

# report (1)

December 10, 2019

## Table of Contents

- 1 Introduction
  - 1.1 Background
  - 1.2 Purpose of this project
- 2 Data Exploration
  - 2.1 Wellness Data
  - 2.2 Training Load Data(RPE Dataset)
  - 2.3 Performance Measure (GPS Dataset)
- 3 Model Selection and Fitting
  - 3.1 Variables Description
  - 3.2 Model Selection
    - 3.2.1 Linear Mixed Effects Model (LME)
    - 3.2.2 Model Assumptions
  - 3.3 Model Fitting
  - 3.4 Assumption Checking
  - 3.5 Inference Checking
- 4 Conclusion
  - 4.1 How to improve player's performance?
  - 4.2 What are the problems in our model?
- 5 Reference

## **1 Main Title**

## **2 Exploring the relationship between Wellness, Training load and Performance in the Canadian Women's Rugby 7s Team**

Sanghamesh S Vastrad [1005912096], Lixiang Wei[1001117374]

### **2.1 Introduction**

#### **2.1.1 Background**

The Canadian National Women's Rugby Team is interested in the relationship between training load, performance and wellness in Rugby 7s. Rugby 7s is a fast-paced, physically demanding sport that pushes the limits of athlete speed, endurance and toughness. Rugby 7s players may play in up to three games in a day, resulting in a tremendous amount of athletic exertion. Substantial

exertion results in fatigue, which may lead to physiological deficits (e.g., dehydration), reduced athletic performance, and greater risk of injury.

Despite the importance of managing training load in professional athletics, very little is known about its effects, and many training decisions are based on “gut feel.” Currently, training load is measured through a combination of subjective measurements (asking players how hard they worked) and objective measurements from wearable technology. Wellness is typically estimated by asking players how they feel in wellness surveys. However, there is no agreed-upon standard of defining wellness so the relationship between training load, performance and wellness is unclear.

### 2.1.2 Purpose of this project

1. Define appropriate measurement for wellness, performance, training load based on subjective survey, RPE, and GPS data sets
2. Predict player’s performance based on wellness and training load measurement
3. Come up with statistical inference to help team prepare for the game better

## 2.2 Data Exploration

### 2.2.1 Wellness Data

(Notebook: [WellnessMeasure.ipynb](#))

The original wellness data of athletes consists of 19 columns, of which there were 11 quantitative measures and 6 qualitative measures with the other two columns being Date and PlayerID. Our aim for this dataset is to come up with a quantitative summary measure(s). For understanding what each variable means, please refer to the [CodeBook](#).

Note: We decided to take two routes. One was that of factor analysis which can be found at the notebook over here: [Wellness-FactorAnalysis.ipynb](#) and the other which we are going to describe in this report. The one we describe over here turns out to be better in terms of summary measures being less correlated. The approach followed was to try to come up with one quantitative value and keep the categorical variables intact.

For complete understanding as to how we cleaned the data, please refer to this notebook: [WellnessMeasure.ipynb](#). In this notebook, we add all the quantitative values after scaling them from 1-7 and come up with a WellnessScore. But we soon realize that the data we had wasn’t completely independent. Intuitively, doesn’t our wellness today depend on how we felt yesterday or even a week before? So, we define a new score by applying exponential moving average to the WellnessScore. Now, each player’s ‘EWMScore’ will depend on a week of WellnessScores with exponentially decreasing weights as we move further back in time. The dataset looks like this:

[48]:

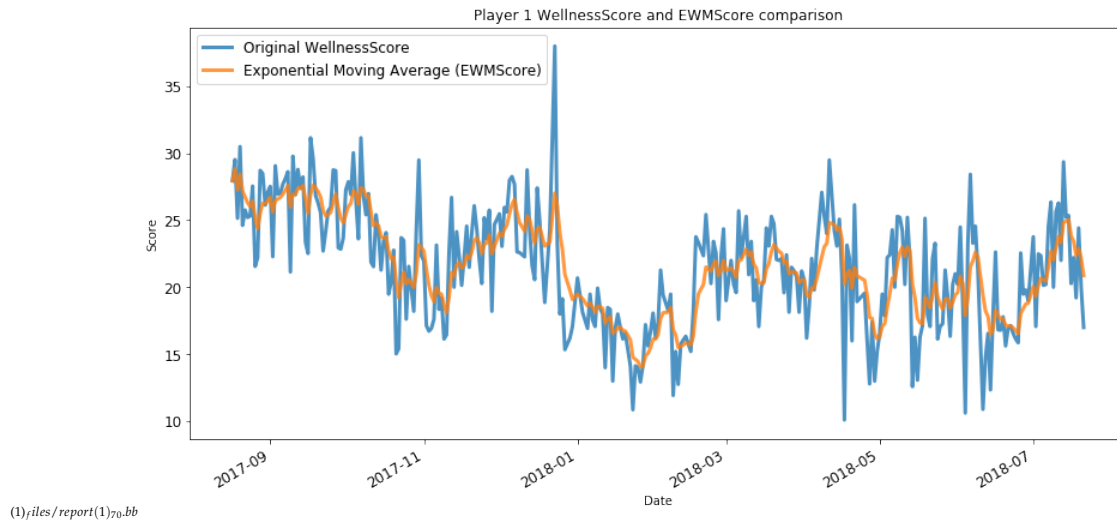
	Date	PlayerID	Pain	Illness	Menstruation	Nutrition	\
0	2017-08-01	14	No	No	No	Okay	
1	2017-08-01	2	No	No	No	Excellent	
2	2017-08-01	3	No	No	No	Excellent	
3	2017-08-01	5	No	No	No	Excellent	
4	2017-08-01	13	No	No	No	Okay	

	NutritionAdjustment	WellnessScore	EWMScore
0	Yes	34.000000	34.000000
1	Yes	35.824138	35.824138

2	Yes	30.000000	30.000000
3	Yes	31.596552	31.596552
4	Yes	29.862069	29.862069

Let us visualize the scores so that we can see what this actually means. The comparison of WellnessScore and EWMScore for Player with ID 1 is as follows:



## 2.2.2 Training Load Data(RPE Dataset)

(Notebook:[RPE\\_cleaning.ipynb](#))

The original wellness data of athletes consists of 14 columns, of which there were 7 quantitative measures and 5 qualitative measures with the other two columns being Date and PlayerID. Our aim for this dataset is to come up with a quantitative summary measure(s) to. For understanding what each variable means, please refer to the [CodeBook](#).

Note: We decided to take two routes. One was that of factor analysis (which can be found at the notebook over here: [RPE\\_cleaning.ipynb](#)) and the other which we are going to describe in this report. However, factor analysis does not work well. When we plot the new factors compared to AccelerationImpulse, we do not see any clear pattern. Hence, we decide to either come up with a new performance measure instead of AccelerationImpulse.

Firstly, we captured all the missing values from the same date, and cleaned the data. We dropped the columns ObjectiveRating, BestoutofSelf, and FocusRating, since there are too many missing values here. Besides, we removed all the rows still with missing values. Up to here, dimension of our RPE data set is (7621, 11). Secondly, we have players who did same training session multiple times. We would like to average their performance for the same session types. Finally, the dimension was reduced to (7192, 11).

After we came up with new performance measure, I investigated the relationship between some variables and new performance measure. Here is an example: I would like to see whether load status will affect player's performance. I conducted Tukey test, which is a pairwise group mean comparison.

[51]: <class 'statsmodels.iolib.table.SimpleTable'>

title

As we can see, the test for normal group and recovering group suggests there is no difference between group mean. However, we can see that high group has significant mean difference compared to normal group and recovering group. Also, the mean difference is negative, which means if a player has higher training load, it has higher performance score than other two groups.

### 2.2.3 Performance Measure (GPS Dataset)

(Notebook:[PerformanceMeasure.ipynb](#))

The aim for this dataset is to find a single performance measure per game for each player. We first thought of having a measure of a highest acceleration value per player, but it didn't show much variation and all the players performance seemed to be very similar.

We then came up with the idea of having a PerformanceScore, which was a score on the scale of 0-100 and would be calculated in the following manner:

Find out the frame of highest speed within each game for each player

Find the corresponding maximum values of accelLoad and accelImpulse if there are multiple frames with highest speed

Give Speed, AccelLoad and AccelImpulse equal weights and scale the score to 100.

We were also curious to see if the outcome of the game or the points difference had any impact on the players. We wanted to know if players perform better when their team is under pressure with a large points difference and if players performance better or worse when the team is winning or losing. So we added those measures into the dataset using the games data as well. Our final dataset looked like this:

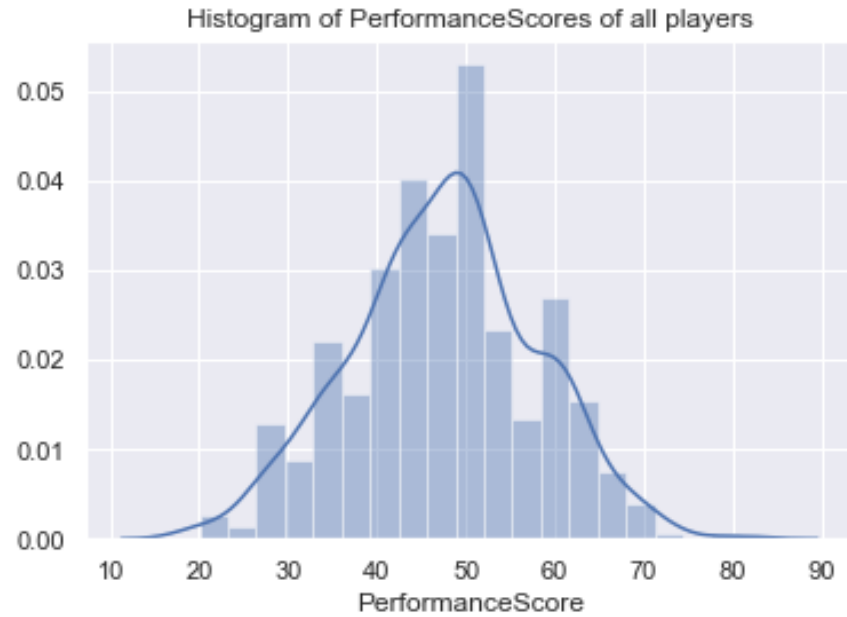
```
[31]:
```

	GameID	Date	PlayerID	AccelImpulse	AccelLoad	Speed	\
0	1	2017-11-30	2	0.524367	0.264378	0.754193	
1	2	2017-11-30	2	0.524367	0.264378	0.754193	
2	3	2017-11-30	2	0.524367	0.264378	0.754193	
3	1	2017-11-30	3	0.452520	0.333518	0.753256	
4	2	2017-11-30	3	0.452520	0.333518	0.753256	

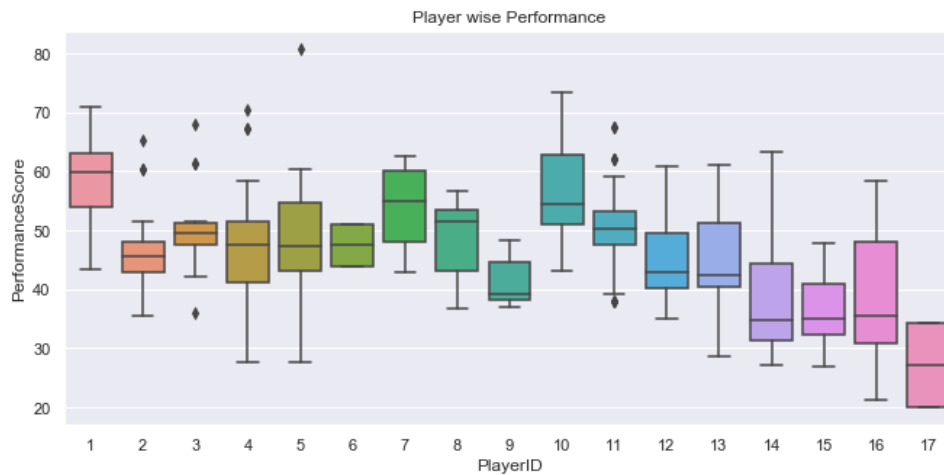
	PerformanceScore	Outcome	PointsDiff
0	51.431257	W	19
1	51.431257	W	31
2	51.431257	W	17
3	51.309794	W	19
4	51.309794	W	31

The PerformanceScore we came up with looks decent with values ranging from 17 all the way up to 80. The distribution of PerformanceScore is as follows:



(1)files/report(1)150.bb

We can also see from the Player wise PerformanceScore below that there is far more variability in our performance measure now. Also, note that we have removed players 18-21 as they didn't play any game.



(1)files/report(1)170.bb

## 2.3 Model Selection and Fitting

### 2.3.1 Variables Description

- PlayerID: A numerical number range from 1 to 17, indicating a unique playerID for each player

- SessionType: A categorical variable describing the training session types: Skills, Mobility/Recovery, Game, Conditioning, Strength, Combat, Speed
- Duration: A float representing session length, ranging from 3.00 to 120.00
- RPE: An integer representing rate of perceived exertion (0-10 scale)
- SessionLoad: A float obtained by  $\text{Duration} * \text{RPE}$
- DailyLoad: A float that is the sum of SessionLoad for a given day
- AcuteLoad: A float that is the average daily load over past 7 days
- ChronicLoad: A float that is the average daily load over past 30 days
- AcuteChronicRatio: A float that is calculated by  $\text{AcuteLoad} / \text{ChronicLoad}$
- Load\_Status: A categorical variable, which is dependent on AcuteChronicRatio. If ratio > 1.2, status is high. If ratio < 0.8, status is recovering. In between it is normal
- GameID: An integer representing a unique game ID, there are 38 games in total
- game\_date: A date that indicates the date of the game
- train\_date: A date that indicates the training date, which is one day ahead of game day
- AccelImpulse: A float that is the max absolute value of change in speed divided by change in time
- AccelLoad: A float that indicates the max load detected by the accelerometer
- Speed: A float that is the max movement speed of the player, in meters per second
- PerformanceScore: A measurement that is the weighted average of AccelImpulse, AccelLoad, and Speed. This measurement is considered as response variable in our model. It indicates how good the performance is for each player
- Outcome: A binary categorical variable, W indicates win, and L indicates loss.
- PointsDiff: An integer, which is the difference of scores. A positive difference indicates winning by how many points.
- Pain: A binary categorical variable(Yes or No) indicates whether a player is in pain
- Illness: A categorical variable indicates whether a player is feeling ill, all possible values are Yes, No, and Slightly Off
- Menstruation: A binary categorical variable(Yes or No) indicates whether a player is currently menstruating.
- Nutrition: A binary categorical variable(Excellent or Okay)
- NutritionAdjustment: A binary categorical variable indicates whether the player has made a nutrition adjustment that day
- EWMScore: A float number that measures the exponential moving average of wellness score

### 2.3.2 Model Selection

The typical regression model assumes all rows should be independent. However, in our data set, we have multiple observations(rows) for each player. These rows are dependent. We should not use typical linear regression.

Linear mixed models are an extension of simple linear models to allow both fixed and random effects, and are particularly used when there is non independence in the data, such as arises from a hierarchical structure.

When there are multiple levels, such as performances on different date from same player, the variability in the outcome can be thought of as being either within group or between group. Performance level observations are not independent, as within a given player performance are more similar.

The advantages of using mixed: 1. Keep as many rows of data as possible, instead of aggregating data by player. Although aggregate data analysis yields consistent and effect estimates and

standard errors, it does not really take advantage of all the data, because player data are simply averaged.

2. Even if we can run regression for each player, each model does not take advantage of the information in data from other players. This can cause some bias since we consider only for one specific player.
3. It keeps the flexibility in treating time as continuous.

**Linear Mixed Effects Model (LME)** A typical regression model:

$$Y = X\beta + \epsilon, \text{ where } X \text{ is fixed effects, } \epsilon \text{ is random noise}$$

Linear Mixed Effects Model:

$$Y = X\beta + Z\mu + \epsilon$$

Where  $y$  is a  $N \times 1$  column vector of the outcome variables,  $X$  is a  $N \times p$  matrix of the  $p$  predictor variables,  $\beta$  is a  $p \times 1$  column vector of the fixed-effects regression coefficients,  $Z$  is a  $N \times q$  design matrix for random effects,  $u$  is a  $q \times 1$  random complement to fixed  $\beta$ ,  $\epsilon$  is a  $N \times 1$  column vector of the residuals

### Model Assumptions

1. Performance variance should be homogeneous for all players
2. Linearity(Residuals should be random)
3. Normality(Residuals should form a normal distribution)

### 2.3.3 Model Fitting

Our goal is to predict a player's overall performance based on variables from wellness, rpe, gps data as described above. We would like to build a multiple linear mixed effects model on variables that are significantly impacting player's performance. In order to achieve this, we applied Backward Elimination. We will firstly build a full model based on all variables we have. Consider the feature with the highest P-Value. If its P-value is greater than significance level (We fix 0.1 as significance level), we will eliminate this variable and build a new model. Until every variable has p-value  $< 0.1$ , our model is ready.

The reason for choosing Backward Elimination: 1. Although backward elimination and forward elimination will give the same result, by using backward elimination, we can manually reduce our model step by step, instead of running a built-in function and only getting the final model 2. By reducing on variable at a time, we can also compare some other criteria to determine if the reduced model is as good as the previous model. For example, we can compare AIC, BIC, Likelihood, and Anova table.

In linear mixed effects model, in order to measure how good the model is, a useful technique is called REML – residual maximum likelihood. We will use this criteria to determine whether the reduced model is as good as full model.

Let's take a look at our full model

Mixed Linear Model Regression Results			
Model:	MixedLM	Dependent Variable:	PerformanceScore
No. Observations:	562	Method:	REML
No. Groups:	17	Scale:	58.6021

Min. group size:	3	Likelihood:	-1956.4791
Max. group size:	80	Converged:	Yes
Mean group size:	33.1		

	Coef.	Std.Err.	z	P> z	[0.025	0.975]
Intercept	50.766	8.679	5.849	0.000	33.755	67.777
SessionType[T.Game]	-0.501	2.385	-0.210	0.834	-5.175	4.173
SessionType[T.Mobility/Recovery]	0.331	2.774	0.119	0.905	-5.106	5.769
SessionType[T.Skills]	1.829	2.469	0.741	0.459	-3.010	6.668
SessionType[T.Speed]	2.498	3.187	0.784	0.433	-3.748	8.744
SessionType[T.Strength]	5.037	3.126	1.611	0.107	-1.090	11.163
Load_Status[T.normal]	-6.476	2.475	-2.617	0.009	-11.327	-1.626
Load_Status[T.recovering]	-4.472	3.045	-1.468	0.142	-10.440	1.497
Outcome[T.W]	-2.591	1.320	-1.963	0.050	-5.178	-0.004
Pain[T.Yes]	3.424	1.951	1.755	0.079	-0.399	7.248
Illness[T.Slightly Off]	4.863	1.910	2.547	0.011	1.120	8.606
Illness[T.Yes]	-10.834	5.806	-1.866	0.062	-22.215	0.546
Menstruation[T.Yes]	-0.170	1.288	-0.132	0.895	-2.695	2.355
Nutrition[T.Okay]	4.690	1.452	3.230	0.001	1.844	7.537
NutritionAdjustment[T.No]	6.456	4.876	1.324	0.185	-3.100	16.013
NutritionAdjustment[T.Yes]	8.217	3.508	2.342	0.019	1.341	15.092
Duration	0.026	0.042	0.626	0.531	-0.056	0.109
RPE	0.419	0.345	1.215	0.224	-0.257	1.095
SessionLoad	-0.014	0.007	-1.926	0.054	-0.028	0.000
DailyLoad	0.009	0.002	3.813	0.000	0.004	0.013
AcuteLoad	-0.034	0.016	-2.043	0.041	-0.066	-0.001
ChronicLoad	0.000	0.013	0.010	0.992	-0.025	0.026
AcuteChronicRatio	6.841	5.802	1.179	0.238	-4.532	18.213
PointsDiff	0.125	0.039	3.182	0.001	0.048	0.202
EWMScore	-0.412	0.244	-1.685	0.092	-0.891	0.067
Group Var	79.990	4.328				

As we can see, ChronicLoad has p-value 0.992, which is significantly larger than 0.1, we will drop it for next model. Keep in mind, the REML is about 58.6.

Since the process is repetitive, please see our notebook for more details. We will drop SessionType, Menstruation, RPE, SessionLoad, and Duration step by step. Again, even if we have insignificant p-value for some levels in a categorical variable, we still need to keep it if there exist at least some significant levels.

#### Mixed Linear Model Regression Results

Model:	MixedLM	Dependent Variable:	PerformanceScore
No. Observations:	562	Method:	REML
No. Groups:	17	Scale:	58.9085
Min. group size:	3	Likelihood:	-1962.3783



Max. group size: 80 Converged: Yes  
Mean group size: 33.1

	Coef.	Std.Err.	z	P> z	[0.025	0.975]
Intercept	58.509	7.704	7.594	0.000	43.409	73.609
Load_Status[T.normal]	-8.456	2.204	-3.836	0.000	-12.776	-4.136
Load_Status[T.recovering]	-8.037	2.309	-3.481	0.000	-12.563	-3.512
Outcome[T.W]	-2.655	1.311	-2.025	0.043	-5.225	-0.086
Pain[T.Yes]	3.473	1.864	1.864	0.062	-0.179	7.126
Illness[T.Slightly Off]	5.687	1.794	3.170	0.002	2.171	9.203
Illness[T.Yes]	-9.025	5.768	-1.565	0.118	-20.331	2.281
Nutrition[T.Okay]	4.851	1.401	3.461	0.001	2.104	7.597
NutritionAdjustment[T.No]	5.207	4.804	1.084	0.278	-4.210	14.623
NutritionAdjustment[T.Yes]	8.095	3.482	2.325	0.020	1.270	14.919
DailyLoad	0.006	0.002	2.852	0.004	0.002	0.010
AcuteLoad	-0.022	0.006	-3.566	0.000	-0.034	-0.010
PointsDiff	0.121	0.039	3.117	0.002	0.045	0.198
EWMScore	-0.412	0.213	-1.929	0.054	-0.830	0.007
Group Var	75.393	4.051				

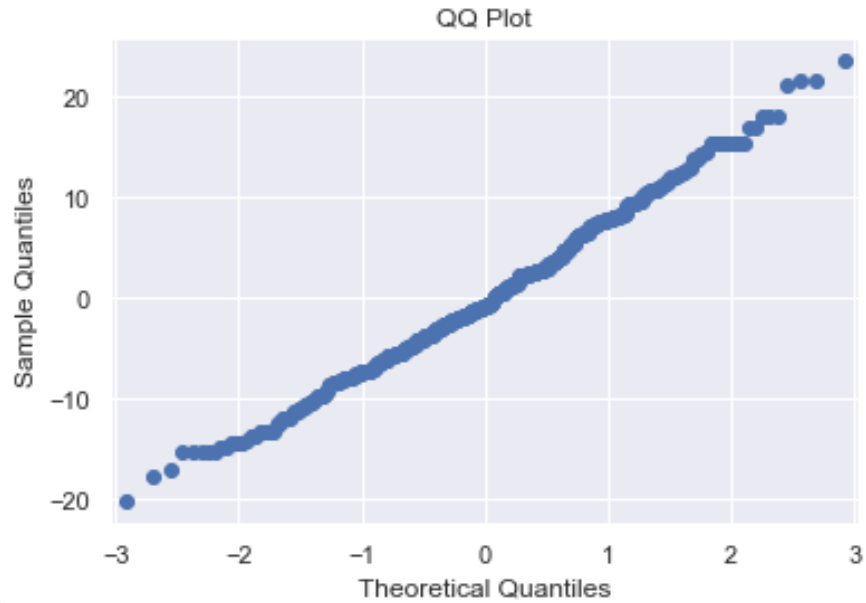
Up until this model, our REML climbs up to 58.9, and all variables or at least one level of categorical variable have p-value  $< 0.1$ , we confirm this is our final model.

Some Interpretations: 1. The coefficients for two levels of Load\_Status are about -8. The baseline level is high, so we can conclude if a player did a high load training before the game, their performance score is about 8 higher than normal training or recovering.

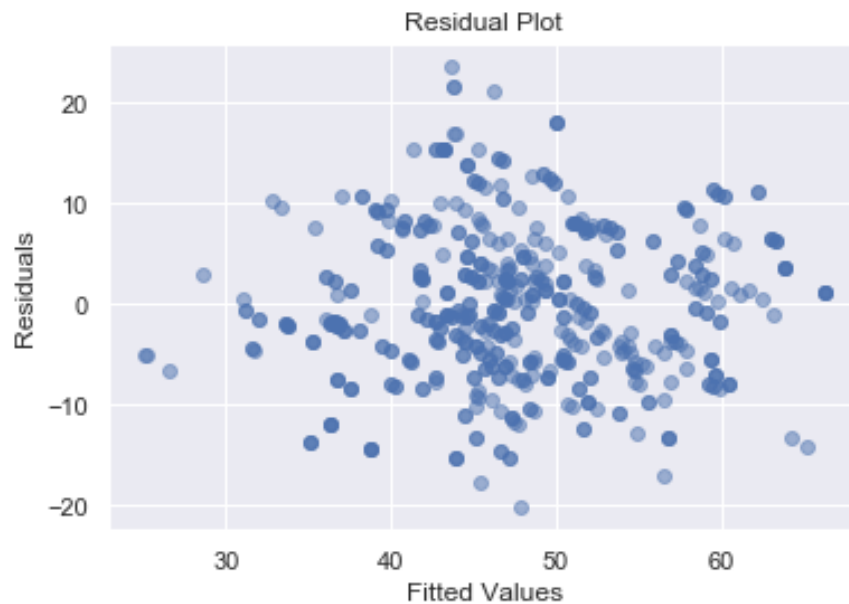
2. The coefficients for two levels of Illness are about 5.6 and -9. The baseline level is No, so we can conclude if a player is ill, her performance score is lower by 9. It is noticeable that if the player feels slightly off, her performance score is actually 5.67 higher.
3. EWMScore has coefficient -0.412, which indicates a negative relationship between performance and EWM score. This makes sense, since a high EWMScore means less soreness, fatigue. This indicates their training load is not high. Thus, it is consistent with first interpretation.

### 2.3.4 Assumption Checking

We will plot the residuals versus fitted value to verify if residuals are random. Also, qq-plot will check the normality assumptions.



(1)files/report(1)320.bb



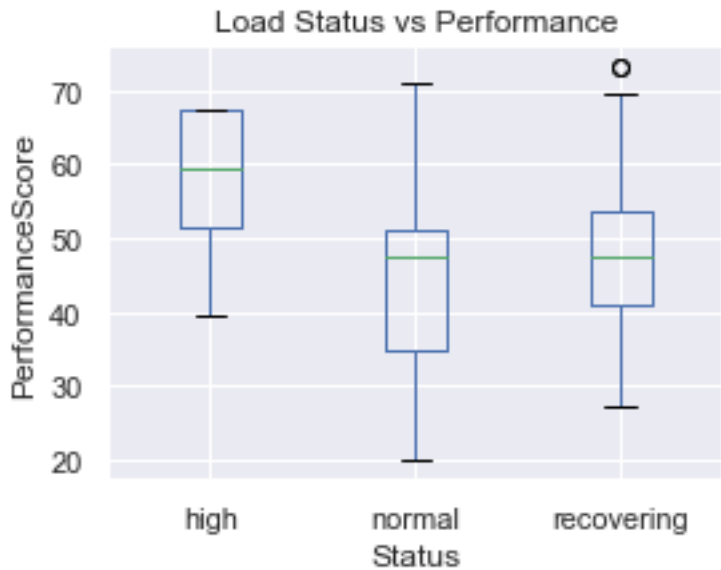
(1)files/report(1)330.bb

As we can see, the normality is good, as all data points are roughly on or closed to line. All residuals are pretty random, there is no apparent pattern or polynomial that can describe the residuals.

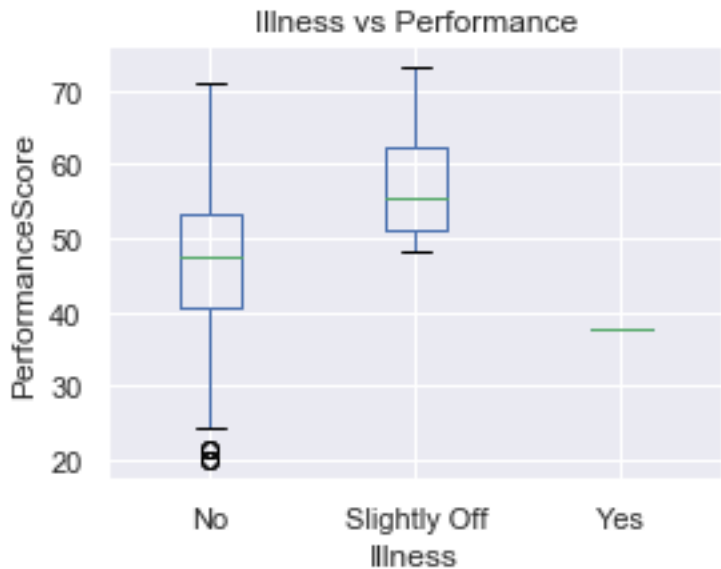
### 2.3.5 Inference Checking

After we conclude some associations between performance and other variables in our model, we would like to include some visualizations to verify our results. Here, we will plot Load Status,

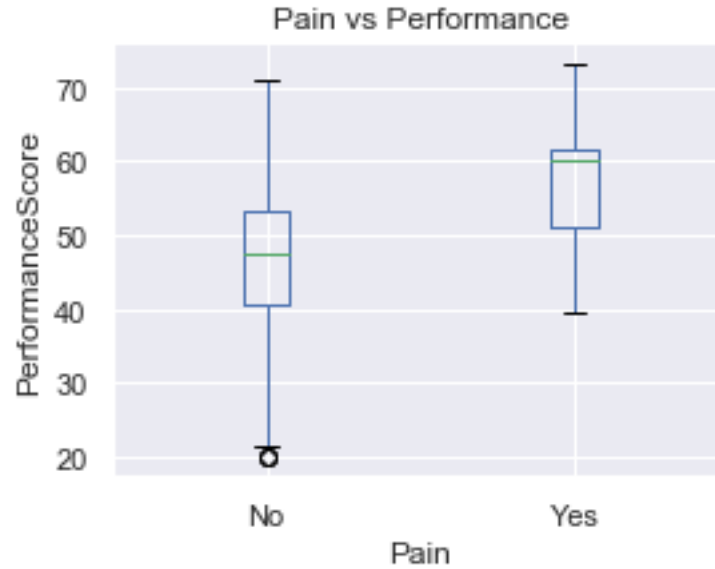
Illness, Pain, and Nutrition as x-axis, and performance score as response variable. As shown in the box plots, the player with high training load has significant better performance than other two status. The players feel sightly off performance slightly better than other two groups. But here, we only have 1 observation for ill player, we may doubt if a player feels ill, probably she will be out of tomorrow's games. If a player feels painful, she usually has better performance. This is reasonable, since pain is highly correlated with load status, if the status is high, most of player feels painful. For Nutrition, the difference is not significant, and the result is a bit surprised. Since a player has excellent nutrition has lower performance than those who is just okay.



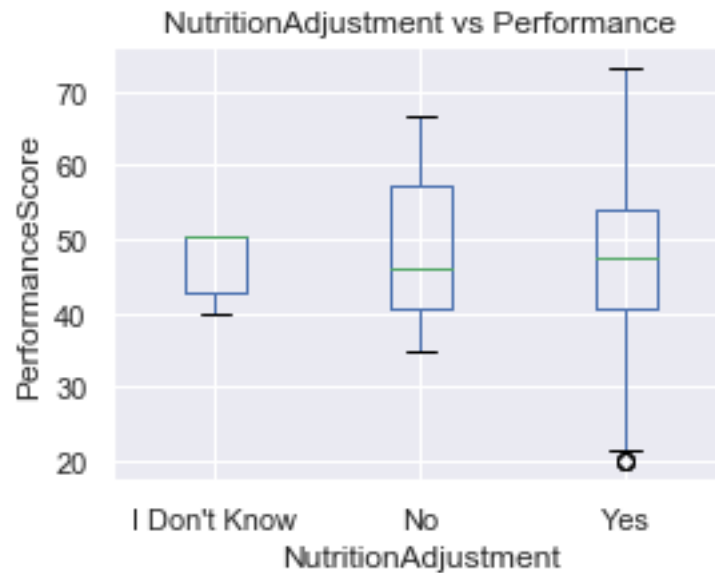
(1)files/report(1)360.bb



(1)files/report(1)361.bb



(1)*files/report(1)*<sub>362.bb</sub>



(1)*files/report(1)*<sub>363.bb</sub>

## 2.4 Conclusion

### 2.4.1 How to improve player's performance?

1. I would suggest players to have more acute exercise, and less chronic exercise, so that the AcuteChronic Ratio is higher. In this way, they would have high training load status.
2. NutritionAdjustment is important for player's performance. A player being self-aware about Nutrition Adjustment seems to performance better than those who don't know about

their nutrition levels. In our model, if the player has made nutrition adjustment for the training day, her performance is about 3 points higher on our performance scale.

#### **2.4.2 What are the problems in our model?**

1. Should not include game result and points difference to predict future performance
2. A player may have multiple types of training sessions on the day before game. We should build regression models grouped by session types
3. Better/Complete data like heart rate would have allowed us to use models like the Bannister model and we could have come up with better summary measures.

#### **2.5 Reference**

1. Avinash Navlani (2019), <https://www.datacamp.com/community/tutorials/introduction-factor-analysis>
2. Gabriela K Hajduk(2017), Introduction to linear mixed models <https://ourcodingclub.github.io/2017/03/15/mixed-models.html>
3. Gelman, A. & Hill, J. (2006). Data Analysis Using Regression and Multilevel/Hierarchical Models. Cambridge University Press
4. Singer, J. D. & Willett, J. B. (2003). Applied Longitudinal Data Analysis: Modeling Change and Event Occurrence. Oxford University Press