FRC Tech Foul Estimator

Jon Schutz, Pedare Robotics Mentor

April 27, 2018

In each FRC round, two alliances of three teams each compete, and scores are recorded for each alliance.

The purpose of this analysis is to estimate the rate of tech fouls *per team* based on the data available *per alliance*.

1 Introduction

FRC scoring statistics are recorded against the alliance of three teams rather than the individual team. It is useful to be able to get an estimate of the contribution of each team to the overall score, either to assess own-team performance, or to help decide who to pick as an alliance partner. Different methods are available for estimating the individual statistics; here we look at three different methods to determine which one is likely to give the best estimate of individual team tech fouls.

The three methods considered are denoted ML, LS and MAP, which respectively stand for Maximum Likelihood, Least Squares and Maximum A-Posteriori. The details of each are given in an appendix.

As the analysis will show, in a single competition there is not enough data to generate accurate individual team estimates. Having an estimate of the actual tech foul rates may not be the most important thing, however; in fact a straight ranking of teams in order of best to worst could be just as valuable in helping to gain an understanding of where each team stands.

Another consideration is that any team can have a good day or a bad day, and the statistics on the day may not be a good reflection of a team's real capacity. If we think of each team as having some underlying performance distribution (probability distribution), and that the scores at any given competition are just samples from a random number generator with that distribution, then from the data we have we can estimate the possible ranges for the true underlying average. The information that shows the difference between the on-the-day performance and the true underlying performance is hidden in the variance of the data.

In this analysis, the three estimation methods have been tested to find out:

- which is the best for estimating team scores for the on-the-day performance
- which is the best for estimating team rankings for the on-the-day performance

- which is the best for estimating team scores or team rankings for underlying performance

2 Test Data

To understand how well the estimators are working, some data is needed where we know the underlying performance and can observe actual team performance. For this we generate test data. Each of 30 teams is assigned an underlying performance (mean tech foul rate per game) and a competition is simulated where the tech fouls for each team are generated by a random number generator with a mean value equal to the underlying performance. The rounds of the competition are as run in the 2018 Southern Cross competition qualifying rounds, which was 30 teams and 65 matches of 2 alliances per match and 3 teams per alliance. Since the opposing alliance is not relevant for this analysis, the data can be treated as 130 rounds of 3 teams each.

3 Boostrapping

Running each method against one data set gives a certain answer, but one might be left wondering what would have happened if the data was a little bit different - each match is, after all, a random sample of each teams "true" ability and sometimes random chance falls a long way from the mean. "Bootstrapping" is a process that helps to estimate the sensitivity of the results to these random effects. Ideally, if it were possible to observe the true ability of each team and the random distribution that generates how each team performs on the day, the experiments could be run repeatedly and the distribution of results would show how much the estimates vary due to the random nature of the data. In reality we can not observe the random distribution that generates the data, but we do have one sample of it - the data itself. So by taking samples (with replacement) of the data available, we can create an approximation to the random distribution that generates the data. Using this approach, it is possible to generate the mean expected results and the range (e.g. the 10% to 90% percentile range) that the results lie in.

4 Test Results

The first step is to verify that all of the methods are working correctly (that is, there are no programming errors and the results are consistent with theory). To do this we provide the methods with a large amount of data and show that the results have a small error compared to the expected values. The second step is to try the methods with the data available in a single competition to see how the error compares to when there is a large amount of data. To this end the three methods were tested using synthetic data comprising:

SSE	vs. Underlying Mean		vs. Data Mean	
Method	500×	1×	500×	1×
Method 1 (Maximum Likelihood)	0.0028	**0.247**	0.0025	**0.399**
Method 2 (Least Squares)	**0.0015**	0.912	**0.0014**	0.621
Method 3 (MAP)	0.0023	0.251	0.0019	0.429

Table 1: Sum of Squared Errors (SSE) for each method for each experiment

- 500 competitions (130 rounds × 500 competitions)
- 1 competition (130 rounds)

For the three methods and two data sets, the resulting estimation errors (sum of squared error, SSE) are shown in table 1. The SSE against the underlying mean (the mean of the distribution used to generate the data) and the data mean (the mean of the bootstrapped values) are shown; the SSE against the underlying mean indicates the usefulness of the algorithm for predicting the teams true performance, and the SSE against the data mean indicates usefulness for predicting performance on the day of competition. For large amounts of data, the errors are relatively small for all methods but LS has the lowest error. For the small data set, LS is the worst and ML the best.

Figures 1 to 3 show, for each team, the mean estimated foul rate (dots) and the 10% to 90% range of estimated values for each, compared to the actual values used to create the synthetic data (red crosses), for the large data set.

Figures 4 to 6 show the same for the small data set. Clearly, for the small data set, the algorithms are unable to accurately recover the underlying rates as shown by the red crosses. The average of the actual samples of tech foul values generated for each team are shown as red dots; it can be seen that many are zero, so very little data is actually available to the estimators. The lack of data shows up weaknesses in the estimators; for LS, the credible intervals are about twice as large as for the other two methods, and frequently stretch into the infeasible negative region; the MAP estimates deviate very little from their initial values since the algorithm relies on its prior knowledge in the face of little data.

Table 2 shows the Spearman ranking coefficients for the same experiments. The Spearman ranking coefficient is an indicator of the relative ordering of two data sets regardless of their actual values. Two sets which have values that are ordered the same have a coefficient of 1, if ordered in exactly the opposite order the coefficient is -1, and if there is no correlation in the ordering the value will be 0. From the table, LS gives the best performance for low data and good performance in all cases; however, the coefficients are quite small in the low data case, so whilst LS is the best estimator it is still not particularly accurate.

The above results were conducted with one data set and therefore may have reflected the performance of the estimates just on that one data set. However, by repeating the same experiments 100 times (on the 1× competition set), it was confirmed that ML most consistently provided the best estimate in terms

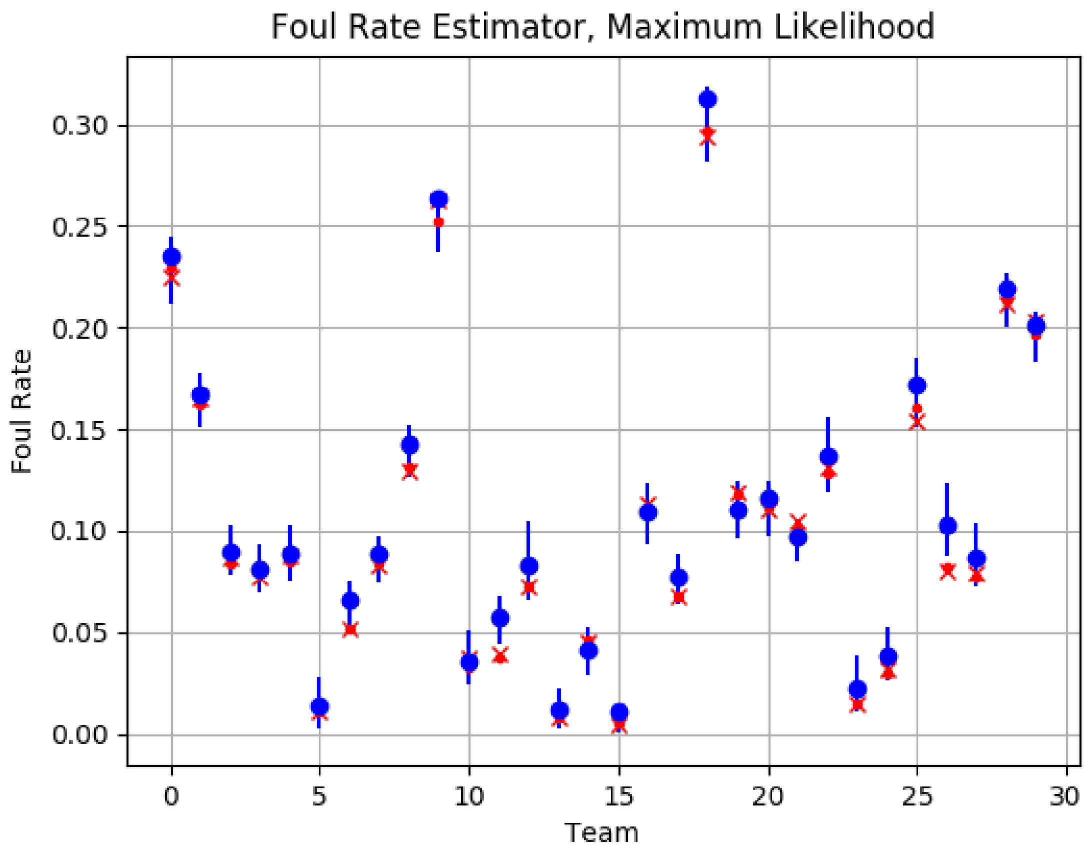

Figure 1: Maximum Likelihood estimate on synthetic data (500× competition data set). Estimates are blue dots, underlying means are red crosses, and actual data are red dots.

Spearman	vs. Underlying Mean		vs. Data Mean	
Method	500×	1×	500×	1×
Method 1 (Maximum Likelihood)	0.991	0.270	**0.989**	0.358
Method 2 (Least Squares)	0.992	**0.286**	0.987	**0.424**
Method 3 (MAP)	**0.993**	0.160	0.993	0.341

Table 2: Spearman Ranking Coefficient for each method for each experiment

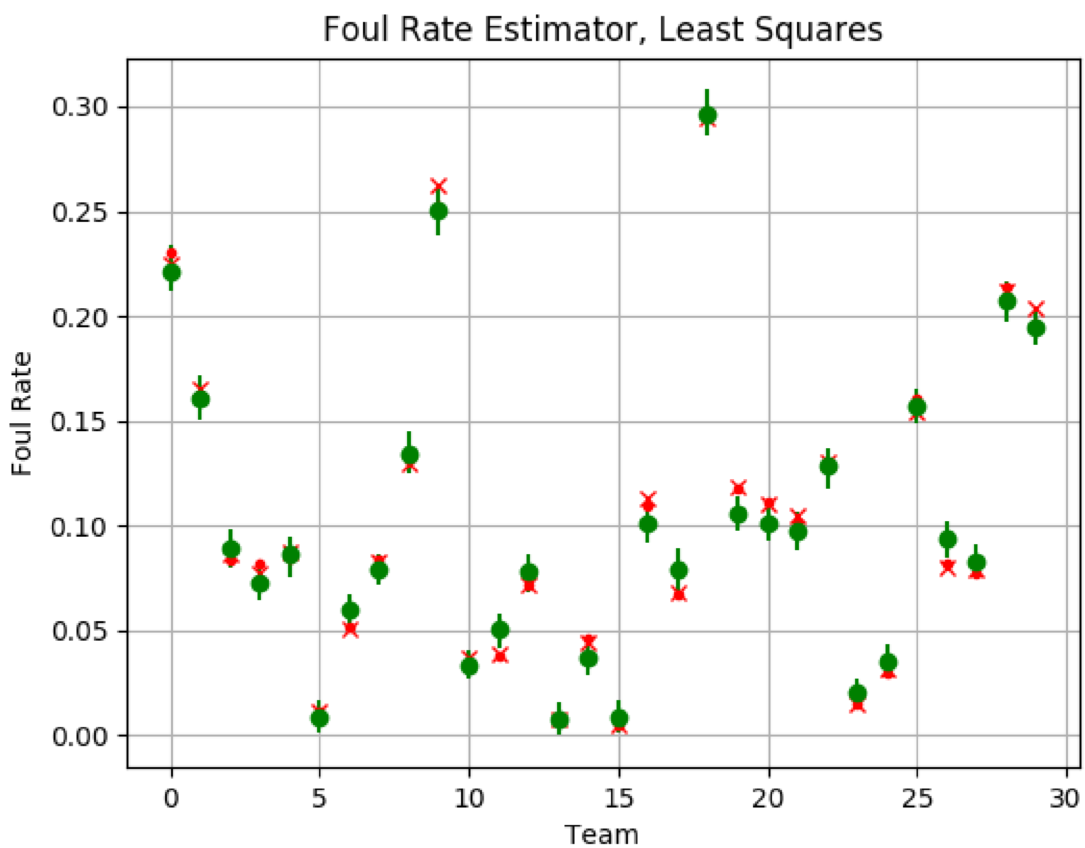

Figure 2: Least Squares estimate on synthetic data (500× competition data set). Estimates are green dots, underlying means are red crosses, and actual data are red dots

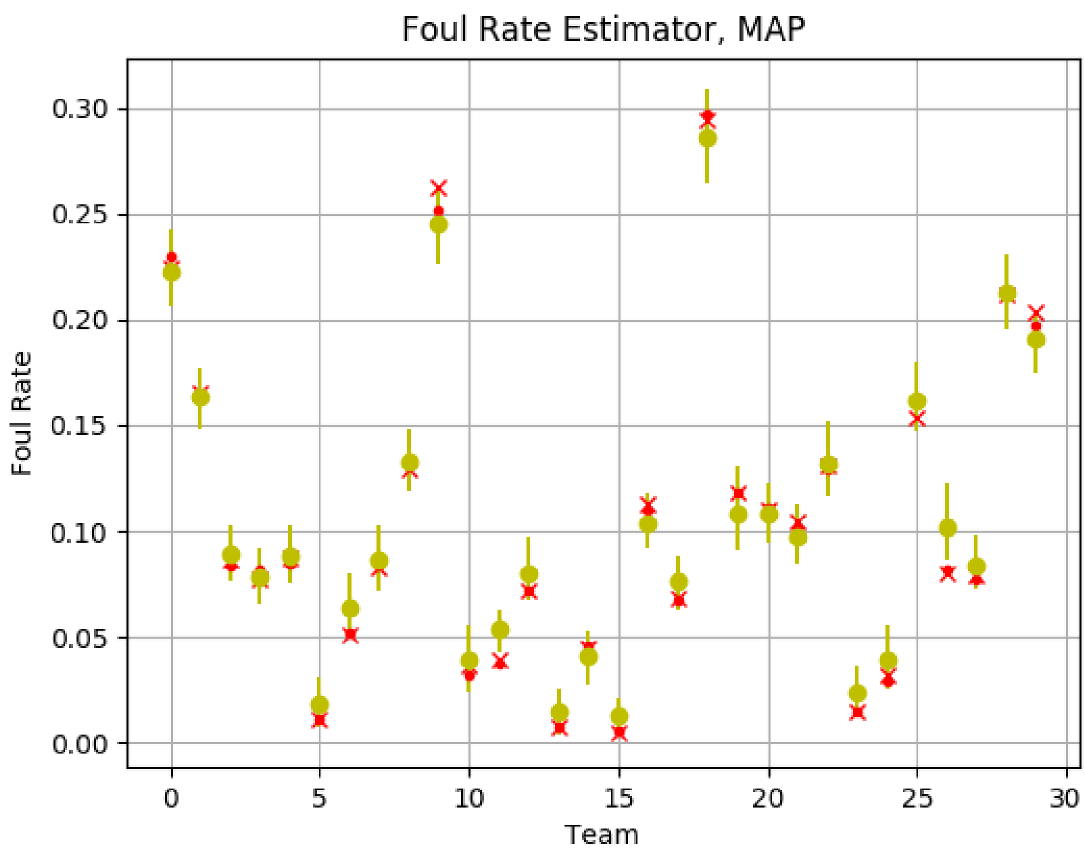

Figure 3: MAP estimate on synthetic data (500× competition data set). Estimates are light green dots, underlying means are red crosses, and actual data are red dots

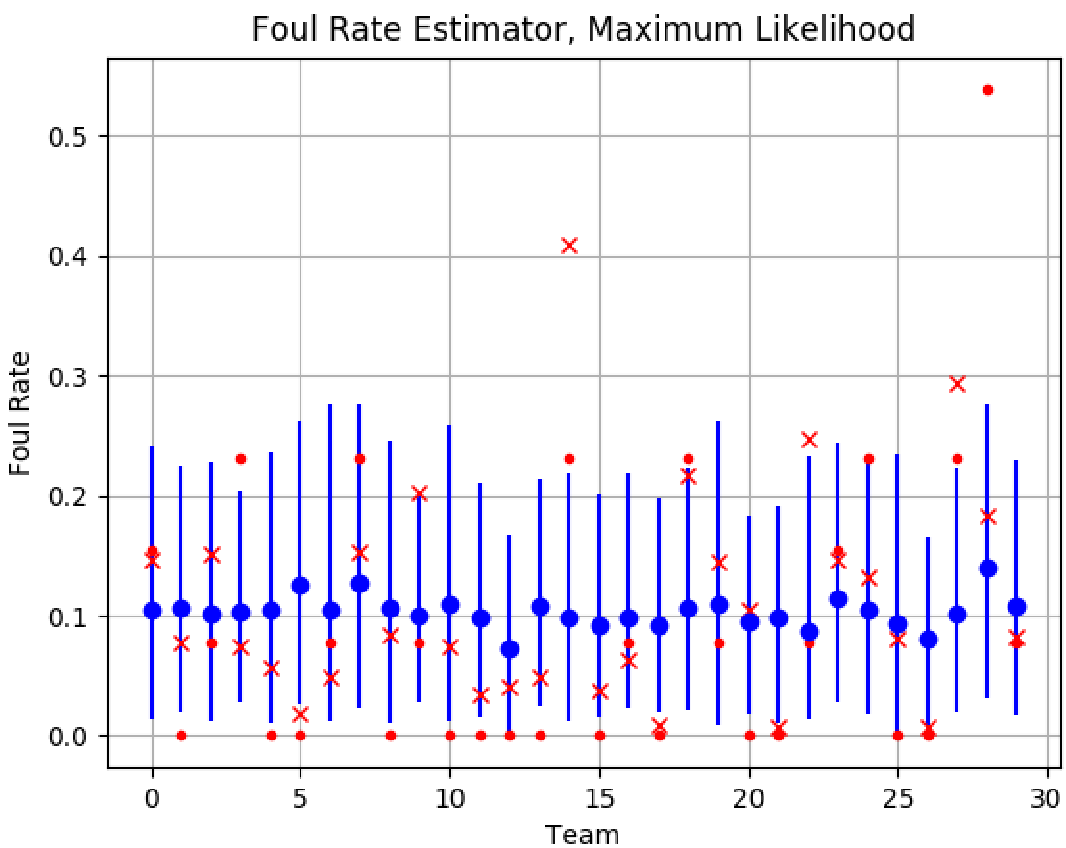

Figure 4: Maximum Likelihood estimate on synthetic data (single competition data set). Estimates are blue dots, underlying means are red crosses, and actual data are red dots

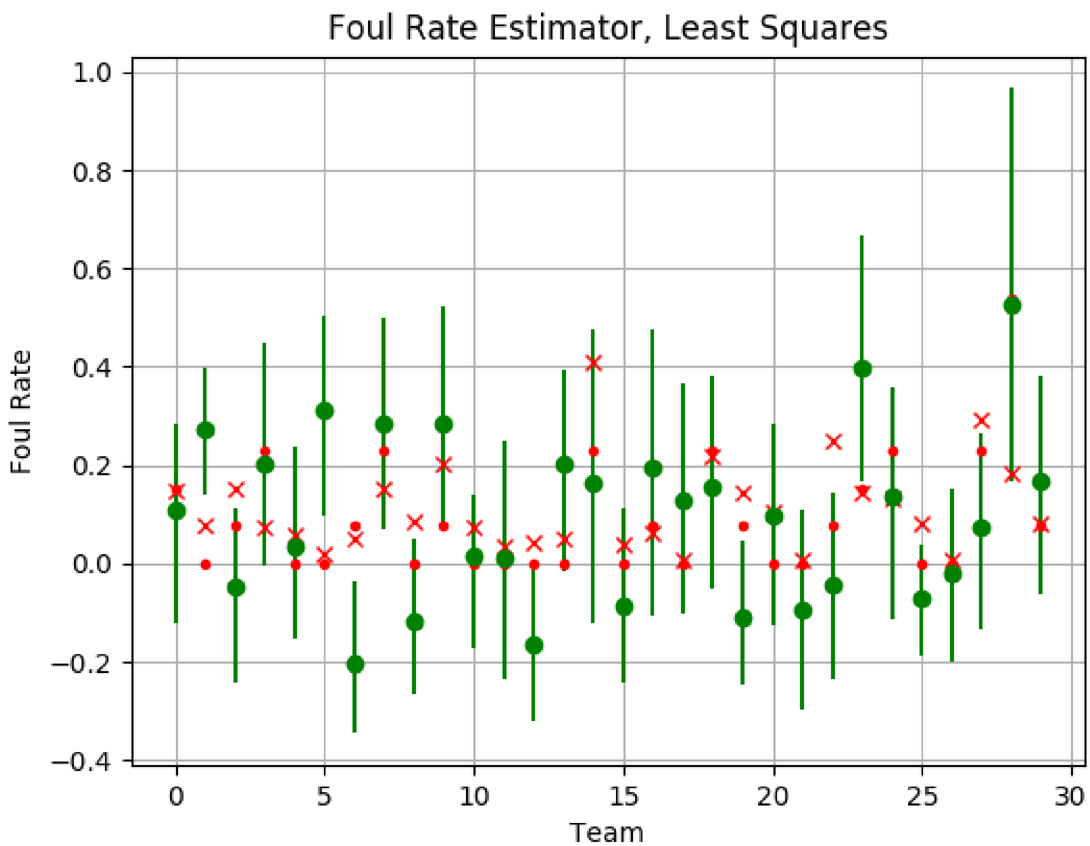

Figure 5: Least Squares estimate on synthetic data (single competition data set)

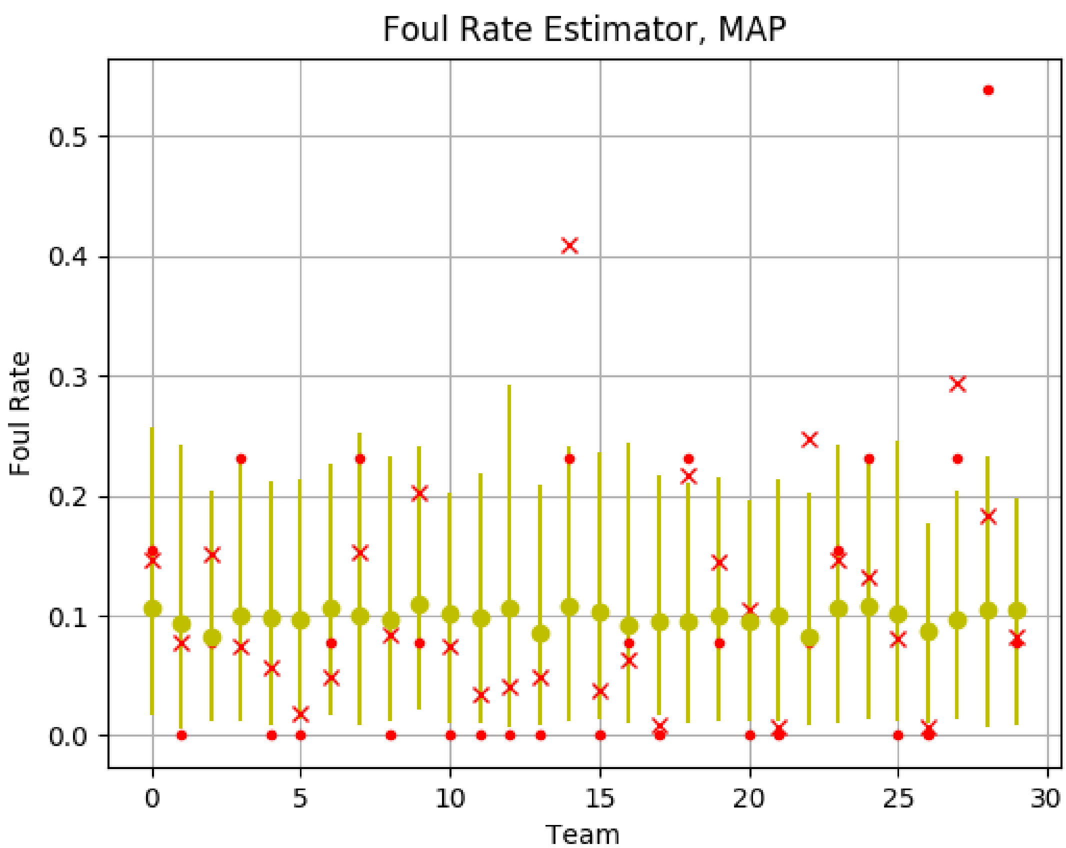

Figure 6: MAP estimate on synthetic data (single competition data set). Estimates are light green dots, underlying means are red crosses, and actual data are red dots

Wins 1×	SSE		Spearman	
	vs. Underlying Means	vs. Data Means	vs. Underlying Means	vs. Data Means
ML	**89**	**51**	19	9
LS	1	46	**77**	**87**
MAP	10	3	7	4

Table 3: Algorithms wins on SSE and Spearman over repeated tests

SSE	vs. Underlying Mean		vs. Data Mean		Spearman	vs. Underlying Mean		vs. Data Mean	
	Median	Range	Median	Range		Median	Range	Median	Range
ML	0.26	0.15 - 0.41	0.51	0.29 - 0.82	ML	0.28	0.04 - 0.5	0.40	0.19 - 0.63
LS	0.77	0.47 - 1.12	0.54	0.31 - 0.85	LS	0.44	0.23 - 0.61	0.60	0.46 - 0.73
MAP	0.29	0.17 - 0.46	0.56	0.31 - 0.88	MAP	0.05	-0.21 - 0.32	0.095	0.16 - 0.35

Table 4: Performance over repeated tests

of smallest error, and LS provided the best estimate in terms of rank ordering (table 3). table 4 summarises the estimates, showing median values and the 10% to 90% percentiles of the estimates over the 100 repeated tests.

5 Actual Results

Table 5 shows the actual results for the FRC 2018 Southern Cross competition, with values for all methods but shown ranked by LS. There does not appear to be any particular correlation between the teams foul ranking and their final position in the competition.

6 Conclusion

Having data for a single competition as the basis for estimating per-team statistics is a limitation, particular for tech fouls where the data is sparse. Nevertheless, the methods shown demonstrate some correlation between estimates and actual, and therefore provide some guidance - at least, a level of guidance which is better than none at all. The LS (least squares) method proved to be the most versatile for generating rankings in tech foul order, although ML (maximum likelihood) was better at estimating actual numbers of tech fouls.

A Method 1 (Maximum Likelihood, ML)

The first method has a sound theoretical foundation, but requires maximisation of a likelihood function so is slightly more complicated to implement.

Suppose that each team produces fouls at an average rate which is constant for that team and independent of all other teams. Thus the ith team produces

Rank	Team	ML	LS	MAP	Final Rank	Team Name
1	FRC 6520	0.09	-0.33	0.10	25	Hanoi Amsterdam School
2	FRC 6024	0.10	-0.26	0.11	11	R Factor
3	FRC 6579	0.09	-0.25	0.10	3	Komplete Kaos
4	FRC 4817	0.11	-0.15	0.11	28	One Degree North
5	FRC 4537	0.10	-0.12	0.11	12	RoboRoos
6	FRC 6434	0.11	-0.07	0.11	10	Bossley Park
7	FRC 6998	0.12	-0.07	0.12	9	NNKIEH
8	FRC 7278	0.11	-0.06	0.11	7	Toormina High School
9	FRC 4802	0.11	-0.03	0.11	17	Blacktown Girls
10	FRC 5983	0.11	-0.01	0.11	14	Blast Furnace bots
11	FRC 6187	0.10	-0.00	0.11	30	Narrabri
12	FRC 4774	0.12	0.01	0.13	26	The Drop Bears
13	FRC 6508	0.12	0.06	0.12	8	Hastings Heroes
14	FRC 4253	0.12	0.06	0.14	1	Raid Zero
15	FRC 3132	0.11	0.08	0.12	23	Thunder Down Under
16	FRC 4739	0.12	0.11	0.12	19	Thunderbolts
17	FRC 6836	0.14	0.16	0.14	5	The Tinkerers
18	FRC 7130	0.13	0.19	0.15	29	FABLAB
19	FRC 5331	0.14	0.22	0.13	21	Lightning Bolts
20	FRC 6083	0.13	0.25	0.13	16	Overlooking
21	FRC 6204	0.14	0.29	0.15	27	Pedare
22	FRC 7023	0.14	0.31	0.14	22	Oxley High School
23	FRC 7074	0.14	0.34	0.14	6	Reapers
24	FRC 4613	0.14	0.43	0.16	2	Barker Redbacks
25	FRC 6996	0.16	0.44	0.19	15	Koalafied
26	FRC 5593	0.16	0.47	0.16	13	UTAS
27	FRC 5985	0.16	0.48	0.16	4	Project Becephalus
28	FRC 6118	0.16	0.54	0.17	24	Dunedoo
29	FRC 6191	0.18	0.88	0.21	18	RoboKryptonite
30	FRC 3008	0.21	1.18	0.25	20	Team Magma

Table 5: Actual per team foul estimation results, shown ranked according to the LS method (lowest number of fouls first)

an average of λ_i fouls per round. The actual number produced must be an integer. The probability of team i with mean foul rate λ_i producing k_{ij} fouls in round j is described by a Poisson probability distribution,

$$P(k_{ij}) = \frac{\lambda_i^{k_{ij}}}{k_{ij}!} e^{-\lambda_i}$$

However, k_{ij} is not observed directly. Instead, the combination of the tech fouls of the three teams in the alliance is observed. Let the observed fouls in round j be F_j. The winner and red/blue alliance association is irrelevant to the number of tech fouls so for simplicity, let j run from 1 to J where J is double the number of actual rounds, and red alliance and blue alliance data are treated separately as one round each. So for round j the sum of tech fouls across the alliance members can be written as:

$$\sum_{i \in A_j} k_{ij} = F_j$$

where the notation $i \in A_j$ means to select just the teams which are members of the alliance for round j.

The sum of independent Poisson distributed variables is also Poisson distributed with a mean which is the sum of the means of the contributors. i.e. given $k_1 \sim Poisson(\lambda_1)$ and $k_2 \sim Poisson(\lambda_2)$, then $k_1+k_2 \sim Poisson(\lambda_1+\lambda_2)$. Thus for an alliance round $F_j \sim Poisson(\sum_{i \in A_j} \lambda_i)$.

To work out the most likely value for all of the λ_i, we write down the probability of observing all of the F_j that occurred in the match play, and then find the values for λ_i that maximise it (or the log probability).

Since we assume that all of the rounds are independent, the probability of observing $F_1$and F_2 and so on, is the product of the individual probabilities:

$$\begin{aligned} P(F_1, F_2, ..., F_J) &= \prod_{j=1}^{J} P(F_j) \\ &= \prod_{j=1}^{J} \frac{\left(\sum_{i \in A_j} \lambda_i\right)^{F_j}}{F_j!} e^{-\sum_{i \in A_j} \lambda_i} \end{aligned}$$

The log probability is:

$$\log P(F_1, F_2 ..., F_J) = \sum_{j=1}^{J} \left(F_j \log \sum_{i \in A_j} \lambda_i - \log F_j! - \sum_{i \in A_j} \lambda_i \right) \tag{1}$$

To find the maximum, differentiate with respect to each of the λ_i and equate to 0:

$$\frac{d \log P}{d\lambda_m} = \sum_{j: m \in A_j} \left(\frac{F_j}{\sum_{i \in A_j} \lambda_i} - 1 \right) \tag{2}$$

Here the sum is over all rounds in which team m participated. The above leads to M equations (M=number of teams) in M unknowns. However, they are non-linear and therefore it may be easier to maximise the log probability function directly. The derivative can be utilised to speed up convergence of the maximiser.

To calculate the log probability, the term $\sum_{i\in A_j} \lambda_i$ must be calculated. If the λ_i are stored in a vector Λ and the alliance data stored in an $J \times M$ indicator matrix X as follows:

$$\Lambda = [\lambda_1, \lambda_2, \ldots, \lambda_M]^T$$

$$X = \begin{bmatrix} 1 & 1 & 1 & & & & & \\ & & & 1 & 1 & 1 & & \\ & & & & & & \cdots & \\ & & & & & & & \cdots \\ & & & & & & & \end{bmatrix}$$

The columns of X correspond to a team and rows to a round. Thus a 1 appears in row j and column i if team i participated in round j. Then the vector $X\Lambda$ is the vector of $\sum_{i\in A_j} \lambda_i$ for $j = 1 \ldots J$.

Hence $\log P$ can be readily calculated given λ_i (ignoring the $\log F_j!$ term which does not affect the location of the maximum).

B Method 2 (Least Squares, LS)

Method 2 is more straightforward, but because it does not assume a model for the observations like Method 1, it is possible for it to yield negative results.

In this case, each round is modelled as:

$$\sum_{i\in A_j} \lambda_i + e_j = F_j$$

That is, the number of tech fouls per round is the sum of the means of each team plus a random error term. The objective is to find the λ_i which minimise the mean squared value of the error term.

Using the same matrix notation as the previous section, this can be written in matrix form as:

$$X\Lambda + E = F$$

where E the vector of error terms and F the vector of F_j of length J. This is a well-known least-squares problem which has solution:

$$\Lambda = (X^T X)^{-1} X^T F$$

C Method 3 (Maximum A-Posteriori, MAP)

In method 2, it was noted that it is possible for the foul rates to be negative. This highlights an issue; that we have prior knowledge of some of the characteristics of the rates - e.g. that they must be positive and that the average rate should be near the overall average across all rounds - but neither of the previous methods allowed this information to be incorporated into the model. Method 1 maximised the likelihood of the observations F_j; this method extends on Method 1 but instead maximises the probability of the estimates of the λ_m using our prior knowledge. Method 1 is known as Maximum Likelihood (ML) and Method 3 is known as Maximum A-Posteriori (MAP) because it maximises the posterior likelihood, or the likelihood that is the combination of prior knowledge with what we learn after observing data.

Thus we wish to maximise $P(\lambda_1,\lambda_2,\ldots\lambda_M|F_1,F_2,\ldots F_J)$ which we write as $P(\Lambda|F)$. We capture our prior knowledge of where we think λ_m should like by crafting another probability distribution for it; a convenient choice is to use a Gamma distribution with parameters α and β, and choosing the parameters so that the Gamma distribution has the characteristics we expect. Using Bayes' Rule:

$$P(\Lambda|F) = \frac{P(F|\Lambda)P(\Lambda|\alpha,\beta)}{\int P(F|\Lambda)P(\Lambda|\alpha,\beta)d\Lambda}$$

Maximising the log probability is the same as maximising the log of the numerator since the denominator is fixed (after integration, it is no longer a function of Λ). $\log P(F|\Lambda)$ is as given in equation (1). The second term is:

$$\log P(\Lambda|\alpha,\beta) = \sum_{m=1}^{M} \log P(\lambda_m|\alpha,\beta) \tag{3}$$

where $P(\lambda_m|\alpha,\beta)$ is the Gamma PDF evaluated at λ_m.

Thus the log probability to maximise is the sum of equations 1 and 3.

The derivative of the prior is:

$$\frac{d\log P(\Lambda|\alpha,\beta)}{d\lambda_m} = \frac{\alpha-1}{\lambda_m} - \beta \tag{4}$$

and the overall derivative is the sum of equations 2 and 4.