

# Bradley-Terry models in R: The BradleyTerry2 package

Heather Turner and David Firth

*University of Warwick, UK*

<http://go.warwick.ac.uk/heatherturner>

For BradleyTerry2 version 0.9-2, 2010-03-18

## Abstract

This is a short overview of the R add-on package BradleyTerry2, which facilitates the specification and fitting of Bradley-Terry logit, probit or cauchit models to pair-comparison data. Included are the standard ‘unstructured’ Bradley-Terry model, structured versions in which the parameters are related through a linear predictor to explanatory variables, and the possibility of an order or ‘home advantage’ effect or other ‘contest-specific’ effects. Model fitting is either by maximum likelihood, by penalised quasi-likelihood (for models which involve a random effect), or by bias-reduced maximum likelihood in which the first-order asymptotic bias of parameter estimates is eliminated. Also provided are a simple and efficient approach to handling missing covariate data, and suitably-defined residuals for diagnostic checking of the linear predictor.

**Key words:** generalized linear model, logistic regression, Jeffreys prior, penalised quasi-likelihood, ranking, tournament analysis, working residuals

## 1 Bradley-Terry model

### 1.1 Introduction

The Bradley-Terry model (Bradley and Terry, 1952) assumes that in a ‘contest’ between any two ‘players’, say player  $i$  and player  $j$  ( $i, j \in \{1, \dots, K\}$ ), the odds that  $i$  beats  $j$  is  $\alpha_i/\alpha_j$ , where  $\alpha_i$  and  $\alpha_j$  are positive-valued parameters which might be thought of as representing ‘ability’. For a good general introduction see Agresti (2002). Applications are many, ranging from experimental psychology to the analysis of sports tournaments to genetics (for example, the allelic transmission/disequilibrium test of Sham and Curtis (1995) is based on a Bradley-Terry model in which the ‘players’ are alleles). The model can alternatively be expressed in the logit-linear form

$$\text{logit}[\text{pr}(i \text{ beats } j)] = \lambda_i - \lambda_j, \quad (1)$$

where  $\lambda_i = \log \alpha_i$  for all  $i$ . Thus, assuming independence of all contests, the parameters  $\{\lambda_i\}$  can be estimated by maximum likelihood using standard software for generalized linear models, with a suitably specified model matrix. The primary purpose of the BradleyTerry2 package, implemented in the R statistical computing environment (Ihaka and Gentleman, 1996; R Development Core Team, 2003), is to facilitate the specification and fitting of such models, including special cases in which the ability parameters are related to available explanatory variables through a linear predictor of the form  $\lambda_i = \sum_{r=1}^p \beta_r x_{ir} + U_i$ . The logit link can be replaced, if required, by a different symmetric link function (probit or cauchit).

### 1.2 Example: analysis of journal citations

The following comes from page 448 of Agresti (2002), extracted from the larger table of Stigler (1994). The data are counts of citations among four prominent journals of statistics and are included in the BradleyTerry2 package as the dataset `citations`:

```
> library(BradleyTerry2)
> data(citations)
```

```
> citations
```

	cited	citing			
		Biometrika	Comm Statist	JASA	JRSS-B
	Biometrika	714	730	498	221
	Comm Statist	33	425	68	17
	JASA	320	813	1072	142
	JRSS-B	284	276	325	188

Thus, for example, *Biometrika* was cited 498 times by papers in *JASA* during the period under study. In order to fit a Bradley-Terry model to these data using *BTm* from the *BradleyTerry2* package, the data must first be converted to binomial frequencies. That is, the data need to be organised into pairs (*player1*, *player2*) and corresponding frequencies of wins and losses for *player1* against *player2*. The *BradleyTerry2* package provides the utility function *countsToBinomial* to convert a contingency table of wins to the format just described:

```
> citations.sf <- countsToBinomial(citations)
> names(citations.sf)[1:2] <- c("journal1", "journal2")
> citations.sf
```

	journal1	journal2	win1	win2
1	Biometrika	Comm Statist	730	33
2	Biometrika	JASA	498	320
3	Biometrika	JRSS-B	221	284
4	Comm Statist	JASA	68	813
5	Comm Statist	JRSS-B	17	276
6	JASA	JRSS-B	142	325

The binomial response can then be modelled by the difference in player abilities as follows:

```
> citeModel <- BTm(cbind(win1, win2), journal1, journal2, ~journal,
+   id = "journal", data = citations.sf)
> citeModel
```

Bradley Terry model fit by glm.fit

Call: BTm(outcome = cbind(win1, win2), player1 = journal1, player2 = journal2, formula = ~journal, id =

Coefficients:

journalComm Statist	journalJASA	journalJRSS-B
-2.9491	-0.4796	0.2690

Degrees of Freedom: 6 Total (i.e. Null); 3 Residual

Null Deviance: 1925

Residual Deviance: 4.293 AIC: 46.39

The coefficients here are maximum likelihood estimates of  $\lambda_2, \lambda_3, \lambda_4$ , with  $\lambda_1$  (the log-ability for *Biometrika*) set to zero as an identifying convention.

The *id* argument here gives a name to be used for the factor that identified the ‘players’ (here the four journals). The (necessarily one-sided) model formula

~ journal

here specifies that there is a separate parameter (or ‘ability’) for each level of that factor.

If a different ‘reference’ journal is required, this can be achieved using the optional *refcat* argument: for example, making use of *update* to avoid re-specifying the whole model,

```
> update(citeModel, refcat = "JASA")
```

Bradley Terry model fit by glm.fit

Call: BTm(outcome = cbind(win1, win2), player1 = journal1, player2 = journal2, formula = ~journal, id =

Coefficients:

journalBiometrika	journalComm Statist	journalJRSS-B
0.4796	-2.4695	0.7485

Degrees of Freedom: 6 Total (i.e. Null); 3 Residual

Null Deviance: 1925  
 Residual Deviance: 4.293 AIC: 46.39

— the same model in a different parameterization.

The use of the standard Bradley-Terry model for this application might perhaps seem rather questionable — for example, citations within a published paper can hardly be considered independent, and the model discards potentially important information on self-citation. [Stigler \(1994\)](#) provides arguments to defend the model’s use despite such concerns.

## 2 Abilities predicted by explanatory variables

### 2.1 ‘Player-specific’ predictor variables

In some application contexts there may be ‘player-specific’ explanatory variables available, and it is then natural to consider model simplification of the form

$$\lambda_i = \sum_{r=1}^p \beta_r x_{ir} + U_i, \quad (2)$$

in which ability of each player  $i$  is related to explanatory variables  $x_{i1}, \dots, x_{ip}$  through a linear predictor with coefficients  $\beta_1, \dots, \beta_p$ ; the  $\{U_i\}$  are independent errors. See, for example, [Springall \(1973\)](#) (but note that the error term  $U_i$  is omitted there). Dependence of the player abilities on explanatory variables can be specified via the *formula* argument, using the standard *S*-language model formulae. The difference in the abilities of player  $i$  and player  $j$  is modelled by

$$\sum_{r=1}^p \beta_r x_{ir} - \sum_{r=1}^p \beta_r x_{jr} + U_i - U_j, \quad (3)$$

where  $U_i \sim N(0, \sigma^2)$  for all  $i$ . The Bradley-Terry model is then a generalized linear mixed model, which the *BTm* function currently fits using the penalized quasi-likelihood algorithm of [Breslow and Clayton \(1993\)](#).

As a simple illustration, consider the response-surface model studied by [Springall \(1973\)](#):

```
> options(show.signif.stars = FALSE)
> data(springall)
> summary(springall.model <- BTm(cbind(win.adj, loss.adj), col,
+   row, ~flav[.] + gel[.] + flav.2[.] + gel.2[.] + flav.gel[.] +
+   (1 | .)), data = springall))
```

PQL algorithm converged to fixed effects model

Call:

```
BTm(outcome = cbind(win.adj, loss.adj), player1 = col, player2 = row,
    formula = ~flav[.] + gel[.] + flav.2[.] + gel.2[.] +
    flav.gel[.] + (1 | .), data = springall)
```

Deviance Residuals:

Min	1Q	Median	3Q	Max
-1.5306	-0.4086	0.2114	0.4454	0.9377

Coefficients:

	Estimate	Std. Error	z value	Pr(> z )
flav[.]	-0.411944	0.065948	-6.246	4.20e-10
gel[.]	-0.325776	0.102990	-3.163	0.00156
flav.2[.]	0.015650	0.006372	2.456	0.01404
gel.2[.]	0.105062	0.019983	5.258	1.46e-07
flav.gel[.]	0.023759	0.008414	2.824	0.00475

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 327.945 on 36 degrees of freedom  
 Residual deviance: 15.468 on 31 degrees of freedom  
 AIC: 112.98

Number of Fisher Scoring iterations: NA

Differences from the fit reported in [Springall \(1973\)](#) are minor, arising from the different treatment of ties: here we have simply treated a tie as half a ‘win’ for each ‘player’, rather than using the [Rao and Kupper \(1967\)](#) model as [Springall \(1973\)](#) did.

In [Springall \(1973\)](#), the random effect (specified here via the term  $(1/. .)$ ) was omitted: that is, the quadratic response surface was assumed to predict ‘ability’ without error. In general such a zero-error assumption will be unrealistic. In this instance, though, including the random term turns out to make no difference to the results: the variance of the errors  $\{U_i\}$  is estimated as zero.

The special name “.” appears here as the default identifier for players, in the absence of a user-specified *id* argument. The predictor variables in this case are taken from the *springall\$predictors* data frame, whose rows are indexed by the levels of “.”.

## 2.2 Missing values

The contest data may include all possible pairs of players and hence rows of missing data corresponding to players paired with themselves. Such rows contribute no information to the Bradley-Terry model and are simply discarded by *BTm*.

Where there are missing values in player-specific *predictor* (or *explanatory*) variables which appear in the formula, it will typically be very wasteful to discard all contests involving players for which some values are missing. Instead, such cases are accommodated by the inclusion of one or more parameters in the model. If, for example, player 1 has one or more of its predictor values  $x_{11}, \dots, x_{1p}$  missing, then the combination of (1) and (3) above yields

$$\text{logit}[\text{pr}(1 \text{ beats } j)] = \lambda_1 - \left( \sum_{r=1}^p \beta_r x_{jr} + U_j \right),$$

for all other players  $j$ . This results in the inclusion of a ‘direct’ ability parameter for each player having missing predictor values, in addition to the common coefficients  $\beta_1, \dots, \beta_p$  — an approach which will be appropriate when the missingness mechanism is unrelated to contest success. The same device can be used also to accommodate any user-specified departures from a structured Bradley-Terry model, whereby some players have their abilities determined by the linear predictor but others do not.

For an illustration of this device in action, see *example(flatlizards)*: two of the lizards in that study (*lizard096* and *lizard099*) have some missing covariate data, and those two lizards therefore have their abilities estimated by separate coefficients.

## 2.3 Order effect

In certain types of application some or all contests have an associated ‘bias’, related to the order in which items are presented to a judge or with the location in which a contest takes place, for example. A natural extension of the Bradley-Terry model (1) is then

$$\text{logit}[\text{pr}(i \text{ beats } j)] = \lambda_i - \lambda_j + \delta z,$$

where  $z = 1$  if  $i$  has the supposed advantage and  $z = -1$  if  $j$  has it. (If the ‘advantage’ is in fact a disadvantage,  $\delta$  will be negative.) The scores  $\lambda_i$  then relate to ability in the absence of any such advantage.

As an example, consider the baseball data given in [Agresti \(2002\)](#), p438:

```
> data(baseball)
> head(baseball)

  home.team away.team home.wins away.wins
1 Milwaukee  Detroit         4         3
2 Milwaukee  Toronto         4         2
3 Milwaukee  New York         4         3
4 Milwaukee   Boston         6         1
5 Milwaukee Cleveland         4         2
6 Milwaukee Baltimore         6         0
```

The dataset records the home wins and losses for each baseball team against each of the 6 other teams in the dataset. The *head* function is used to show the first 6 records, which are the Milwaukee home games. We see for example that Milwaukee played 7 home games against Detroit and won 4 of them. The ‘standard’ Bradley-Terry model without a home-advantage parameter is fitted as before:

```
> data(baseball)
> baseballModel1 <- BTm(cbind(home.wins, away.wins), home.team,
+   away.team, data = baseball, id = "team")
> summary(baseballModel1)
```

```
Call:
Btm(outcome = cbind(home.wins, away.wins), player1 = home.team,
    player2 = away.team, id = "team", data = baseball)
```

Deviance Residuals:

Min	1Q	Median	3Q	Max
-1.6539	-0.0508	0.4133	0.9736	2.5509

Coefficients:

	Estimate	Std. Error	z value	Pr(> z )
teamBoston	1.1077	0.3339	3.318	0.000908
teamCleveland	0.6839	0.3319	2.061	0.039345
teamDetroit	1.4364	0.3396	4.230	2.34e-05
teamMilwaukee	1.5814	0.3433	4.607	4.09e-06
teamNew York	1.2476	0.3359	3.715	0.000203
teamToronto	1.2945	0.3367	3.845	0.000121

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 78.015 on 42 degrees of freedom  
 Residual deviance: 44.053 on 36 degrees of freedom  
 AIC: 140.52

Number of Fisher Scoring iterations: 4

The reference team is Baltimore, estimated to be the weakest of these seven, with Milwaukee and Detroit the strongest.

To estimate the home-advantage effect, some re-organisation of the data is needed. The *at.home* variable is needed for both the home team and away team, so that it can be differenced as appropriate in the linear predictor.

```
> baseball$home.team <- data.frame(team = baseball$home.team, at.home = 1)
> baseball$away.team <- data.frame(team = baseball$away.team, at.home = 0)
> baseballModel2 <- update(baseballModel1, formula = ~team + at.home)
> summary(baseballModel2)
```

Call:

```
Btm(outcome = cbind(home.wins, away.wins), player1 = home.team,
    player2 = away.team, formula = ~team + at.home, id = "team",
    data = baseball)
```

Deviance Residuals:

Min	1Q	Median	3Q	Max
-2.03819	-0.40577	0.04326	0.61163	2.26001

Coefficients:

	Estimate	Std. Error	z value	Pr(> z )
teamBoston	1.1438	0.3378	3.386	0.000710
teamCleveland	0.7047	0.3350	2.104	0.035417
teamDetroit	1.4754	0.3446	4.282	1.85e-05
teamMilwaukee	1.6196	0.3474	4.662	3.13e-06
teamNew York	1.2813	0.3404	3.764	0.000167
teamToronto	1.3271	0.3403	3.900	9.64e-05
at.home	0.3023	0.1309	2.308	0.020981

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 78.015 on 42 degrees of freedom  
 Residual deviance: 38.643 on 35 degrees of freedom  
 AIC: 137.11

Number of Fisher Scoring iterations: 4

This reproduces the results given on page 438 of [Agresti \(2002\)](#): the home team has an estimated odds-multiplier of  $\exp(0.3023) = 1.35$  in its favour.

## 2.4 More general (contest-specific) predictors

The ‘home advantage’ effect is an simple example of a contest-specific predictor. Such predictors are necessarily interaction terms, between aspects of the contest and (aspects of) the two ‘players’ involved.

For more elaborate examples of such effects, see `?chameleons` and `?CEMS`. The former includes an ‘experience’ effect, which changes through time, on the fighting ability of male chameleons. The latter illustrates a common situation in psychometric applications of the Bradley-Terry model, where *subjects* express preference for one of two *objects* (the ‘players’), and it is the influence on the results of subject attributes that is of primary interest.

As an illustration of the way in which such effects are specified, consider the following model specification taken from the examples in `?CEMS`:

```
> data(CEMS)
> table8.model <- BTm(outcome = cbind(win1.adj, win2.adj), player1 = school1,
+   player2 = school2, formula = ~.. + WOR[student] * LAT[..] +
+   DEG[student] * St.Gallen[..] + STUD[student] * Paris[..] +
+   STUD[student] * St.Gallen[..] + ENG[student] * St.Gallen[..] +
+   FRA[student] * London[..] + FRA[student] * Paris[..] +
+   SPA[student] * Barcelona[..] + ITA[student] * London[..] +
+   ITA[student] * Milano[..] + SEX[student] * Milano[..],
+   refcat = "Stockholm", data = CEMS)
```

Here the subjects are students, and the objects (six European management schools) are referenced by “..”. The subject-specific variables `WOR`, `DEG`, etc., are found in the data frame `CEMS$students` which contains student-specific variables; the variable `LAT`, for example, is in the data frame `CEMS$schools` which contains school-specific variables.

## 3 Ability scores

The function `BAbilities` extracts estimates and standard errors for the log-ability scores  $\lambda_1, \dots, \lambda_K$ . These will either be ‘direct’ estimates, as in the standard Bradley-Terry model or for players with one or more missing predictor values, or ‘model-based’ estimates of the form  $\hat{\lambda}_i = \sum_{r=1}^p \hat{\beta}_r x_{ir}$  for players whose ability is predicted by explanatory variables.

As a simple illustration, team ability estimates in the home-advantage model for the `baseball` data are obtained by:

```
> BAbilities(baseballModel2)

      ability      s.e.
Baltimore 0.0000000 0.0000000
Boston    1.1438027 0.3378422
Cleveland 0.7046945 0.3350014
Detroit   1.4753572 0.3445518
Milwaukee 1.6195550 0.3473653
New York  1.2813404 0.3404034
Toronto   1.3271104 0.3403222
```

This gives, for each team, the estimated ability when the team enjoys no home advantage.

Similarly, ability estimates (for the nine experimental settings found in `springall.predictors`) can be obtained for the response-surface model of [Springall \(1973\)](#):

```
> BAbilities(springall.model)

PQL algorithm converged to fixed effects model

      ability      s.e.
1 -0.2415324 0.03750037
2 -1.6167549 0.19347841
3 -2.4398442 0.25061106
4 -0.3840280 0.15891106
5 -1.5197616 0.22979385
6 -2.1033621 0.22747359
7  0.6837863 0.22408367
8 -0.2124585 0.22083284
9 -0.5565702 0.18573246
```

The ability estimates in an un-structured Bradley-Terry model are particularly well suited to presentation using the device of *quasi variances* ([Firth and de Menezes, 2004](#)). The `qvcalc` package (version 0.8-5 or later) contains a function of the same name which does the necessary work:

```

> library(qvcalc)
> baseball.qv <- qvcalc(BTabilities(baseballModel2))
> plot(baseball.qv, levelNames = c("Bal", "Bos", "Cle", "Det", "Mil", "NY", "Tor"))

```

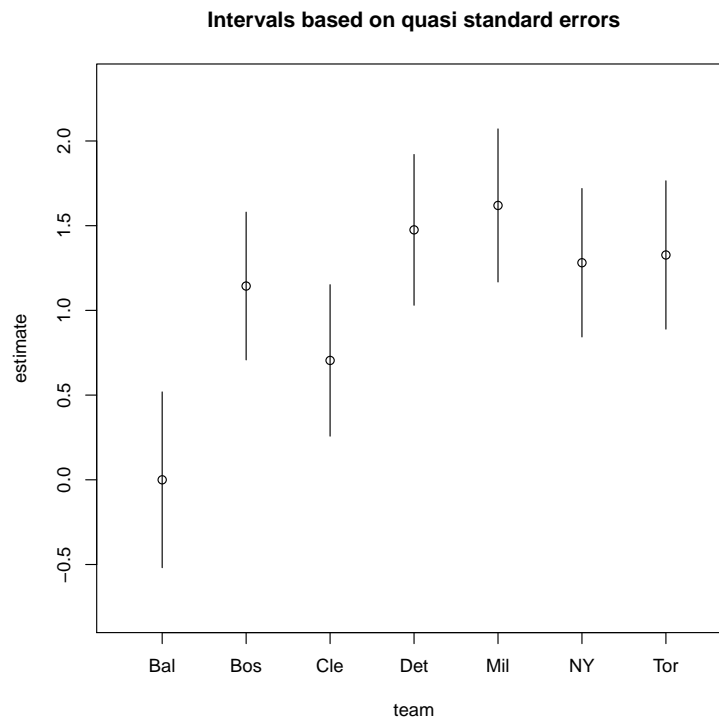


Figure 1: Estimated relative abilities of baseball teams

## 4 Residuals

There are two main types of residuals available for a Bradley-Terry model object.

First, there are residuals obtained by the standard methods for models of class *"glm"*. These all deliver one residual for each contest or type of contest. For example, Pearson residuals for the model *springall.model* can be obtained simply by

```

> res.pearson <- round(residuals(springall.model), 3)
> head(cbind(springall$contests, res.pearson))

```

	row	col	win	loss	tie	win.adj	loss.adj	res.pearson
1	1	2	2	16	7	5.5	19.5	0.225
2	1	3	0	21	1	0.5	21.5	-1.434
3	1	4	5	10	10	10.0	15.0	-0.649
4	1	5	2	15	7	5.5	18.5	0.133
5	1	6	0	22	2	1.0	23.0	-1.531
6	1	7	12	3	9	16.5	7.5	-0.308

More useful for diagnostics on the linear predictor  $\sum \beta_i x_{ir}$  are ‘player’-level residuals, obtained by using the function *residuals* with argument *type = "grouped"*. These residuals can then be plotted against other player-specific variables.

```

> res <- residuals(springall.model, type = "grouped")

```

These residuals estimate the error in the linear predictor; they are obtained by suitable aggregation of the so-called ‘working’ residuals from the model fit. The *weights* attribute indicates the relative information in these residuals — weight is roughly inversely proportional to variance — which may be useful for plotting and/or interpretation; for example, a

large residual may be of no real concern if based on very little information. Weighted least-squares regression of these residuals on any variable already in the model is null. For example:

```
> lm(res ~ flav, weights = attr(res, "weights"), data = springall$predictors)

Call:
lm(formula = res ~ flav, data = springall$predictors, weights = attr(res,
"weights"))

Coefficients:
(Intercept)          flav
  4.117e-09    -8.705e-10

> lm(res ~ gel, weights = attr(res, "weights"), data = springall$predictors)

Call:
lm(formula = res ~ gel, data = springall$predictors, weights = attr(res,
"weights"))

Coefficients:
(Intercept)          gel
 -3.041e-09    1.260e-09
```

## 5 Bias-reduced estimates

Estimation of the Bradley-Terry model in *BTm* is by default (when there are no random effects in the model) computed by maximum likelihood, using an internal call to the *glm* function. An alternative is to fit by bias-reduced maximum likelihood (Firth, 1993): this requires additionally the *brglm* package, and is specified by the optional argument *br = TRUE*. The resultant effect, namely removal of first-order asymptotic bias in the estimated coefficients, is often quite small. One notable feature of bias-reduced fits is that all estimated coefficients and standard errors are necessarily finite, even in situations of ‘complete separation’ where MLEs take infinite values (Heinze and Schemper, 2002).

## 6 Model search

In addition to *update()* as illustrated above, methods for the generic functions *add1()*, *drop1()* and *anova()* are provided. These can be used to investigate the effect of adding or removing a variable, whether that variable is contest-specific, such as an order effect, or player-specific; and to compare the fit of nested models.

## 7 Setting up the data

### 7.1 Contest-specific data

The *outcome* argument of *BTm* represents a binomial response and can be supplied in any of the formats allowed by the *glm* function. That is, either a two-column matrix with the columns giving the number of wins and losses (for *player1*), a factor where the first level denotes a loss and all other levels denote a win, or a binary variable where 0 denotes a loss and 1 denotes a win. Each row represents either a single contest or a set of contests between the same two players.

Any *contest-specific* variables should be of the same length as the variables specified in the *outcome*, *player1* and *player2* arguments. Sometimes this is achieved most economically by appropriate indexing: see, for example, *?CEMS*, where student-specific variables are stored in a data frame with one row per student rather than one row per ‘contest’.

An offset in the model can be specified by using the *offset* argument to *BTm*.

To use only certain rows of the data in the analysis, the *subset* argument may be used in the call to *BTm*. This should either be a logical vector of the same length as the binomial response, or a numeric vector containing the indices of rows to be used.

### 7.2 Player-specific data

Variables indexed by the levels of *player1* and *player2*, i.e., indexed by *id*, are said to be ‘player-specific’. The safest approach is to put all potential predictor (explanatory) variables — including factors and any offset term — into a data frame with one row per (potential) player, and with row names the names of players exactly as used in variables passed to the *player1* and *player2* arguments of *BTm*. Such data frame should then be included in the list specified as the *data* argument of *BTm* to specify where predictors (and any offset) can be found.



### 7.3 Converting data from the format required by the earlier BradleyTerry package

The BradleyTerry package described in [Firth \(2005\)](#) required contest/comparison results to be in a data frame with columns named *winner*, *loser* and *Freq*. The following example shows how *xtabs* and *countsToBinomial* can be used to convert such data for use with the *BTm* function in BradleyTerry2:

```
> library(BradleyTerry) ## the /old/ BradleyTerry package
> data(citations, package = "BradleyTerry") ## data frame with columns "winner", "loser", "Freq"
> citations <- xtabs(Freq ~ winner + loser, citations) ## convert to 2-way table of counts
> citations.sf <- countsToBinomial(citations) ## convert to a data frame of binomial observations
```

The *citations.sf* data frame can then be used with *BTm* as shown in Section 1.2.

## 8 A list of the functions provided in BradleyTerry2

The standard R help files provide the definitive reference. Here we simply list the main user-level functions and their arguments, as a convenient overview:

```
BTabilities(model)
BTm(outcome = 1, player1, player2, formula = NULL, id = "..",
    separate.ability = NULL, refcat = NULL, family = binomial,
    data = NULL, weights = NULL, subset = NULL, na.action = NULL,
    start = NULL, etastart = NULL, mustart = NULL, offset = NULL,
    br = FALSE, model = TRUE, x = FALSE, contrasts = NULL, ...)
countsToBinomial(xtab)
glmmPQL(fixed, random = NULL, family = binomial, data = NULL,
    subset = NULL, weights = NULL, offset = NULL, na.action = NULL,
    start = NULL, etastart = NULL, mustart = NULL, control = glmmPQL.control(...),
    sigma = 0.1, sigma.fixed = FALSE, model = TRUE, x = FALSE,
    contrasts = NULL, ...)
glmmPQL.control(maxiter = 50, IWLSiter = 10, tol = 1e-06, trace = FALSE)
```

## 9 A note on the treatment of ties

The present version of BradleyTerry2 provides no sophisticated facilities for handling tied contests/comparisons; the well-known models of [Rao and Kupper \(1967\)](#) and [Davidson \(1970\)](#) are not implemented here. At present the *BTm* function requires a binary or binomial response variable, the third ('tied') category of response is not allowed.

In several of the data examples (e.g., *?CEMS*, *?springall*, *?sound.fields*), ties are handled by the crude but simple device of adding half of a 'win' to the tally for each player involved; in each of the examples where this has been done it is found that the result is very similar, after a simple re-scaling, to the more sophisticated analyses that have appeared in the literature.

It is likely that a future version of BradleyTerry2 will have a more general method for handling ties.

## Acknowledgment

This work was supported by the UK Engineering and Physical Sciences Research Council.

## References

- Agresti, A. (2002). *Categorical Data Analysis* (Second edition). Wiley.
- Bradley, R. A. and M. E. Terry (1952). Rank analysis of incomplete block designs I: The method of paired comparisons. *Biometrika* 39, 324–45.
- Breslow, N. E. and D. G. Clayton (1993). Approximate inference in generalized linear mixed models. *JASA* 88(421), 9–25.
- Davidson, R. R. (1970). On extending the Bradley-Terry model to accommodate ties in paired comparison experiments. *Journal of the American Statistical Association* 65, 317–328.
- Firth, D. (1993). Bias reduction of maximum likelihood estimates. *Biometrika* 80, 27–38.

- Firth, D. (2005). Bradley-terry models in R. *Journal of Statistical Software* 12(1), 1–12.
- Firth, D. and R. X. de Menezes (2004). Quasi-variances. *Biometrika* 91, 65–80.
- Heinze, G. and M. Schemper (2002). A solution to the problem of separation in logistic regression. *Statistics in Medicine* 21, 2409–2419.
- Ihaka, R. and R. Gentleman (1996). R: A language for data analysis and graphics. *Journal of Computational and Graphical Statistics* 5(3), 299–314.
- R Development Core Team (2003). *R: A language and environment for statistical computing*. Vienna, Austria: R Foundation for Statistical Computing. ISBN 3-900051-00-3.
- Rao, P. V. and L. L. Kupper (1967). Ties in paired-comparison experiments: A generalization of the Bradley-Terry model. *Journal of the American Statistical Association* 62, 194–204.
- Sham, P. C. and D. Curtis (1995). An extended transmission/disequilibrium test (TDT) for multi-allele marker loci. *Annals of Human Genetics* 59(3), 323–336.
- Springall, A. (1973). Response surface fitting using a generalization of the Bradley-Terry paired comparison model. *Applied Statistics* 22, 59–68.
- Stigler, S. (1994). Citation patterns in the journals of statistics and probability. *Statistical Science* 9, 94–108.