6.1:  
 ## : :...nr..sessions.1 <= 0: 1 (2)  
 ## : nr..sessions.1 > 0:  
 ## : :...km.Z5.T1.T2.5 > 0.8: 0 (4)  
 ## : km.Z5.T1.T2.5 <= 0.8: [S5]  
 ## strength.training.4 <= 0:  
 ## :...strength.training.2 > 0:  
 ## :...km.sprinting.4 > 0: 0 (2)  
 ## : km.sprinting.4 <= 0:  
 ## : :...hours.alternative.1 > 0.92: 0 (3)  
 ## : hours.alternative.1 <= 0.92:  
 ## : :...km.Z5.T1.T2.1 <= 4.8: [S6]  
 ## : km.Z5.T1.T2.1 > 4.8:  
 ## : :...km.Z5.T1.T2.1 <= 6.5: 0 (4)  
 ## : km.Z5.T1.T2.1 > 6.5: 1 (2)  
 ## strength.training.2 <= 0:  
 ## :...strength.training.1 > 0:  
 ## :...km.Z3.4 > 3: 0 (2)  
 ## : km.Z3.4 <= 3:  
 ## : :...nr..sessions.3 <= 0: 0 (4/1)  
 ## : nr..sessions.3 > 0:  
 ## : :...nr..sessions.2 <= 0: 1 (4)  
 ## : nr..sessions.2 > 0:  
 ## : :...total.km.5 > 6: 1 (5)  
 ## : total.km.5 <= 6: [S7]  
 ## strength.training.1 <= 0:  
 ## :...nr..sessions.6 > 1:  
 ## :...km.Z3.4.1 <= 4.5: 0 (8)  
 ## : km.Z3.4.1 > 4.5: 1 (2)  
 ## nr..sessions.6 <= 1:  
 ## :...nr..sessions.1 <= 0:  
 ## :...km.sprinting.3 > 0.8: 1 (2)  
 ## : km.sprinting.3 <= 0.8: [S8]  
 ## nr..sessions.1 > 0:  
 ## :...nr..sessions.5 <= 0:  
 ## :...km.Z3.4.3 > 2.5: 0 (3)  
 ## : km.Z3.4.3 <= 2.5: [S9]  
 ## nr..sessions.5 > 0:  
 ## :...total.km.2 > 19: 1 (6)  
 ## total.km.2 <= 19: [S10]  
 ##   
## SubTree [S1]  
 ##   
## nr..sessions.2 <= 1: 1 (3)  
 ## nr..sessions.2 > 1: 0 (2)  
 ##   
## SubTree [S2]  
 ##   
## total.km.2 <= 11.6: 0 (2)  
 ## total.km.2 > 11.6: 1 (2)  
 ##   
## SubTree [S3]  
 ##   
## hours.alternative.6 <= 0.58: 0 (69/8)  
 ## hours.alternative.6 > 0.58:  
 ## :...perceived.exertion.5 <= 0.2: 1 (2)  
 ## perceived.exertion.5 > 0.2: 0 (2)  
 ##   
## SubTree [S4]  
 ##   
## perceived.trainingSuccess.5 > 0.89: 0 (4)  
 ## perceived.trainingSuccess.5 <= 0.89:  
 ## :...strength.training.2 > 0: 1 (11/1)  
 ## strength.training.2 <= 0:  
 ## :...perceived.trainingSuccess.1 > 0.78: 1 (18/1)  
 ## perceived.trainingSuccess.1 <= 0.78:  
 ## :...total\_km\_to\_inj <= 35.6: 1 (7)  
 ## total\_km\_to\_inj > 35.6:  
 ## :...perceived.exertion.2 <= 0.62: 0 (22/5)  
 ## perceived.exertion.2 > 0.62: 1 (11)  
 ##   
## SubTree [S5]  
 ##   
## perceived.trainingSuccess <= 0.57: 0 (2)  
 ## perceived.trainingSuccess > 0.57: 1 (3)  
 ##   
## SubTree [S6]  
 ##   
## perceived.exertion.1 <= 0.63: 1 (17)  
 ## perceived.exertion.1 > 0.63: 0 (3/1)  
 ##   
## SubTree [S7]  
 ##   
## perceived.exertion.4 <= 0.69: 0 (3)  
 ## perceived.exertion.4 > 0.69: 1 (2)  
 ##   
## SubTree [S8]  
 ##   
## perceived.exertion > 0.62: 1 (4)  
 ## perceived.exertion <= 0.62:  
 ## :...perceived.exertion.6 > 0.24: 0 (7)  
 ## perceived.exertion.6 <= 0.24:  
 ## :...km.Z3.4.4 > 0.5: 0 (4)  
 ## km.Z3.4.4 <= 0.5:  
 ## :...perceived.recovery.6 > 0.06: 1 (7/1)  
 ## perceived.recovery.6 <= 0.06:  
 ## :...total.km.4 > 10.8: 1 (5)  
 ## total.km.4 <= 10.8:  
 ## :...perceived.exertion.2 <= 0.18: 0 (10/2)  
 ## perceived.exertion.2 > 0.18: 1 (4/1)  
 ##   
## SubTree [S9]  
 ##   
## km.Z5.T1.T2.4 > 0.5: 0 (7/1)  
 ## km.Z5.T1.T2.4 <= 0.5:  
 ## :...km.Z5.T1.T2.6 > 3: 1 (4)  
 ## km.Z5.T1.T2.6 <= 3:  
 ## :...km.Z5.T1.T2.1 > 0.4: 0 (2)  
 ## km.Z5.T1.T2.1 <= 0.4:  
 ## :...perceived.trainingSuccess.2 > 0.43: 1 (10)  
 ## perceived.trainingSuccess.2 <= 0.43:  
 ## :...km.Z3.4.4 > 1.5: 1 (3)  
 ## km.Z3.4.4 <= 1.5:  
 ## :...km.Z3.4.6 <= 1: 0 (10/1)  
 ## km.Z3.4.6 > 1: 1 (3/1)  
 ##   
## SubTree [S10]  
 ##   
## km.sprinting.6 > 0.2: 0 (8)  
 ## km.sprinting.6 <= 0.2:  
 ## :...km.sprinting.1 > 0.3: 1 (4)  
 ## km.sprinting.1 <= 0.3:  
 ## :...nr..sessions > 1: 0 (6)  
 ## nr..sessions <= 1:  
 ## :...strength.training > 0:  
 ## :...hours.alternative.4 <= 0.5: 1 (6/1)  
 ## : hours.alternative.4 > 0.5: 0 (2)  
 ## strength.training <= 0:  
 ## :...nr..sessions.2 <= 0:  
 ## :...nr..sessions <= 0: 1 (5/1)  
 ## : nr..sessions > 0:  
 ## : :...perceived.exertion.3 <= 0.11: 1 (3)  
 ## : perceived.exertion.3 > 0.11: 0 (7)  
 ## nr..sessions.2 > 0:  
 ## :...hours.alternative.1 > 2.5: 1 (2)  
 ## hours.alternative.1 <= 2.5:  
 ## :...hours.alternative <= 0.83: 0 (40/9)  
 ## hours.alternative > 0.83: 1 (2)  
 ##   
##   
## Evaluation on training data (818 cases):  
 ##   
## Decision Tree   
## ----------------   
## Size Errors   
##   
## 125 57( 7.0%) <<  
 ##   
##   
## (a) (b) <-classified as  
 ## ---- ----  
 ## 389 20 (a): class 0  
 ## 37 372 (b): class 1  
 ##   
##   
## Attribute usage:  
 ##   
## 100.00% nr..sessions.4  
 ## 76.77% sprintkm\_0\_disc  
 ## 69.93% perceived.exertion.6  
 ## 59.29% strength.training.3  
 ## 51.83% nr..sessions.5  
 ## 50.61% strength.training.6  
 ## 49.88% strength.training.2  
 ## 41.93% strength.training.1  
 ## 35.70% km.Z3.4.1  
 ## 33.01% strength.training.4  
 ## 30.68% hours.alternative.1  
 ## 27.75% perceived.exertion.2  
 ## 26.89% nr..sessions.2  
 ## 26.41% km.sprinting.2  
 ## 25.67% km.Z5.T1.T2.6  
 ## 25.43% km.Z3.4  
 ## 25.31% km.Z3.4.6  
 ## 25.18% nr..sessions  
 ## 22.86% nr..sessions.1  
 ## 22.86% hours.alternative.4  
 ## 22.74% nr..sessions.6  
 ## 18.95% km.sprinting.5  
 ## 17.85% km.Z5.T1.T2  
 ## 17.73% perceived.recovery.6  
 ## 15.04% total.km.2  
 ## 12.96% strength.training  
 ## 11.98% perceived.trainingSuccess.5  
 ## 10.76% perceived.exertion  
 ## 10.39% km.sprinting.6  
 ## 10.02% hours.alternative.2  
 ## 9.41% km.sprinting.1  
 ## 9.41% km.Z5.T1.T2.2  
 ## 9.41% km.sprinting.4  
 ## 8.92% hours.alternative.6  
 ## 7.21% hours.alternative.3  
 ## 7.09% perceived.trainingSuccess.1  
 ## 6.60% km.Z5.T1.T2.1  
 ## 5.62% km.Z3.4.4  
 ## 5.50% km.Z3.4.5  
 ## 5.26% km.sprinting.3  
 ## 5.13% hours.alternative  
 ## 5.13% km.Z3.4.3  
 ## 4.89% total\_km\_to\_inj  
 ## 4.77% km.Z5.T1.T2.4  
 ## 4.65% perceived.recovery  
 ## 4.40% km.Z3.4.2  
 ## 4.28% perceived.exertion.5  
 ## 3.67% total.km.3  
 ## 3.18% perceived.trainingSuccess.2  
 ## 3.18% total.km.6  
 ## 2.93% perceived.exertion.1  
 ## 2.32% total.km.4  
 ## 2.20% nr..sessions.3  
 ## 2.08% km.Z5.T1.T2.5  
 ## 1.83% perceived.trainingSuccess.6  
 ## 1.22% perceived.exertion.3  
 ## 1.22% total.km.5  
 ## 1.10% perceived.recovery.1  
 ## 0.86% perceived.recovery.4  
 ## 0.73% total.km  
 ## 0.61% perceived.trainingSuccess  
 ## 0.61% perceived.exertion.4  
 ##   
##   
## Time: 0.0 secs

# Make “boosted” predictions (for Decision Tree 3)

DT\_Pred2\_boosted <- predict(DT2\_boosted, dailytestKMsubset)  
 CrossTable(testlabels, DT\_Pred2\_boosted,   
 prop.chisq = FALSE, prop.c = FALSE, prop.r = FALSE,   
 dnn = c('actual default', 'predicted default'))

##   
##   
## Cell Contents  
 ## |-------------------------|  
 ## | N |  
 ## | N / Table Total |  
 ## |-------------------------|  
 ##   
##   
## Total Observations in Table: 348   
##   
##   
## | predicted default   
## actual default | 0 | 1 | Row Total |   
## ---------------|-----------|-----------|-----------|  
 ## 0 | 112 | 62 | 174 |   
## | 0.322 | 0.178 | |   
## ---------------|-----------|-----------|-----------|  
 ## 1 | 80 | 94 | 174 |   
## | 0.230 | 0.270 | |   
## ---------------|-----------|-----------|-----------|  
 ## Column Total | 192 | 156 | 348 |   
## ---------------|-----------|-----------|-----------|  
 ##   
##

# Random Forest

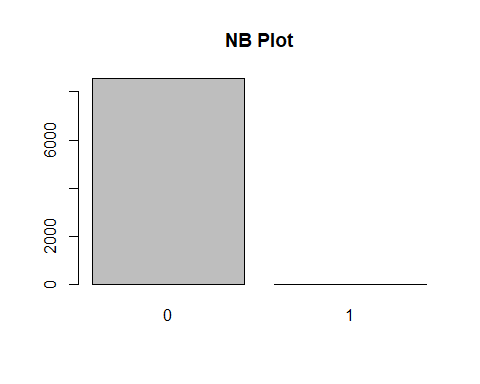
Resulted in 5 or fewer unique values.

# Naive Bayes

The Naive Bayes model took some extra preparations and research before running. The cost was adjusted to 100 because of the results found in the association rule mining section.

#Using Sigmoid kernel High accuracy but no injuries predicted.

NB <- naiveBayes(injury~., data = dailytrain, kernel="sigmoid", cost=100, scale=TRUE)  
  
NB\_pred <- predict(NB, dailytest, type = "class")  
  
plot(NB\_pred, main = "NB Plot")

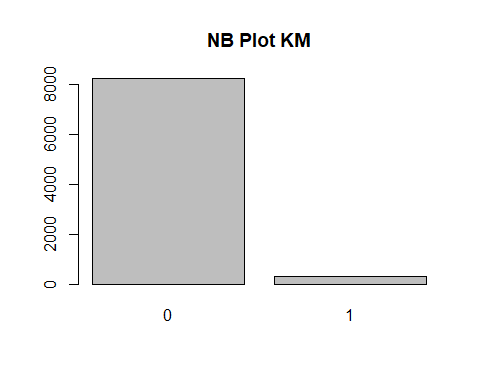


##Confusion Matrix  
  
confusionMatrix(dailytesttarget, NB\_pred)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 7970 0  
## 1 583 0  
##   
## Accuracy : 0.9318   
## 95% CI : (0.9263, 0.9371)  
## No Information Rate : 1   
## P-Value [Acc > NIR] : 1   
##   
## Kappa : 0   
##   
## Mcnemar's Test P-Value : <2e-16   
##   
## Sensitivity : 0.9318   
## Specificity : NA   
## Pos Pred Value : NA   
## Neg Pred Value : NA   
## Prevalence : 1.0000   
## Detection Rate : 0.9318   
## Detection Prevalence : 0.9318   
## Balanced Accuracy : NA   
##   
## 'Positive' Class : 0   
##

#Using Sigmoid kernel for KM dataset Similar accuracy and but still poor performance with only 6 true positives.

NBkm <- naiveBayes(injury~., data = dailytrainKM, kernel="sigmoid", cost=100, scale=TRUE)  
  
NB\_predkm <- predict(NBkm, dailytestKM, type = "class")  
  
plot(NB\_predkm, main = "NB Plot KM")



##Confusion Matrix  
confusionMatrix(dailytesttargetKM, NB\_predkm)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 8142 296  
## 1 109 6  
##   
## Accuracy : 0.9526   
## 95% CI : (0.9479, 0.9571)  
## No Information Rate : 0.9647   
## P-Value [Acc > NIR] : 1   
##   
## Kappa : 0.0095   
##   
## Mcnemar's Test P-Value : <2e-16   
##   
## Sensitivity : 0.98679   
## Specificity : 0.01987   
## Pos Pred Value : 0.96492   
## Neg Pred Value : 0.05217   
## Prevalence : 0.96469   
## Detection Rate : 0.95195   
## Detection Prevalence : 0.98655   
## Balanced Accuracy : 0.50333   
##   
## 'Positive' Class : 0   
##

# kNN

Attempts to run K nearest neighbor on the daily and daily kilometer datasets were unsuccessful.

# SVM

Preparing the daily dataset for SVM. Because the model takes longer to run, pulled out a balanced sample of injured and non-injured runners to expedite results. The first sample size is 818.

#Make a copy of the dataset to use for SVM  
dailysvm <- dailyKM  
dailysvm

#set seed, seperate injured and non-injured, sample and create a new, smaller, dataset.  
set.seed(123)  
noinjuries <- dailysvm %>% filter(injury == "0")  
injuries <- dailysvm %>% filter(injury == "1")  
sample <- sample(nrow(noinjuries), 583, replace = FALSE)  
  
samplenoinjuries <- noinjuries[sample,]  
  
#new dataframe to use for training   
dailysamplesvm <- rbind(samplenoinjuries, injuries)  
  
#scale the data  
  
dailysamplesvm %>%  
 mutate\_at(c(1:28), funs(c(scale(.))))

## 583 -0.35393751 -0.197389020 1

# Create the test and train dataset from the sample set  
trainlist <- createDataPartition(y=dailysamplesvm$injury, p=.7, list=FALSE)  
train\_daily <- dailysamplesvm[trainlist,]  
test\_daily <- dailysamplesvm[-trainlist,]  
  
testlabels <- test\_daily$injury  
test\_daily <- test\_daily %>% select(-injury)

Run with all types of kernels

SVM <- svm(injury~., data = train\_daily, kernel="polynomial", cost=100, scale=TRUE)  
print(SVM)

##   
## Call:  
## svm(formula = injury ~ ., data = train\_daily, kernel = "polynomial",   
## cost = 100, scale = TRUE)  
##   
##   
## Parameters:  
## SVM-Type: C-classification   
## SVM-Kernel: polynomial   
## cost: 100   
## degree: 3   
## coef.0: 0   
##   
## Number of Support Vectors: 574

#Run prediction with the polynomial kernel  
SVM\_Pred <- predict(SVM, test\_daily, type="class")  
#Confusion matrix  
confusionMatrix(testlabels, SVM\_Pred)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 107 67  
## 1 81 93  
##   
## Accuracy : 0.5747   
## 95% CI : (0.5209, 0.6273)  
## No Information Rate : 0.5402   
## P-Value [Acc > NIR] : 0.1079   
##   
## Kappa : 0.1494   
##   
## Mcnemar's Test P-Value : 0.2853   
##   
## Sensitivity : 0.5691   
## Specificity : 0.5813   
## Pos Pred Value : 0.6149   
## Neg Pred Value : 0.5345   
## Prevalence : 0.5402   
## Detection Rate : 0.3075   
## Detection Prevalence : 0.5000   
## Balanced Accuracy : 0.5752   
##   
## 'Positive' Class : 0   
##

SVM2 <- svm(injury~., data = train\_daily, kernel="sigmoid", cost=100, scale=TRUE)  
print(SVM2)

##   
## Call:  
## svm(formula = injury ~ ., data = train\_daily, kernel = "sigmoid",   
## cost = 100, scale = TRUE)  
##   
##   
## Parameters:  
## SVM-Type: C-classification   
## SVM-Kernel: sigmoid   
## cost: 100   
## coef.0: 0   
##   
## Number of Support Vectors: 424

#Run prediction with the sigmoid kernel  
SVM\_Pred2 <- predict(SVM2, test\_daily, type="class")  
#Confusion matrix  
confusionMatrix(testlabels, SVM\_Pred2)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 102 72  
## 1 78 96  
##   
## Accuracy : 0.569   
## 95% CI : (0.5151, 0.6217)  
## No Information Rate : 0.5172   
## P-Value [Acc > NIR] : 0.03005   
##   
## Kappa : 0.1379   
##   
## Mcnemar's Test P-Value : 0.68309   
##   
## Sensitivity : 0.5667   
## Specificity : 0.5714   
## Pos Pred Value : 0.5862   
## Neg Pred Value : 0.5517   
## Prevalence : 0.5172   
## Detection Rate : 0.2931   
## Detection Prevalence : 0.5000   
## Balanced Accuracy : 0.5690   
##   
## 'Positive' Class : 0   
##

svm3 <-svm(injury~., data = train\_daily, kernel = "linear", cost=100, scale=TRUE)  
print(svm3)

##   
## Call:  
## svm(formula = injury ~ ., data = train\_daily, kernel = "linear",   
## cost = 100, scale = TRUE)  
##   
##   
## Parameters:  
## SVM-Type: C-classification   
## SVM-Kernel: linear   
## cost: 100   
##   
## Number of Support Vectors: 692

#Run prediction with the linear kernel  
SVM\_Pred3 <- predict(svm3, test\_daily, type="class")  
#Confusion matrix  
confusionMatrix(testlabels, SVM\_Pred3)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 119 55  
## 1 84 90  
##   
## Accuracy : 0.6006   
## 95% CI : (0.547, 0.6524)  
## No Information Rate : 0.5833   
## P-Value [Acc > NIR] : 0.27559   
##   
## Kappa : 0.2011   
##   
## Mcnemar's Test P-Value : 0.01755   
##   
## Sensitivity : 0.5862   
## Specificity : 0.6207   
## Pos Pred Value : 0.6839   
## Neg Pred Value : 0.5172   
## Prevalence : 0.5833   
## Detection Rate : 0.3420   
## Detection Prevalence : 0.5000   
## Balanced Accuracy : 0.6034   
##   
## 'Positive' Class : 0   
##

Run with different costs

SVM <- svm(injury~., data = train\_daily, kernel="polynomial", cost=500, scale=TRUE)  
print(SVM)

##   
## Call:  
## svm(formula = injury ~ ., data = train\_daily, kernel = "polynomial",   
## cost = 500, scale = TRUE)  
##   
##   
## Parameters:  
## SVM-Type: C-classification   
## SVM-Kernel: polynomial   
## cost: 500   
## degree: 3   
## coef.0: 0   
##   
## Number of Support Vectors: 541

#Run prediction with the polynomial kernel  
SVM\_Pred <- predict(SVM, test\_daily, type="class")  
#Confusion matrix  
confusionMatrix(testlabels, SVM\_Pred)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 104 70  
## 1 76 98  
##   
## Accuracy : 0.5805   
## 95% CI : (0.5267, 0.6329)  
## No Information Rate : 0.5172   
## P-Value [Acc > NIR] : 0.0104   
##   
## Kappa : 0.1609   
##   
## Mcnemar's Test P-Value : 0.6790   
##   
## Sensitivity : 0.5778   
## Specificity : 0.5833   
## Pos Pred Value : 0.5977   
## Neg Pred Value : 0.5632   
## Prevalence : 0.5172   
## Detection Rate : 0.2989   
## Detection Prevalence : 0.5000   
## Balanced Accuracy : 0.5806   
##   
## 'Positive' Class : 0   
##

SVM2 <- svm(injury~., data = train\_daily, kernel="sigmoid", cost=500, scale=TRUE)  
print(SVM2)

##   
## Call:  
## svm(formula = injury ~ ., data = train\_daily, kernel = "sigmoid",   
## cost = 500, scale = TRUE)  
##   
##   
## Parameters:  
## SVM-Type: C-classification   
## SVM-Kernel: sigmoid   
## cost: 500   
## coef.0: 0   
##   
## Number of Support Vectors: 420

#Run prediction with the sigmoid kernel  
SVM\_Pred2 <- predict(SVM2, test\_daily, type="class")  
#Confusion matrix  
confusionMatrix(testlabels, SVM\_Pred2)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 99 75  
## 1 77 97  
##   
## Accuracy : 0.5632   
## 95% CI : (0.5093, 0.616)  
## No Information Rate : 0.5057   
## P-Value [Acc > NIR] : 0.01817   
##   
## Kappa : 0.1264   
##   
## Mcnemar's Test P-Value : 0.93535   
##   
## Sensitivity : 0.5625   
## Specificity : 0.5640   
## Pos Pred Value : 0.5690   
## Neg Pred Value : 0.5575   
## Prevalence : 0.5057   
## Detection Rate : 0.2845   
## Detection Prevalence : 0.5000   
## Balanced Accuracy : 0.5632   
##   
## 'Positive' Class : 0   
##

svm3 <-svm(injury~., data = train\_daily, kernel = "linear", cost=500, scale=TRUE)  
print(svm3)

##   
## Call:  
## svm(formula = injury ~ ., data = train\_daily, kernel = "linear",   
## cost = 500, scale = TRUE)  
##   
##   
## Parameters:  
## SVM-Type: C-classification   
## SVM-Kernel: linear   
## cost: 500   
##   
## Number of Support Vectors: 692

#Run prediction with the linear kernel  
SVM\_Pred3 <- predict(svm3, test\_daily, type="class")  
#Confusion matrix  
confusionMatrix(testlabels, SVM\_Pred3)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 118 56  
## 1 83 91  
##   
## Accuracy : 0.6006   
## 95% CI : (0.547, 0.6524)  
## No Information Rate : 0.5776   
## P-Value [Acc > NIR] : 0.20813   
##   
## Kappa : 0.2011   
##   
## Mcnemar's Test P-Value : 0.02743   
##   
## Sensitivity : 0.5871   
## Specificity : 0.6190   
## Pos Pred Value : 0.6782   
## Neg Pred Value : 0.5230   
## Prevalence : 0.5776   
## Detection Rate : 0.3391   
## Detection Prevalence : 0.5000   
## Balanced Accuracy : 0.6031   
##   
## 'Positive' Class : 0   
##

#SVM Increasing the sample size

Increase the sample size in attempt to get a higher accuracy. Doubled the count to 2000.

dailysvm2 <- dailyKM  
dailysvm2

## 3146 4.7 0.0 0.0 0.0 0.0  
0.0

set.seed(123)  
noinjuries2 <- dailysvm2 %>% filter(injury == "0")  
injuries2 <- dailysvm2 %>% filter(injury == "1")  
sample2 <- sample(nrow(noinjuries), 2000, replace = FALSE)  
  
samplenoinjuries2 <- noinjuries2[sample2,]  
  
#new dataframe to use for training   
dailysamplesvm2 <- rbind(samplenoinjuries2, injuries2)  
  
#scale the data  
  
dailysamplesvm2 %>%  
 mutate\_at(c(1:28), funs(c(scale(.))))

# Create the test and train dataset  
trainlist2 <- createDataPartition(y=dailysamplesvm2$injury, p=.8, list=FALSE)  
train\_daily2 <- dailysamplesvm2[trainlist2,]  
test\_daily2 <- dailysamplesvm2[-trainlist2,]  
  
testlabels2 <- test\_daily2$injury  
test\_daily2 <- test\_daily2 %>% select(-injury)

#SVM for increased sample

SVMls <- svm(injury~., data = train\_daily2, kernel="linear", cost=100, scale=TRUE)  
print(SVMls)

##   
## Call:  
## svm(formula = injury ~ ., data = train\_daily2, kernel = "linear",   
## cost = 100, scale = TRUE)  
##   
##   
## Parameters:  
## SVM-Type: C-classification   
## SVM-Kernel: linear   
## cost: 100   
##   
## Number of Support Vectors: 1426

#Predication for linear kernel  
SVM\_Predls <- predict(SVMls, test\_daily2, type="class")  
#Confusion matrix  
confusionMatrix(testlabels2, SVM\_Predls)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 400 0  
## 1 116 0  
##   
## Accuracy : 0.7752   
## 95% CI : (0.7367, 0.8105)  
## No Information Rate : 1   
## P-Value [Acc > NIR] : 1   
##   
## Kappa : 0   
##   
## Mcnemar's Test P-Value : <2e-16   
##   
## Sensitivity : 0.7752   
## Specificity : NA   
## Pos Pred Value : NA   
## Neg Pred Value : NA   
## Prevalence : 1.0000   
## Detection Rate : 0.7752   
## Detection Prevalence : 0.7752   
## Balanced Accuracy : NA   
##   
## 'Positive' Class : 0   
##

SVMls2 <- svm(injury~., data = train\_daily2, kernel="sigmoid", cost=100, scale=TRUE)  
print(SVMls2)

##   
## Call:  
## svm(formula = injury ~ ., data = train\_daily2, kernel = "sigmoid",   
## cost = 100, scale = TRUE)  
##   
##   
## Parameters:  
## SVM-Type: C-classification   
## SVM-Kernel: sigmoid   
## cost: 100   
## coef.0: 0   
##   
## Number of Support Vectors: 791

#Predication for sigmoid kernel  
SVM\_Predls2 <- predict(SVMls2, test\_daily2, type="class")  
#Confusion matrix  
confusionMatrix(testlabels2, SVM\_Predls2)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 330 70  
## 1 93 23  
##   
## Accuracy : 0.6841   
## 95% CI : (0.6421, 0.724)  
## No Information Rate : 0.8198   
## P-Value [Acc > NIR] : 1.00000   
##   
## Kappa : 0.025   
##   
## Mcnemar's Test P-Value : 0.08486   
##   
## Sensitivity : 0.7801   
## Specificity : 0.2473   
## Pos Pred Value : 0.8250   
## Neg Pred Value : 0.1983   
## Prevalence : 0.8198   
## Detection Rate : 0.6395   
## Detection Prevalence : 0.7752   
## Balanced Accuracy : 0.5137   
##   
## 'Positive' Class : 0   
##

SVMls3 <- svm(injury~., data = train\_daily2, kernel="polynomial", cost=100, scale=TRUE)  
print(SVMls3)

##   
## Call:  
## svm(formula = injury ~ ., data = train\_daily2, kernel = "polynomial",   
## cost = 100, scale = TRUE)  
##   
##   
## Parameters:  
## SVM-Type: C-classification   
## SVM-Kernel: polynomial   
## cost: 100   
## degree: 3   
## coef.0: 0   
##   
## Number of Support Vectors: 1070

#Predication for polynomial kernel  
SVM\_Predls3 <- predict(SVMls3, test\_daily2, type="class")  
#Confusion matrix  
confusionMatrix(testlabels2, SVM\_Predls3)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 345 55  
## 1 87 29  
##   
## Accuracy : 0.7248   
## 95% CI : (0.6841, 0.7629)  
## No Information Rate : 0.8372   
## P-Value [Acc > NIR] : 1.000000   
##   
## Kappa : 0.1247   
##   
## Mcnemar's Test P-Value : 0.009283   
##   
## Sensitivity : 0.7986   
## Specificity : 0.3452   
## Pos Pred Value : 0.8625   
## Neg Pred Value : 0.2500   
## Prevalence : 0.8372   
## Detection Rate : 0.6686   
## Detection Prevalence : 0.7752   
## Balanced Accuracy : 0.5719   
##   
## 'Positive' Class : 0   
##

Results

# ARM

Showed how imbalanced the dataset was by showing rules with equivalency of 0. Interesting Rules are not found when running the Apriori, even after utilizing discretized sprint km column. All rules point to the 0s in the data set. Even with lowering the support and confidence we found no rules indicating injury.

# Decision Trees

**Decision Tree 1**: **56% accuracy with 92 false positives and 58 false negatives**

* If perceived training success for the athlete was no and perceived exertion 7 days before injury was high (tired in the beginning of the wees, then it may possibly lead to injury
* If Perceived exertion is low, the athlete may possible run more km 3 days before injury (km.2) and that may lead to injury
* If perceived training success is higher in the middle of the week, the athlete may sprint 3 days before injury, and it may lead to possible injury.

Drawing out a complexity parameter, it shows that the more complex the dataset becomes, the less accurate. Therefore, tuning is attempted to improve accuracy

**Decision Tree 2: using C.5: 7% error rate with 20 false negatives and 37 false positives**

When creating a prediction model, the model predicted 102/174 non injuries and 88/174 injuries correctly.

For attributes usage, there seems to be a pattern with perceived exertion.6 (7 days before possible injury) and sprinting km

Attribute usage:   
   
 100.00% nr..sessions.4   
 76.77% sprintkm\_0\_disc   
 69.93% perceived.exertion.6   
 59.29% strength.training.3   
 51.83% nr..sessions.5   
 50.61% strength.training.6   
   
**Boosted Tree (Decision Tree 3): also 7% accuracy with 20 false negatives and 37 false positives like previous. For this model, only total km ran and km sprinting variables were used for all 7 days before “event,” with 5 trials in hopes of more accuracy.**

When creating predictions, the model predicted 112/174 non-injuries and 94/174 injuries correctly. That is slightly more accurate than the previous predictions from DT2.

For attributes usage, although subset with only km was used, perceived exertion.6 still appeared and the same attributes usage was seen.

Attribute usage:

100.00% nr..sessions.4   
 76.77% sprintkm\_0\_disc   
 69.93% perceived.exertion.6   
 59.29% strength.training.3   
 51.83% nr..sessions.5   
 50.61% strength.training.6

# Random Forest

Attempts to run Random Forest resulted in a message that there were 5 of fewer unique values. This message was not surprising given the high-level tuning needed to create a decision tree. The model was not able to make any notable classification.

# Naive Bayes

*Daily:* The model for the daily dataset resulted in 93% accuracy but did not predict any injuries. Attempting to add and edit parameters did not yield better results.

*Daily Kilometer Only:* The model resulted in 95% accuracy and was tuned to predict 6 true positives, but 296 false positives.

# kNN

Troubleshooting was conducted for the error message “too many ties in knn” with no answer found.

# SVM

*Daily Sample Set Sample size of 818.*

*Polynomial with cost=100:* The accuracy of this model was 57.4%, with a balanced, but still high rate of false positives (67) and false negatives (81). There is room for improvement.

*Sigmoid with cost=100:* The accuracy of this model was 56.9%, with a also balanced, but still high rate of false positives (72) and false negatives (78). There is room for improvement. The most successful model.

*Linear with cost=100:* The accuracy of this model was 69%, which was a great improvement, but still had a high rate of false positives (54) and false negatives (84). The model leaned more heavily to assigning a false negative or telling someone they were not injured when they were. This was surprising because the assumption was linear would return the best accuracy with the binary inquiry. There is room for improvement.

*Increasing the cost*

*Polynomial with cost=500:* The accuracy of this model was 58%, with a balanced, but still high rate of false positives (70) and false negatives (76). There is room for improvement.

*Sigmoid with cost=500:* The accuracy of this model was 56%, with a balanced, but still high rate of false positives (75) and false negatives (77). There is room for improvement, but this is still the most accurate and balanced result.

*Linear with cost=500:* The accuracy of this model was 60%, with an unbalanced, high rate of false positives (56) and false negatives (83). There is room for improvement.

*Daily Kilometer Sample Set*

Increased the sample set to 2000 in hopes of getting some more accuracy in the original models.

*Increased Sample Set*: Resulted in double or triple the amount of support vectors than the initial daily dataset model, so this was not explored further.

Conclusion

# Further Analysis

Further analysis based on the initial cleaning and deep dive into the dataset would be beneficial. More data leads to better results, but it is also difficult to wrangle and balance in this area of study. Runners train in some form almost every day of the week, and sometimes more than once a day, so there are more instances of sessions than injuries. Looking at smaller samples of high risk to injury athletes, the weeks leading to an injury, and the terrain the runner was training on would be useful in predicting injury.

Models of interest from our report include:

* Decision tree
* Support vector machines, sigmoid kernel

Because the results we were testing and training for were binary a neural network, logistic regression, and a successful kNN would have been given important insights.

The sigmoid kernel of support vector machines gave the most accurate result of all the predicative models attempted. The accuracy was not particularly high, but there was a balance in the rate of false positive and negative results that hint that further tuning could make this a strong predictive model for injuries.

# Business decisions

GPS data from a runner's watch could greatly impact injury prevention. Garmin and other watch brands can tell a runner they are overreaching or over training which may result in injury. The companies that continue to develop this technology could benefit from adding features to their watches or compatible apps to have runners track more data. Did they complete this run on a cushion track? Or on the road of a city? What shoes or insoles are they wearing and how old are they?

Businesses can also turn their attention to the variables and attributes that have the most important weight and information. If the target is time on feet the focus should the be kilometers ran. The length of the workout in hours and minutes could provide more insight to this overuse study. Perceived training data also had a weight in models, like the decision tree. If a runner feels they are not training successfully they will overreach in most cases, especially in competitive running. Overreaching in training will lead to over training and possible strains and injuries. This could be a more physiological study competitive running or sports psychology could undertake.

References

McCallun, K. (June 13, 2022), The 6 Most Common Running Injuries & How to Prevent Them. [https://www.houstonmethodist.org/blog/articles/2022/jun/the-6-most-common-running-injuries-how-to-prevent-them/#](https://www.houstonmethodist.org/blog/articles/2022/jun/the-6-most-common-running-injuries-how-to-prevent-them/):~:text=The%20most%20common%20running%20injuries,hip)%2C%20typically%20felt%20when%20active

<https://irwinsmegastore.ie/blogs/news/history-of-activity-trackers-from-pedometers-to-fitbit-infographic>

<https://pure.rug.nl/ws/portalfiles/portal/183763727/_15550273_International_Journal_of_Sports_Physiology_and_Performance_Injury_Prediction_in_Competitive_Runners_With_Machine_Learning.pdf>

<https://www.kaggle.com/datasets/shashwatwork/injury-prediction-for-competitive-runners>

1. [↑](#footnote-ref-1)
2. [↑](#footnote-ref-2)