Journal of Statistical Software

MMMMMM YYYY, Volume VV, Issue II. *doi: 10.18637/jss.v000.i00*

Luce's Choice Axiom Isn't the Only Choice! Combinatorial Contest and Rank Probabilities Using the **Python** Winning Package

Peter Cotton
Intech Investments

Abstract

A subset of k items will be chosen from n according to values taken by n variables $X_1, \ldots, X_n$ interpreted as performances in a contest. Item i is chosen if $X_i \leq X^{(k)}$ where $X^{(k)}$ is the k'th order statistic. A numerical algorithm is presented for computing many k-combination choice probabilities quickly, for small k but large $n \gg k$ in the millions. Rank probabilities for k can also be computed.

Some users of the winning package may wish to calibrate models for latent X_i from partial information such as winning or losing probabilities. Others may prefer to supply arbitrary performance distributions. The analytical convenience of this method also survives the introduction of dependence in the X_i via a low-dimensional Copula.

Keywords: JSS, style guide, comma-separated, not capitalized, Python.

1. Introduction:

One observes that since the 1930s, bettors have taken a keen interest in the probability of two contestants finishing first and second in a race, in either order. Yet over the intervening ninety years, and despite the popularity of this so-called quinella wagering format, no fast, compelling, deterministic means of estimating all quinella probabilities has been disseminated permitting arbitrary performance modeling.

One might even argue that no approach has been published permitting the use of *any* plausible underlying model. Models that are computationally convenient have relied on, or been shown to be equivalent to, somewhat quizzical assumptions about the random variables representing performance.

1.1. Explicit contests

Quinellas provide but one example of what would seem to be a glaring hole in combinatorial contest probability calculation - a gap that the Python winning package (Cotton (2021)) hopes to address. That said, the main commercial interest in this type of calculation may eventually lie far from the racetrack.

To use the term broadly, races or contests with rank-induced rewards are hardly uncommon - they are an integral part of commerce and, increasingly, interactions between people, machines and systems. A keyword search for documents might be considered a race and the importance of finishing on page one of Google's results is rather obvious.

Likewise, manufacturers compete for distribution but most channels will limit product range to a manageable number $k \ll n$. A sealed bid auction with k winners is a reasonable approximation for the trading of fungible securities subject to market impact, and sometimes $k > 1$.

As an example, a client might solicit several bids and offers, and then split flow amongst the k most competitive respondents. Setting this aside it is also true that on some electronic trading venues, additional information will be disseminated only to those whose bids or offers are deemed competitive.

Students applying to colleges face a contest with multiple winners, even if the yardstick and performance variable may vary greatly in its transparency from one country to another. (In Australia a single quantitative score holds sway whereas in the United States bespoke evaluations are performed).

Government and private tender processes can resemble contests with more than one rank-induced reward also, assuming that the task is not likely to be completed by a single respondent. Similar questions arise when athletes wish to make their way through qualifying rounds, or to job applicants, or to anyone, or anything hoping to make a shortlist of any variety.

To draw another example from internet commerce, hundreds of billions of contests are performed daily when web pages contained space for advertising are opened. The browser initiates a contest between automated bidding engines looking to supply content. These can fan out to ad-exchanges. Intermediate steps in this pipeline feature reward for $k > 1$ respondents.

Both sides of that problem are interesting. When deciding on a subset of possible engines or counterparties to reach out to, the marginal return of doing so must be weighed against the cost. Fast algorithms relating the latent variables X_i to partial information (such as fill percentages) are vital.

In other settings the performance variable may be price, quality or some combination of the two. The variables $X_1, \ldots, X_n$ represent a measure of selection desirability called "worth" in some settings, "fitness", "ability" or even "liability".

This can assist strategy, especially if there is an ability to shift the performance distribution at some cost. A fast, noiseless calculation makes what-if scenarios, and the computation of marginal rewards, far more convenient than the alternative, Monte Carlo.

Rating and ranking models might also employ the algorithms in the winning package. Sporting teams or students participating in exams, are assessed based on longitudinal performance across many contests whose participation varies. It is possible to assess the likelihood of underlying ability models quickly using the approach presented herein.

1.2. Implicit contests and choice

Situations where the performance variable X_i is observable do not comprise the sole motivation for this work. Even when choice of k items from n is not ostensibly an exercise in drawing random variables and ranking them, a fictious race between latent variables can be a strongly motivated way to approach the modeling of choice. This offers coherent probabilities when we compare subset selection probabilities in overlapping groups of items.

The opinion is expressed that latent variable models for preference can be much more popular in different fields such as education, political science and digital commerce if they are more tractable for large n. One hopes to explain decisions, or provide hypothetical probabilities for item group selections, based on limited data.

This is evident in small n use cases. For instance if only a single choice of candidate is made on a ballot, one can nonetheless try to infer the hypothetical outcome if preferential voting were used instead - or if a candidate were to drop out?

Needless to say contest theory might be usefully extended to large n scenarios too. All-pay auctions, where only k participants receive an item but everybody pays regardless (in work output) are a common object of study. It can be problematic to assess the marginal return on effort in large n contests using only Monte Carlo.

1.3. Comparison to Plackett-Luce models and software

In the computation of combinatorial selection probabilities, Plackett Luce models are commonly employed due to the convenience of Luce's Choice Axiom on which they rest. The trivial calculations for quinellas, trifectas and other wagers are implemented in various packages such as the R package called Horse Package Mashaal (2021). The R Package called PlackettLuce (Turner, van Etten, Firth, and Kosmidis (2020)) includes a treatment of ties.

This convenience belies a weakness we have noted. One is asked to *assume* renormalized sampling without replacement, following the path taken by Harville (1973) and in line with one type of Bradley-Terry model (Bradley and Terry (1952)). So the utility of this entire class of software is subject to the same caveat as noted upfront in Luce's original work Cane and Luce (1960).

If horses are replaced by radioactive particles then there is no reason to question the notion that the probabilty of a favoured horse finishing second, conditional on a longshot finishing first, is almost the same as the favoured horse's prior winning probability. However, the reality is different and most racegoers would be aware that the favourite is less likely to finish second than Luce's Choice Axiom would suggest - at least in this scenario.

Elsewhere too, the burden of proof lies squarely with those who would wish to assume Luce Choice Axiom applies. Instead, the winning package provides a less normative approach, and permits any underlying performance model.

1.4. Comparison to other approaches and software

The algorithm presented here is general yet scalable in n. After an initial computation involving all participants is performed (and sometimes a calibration prior to the same) the marginal cost of computing $p(s)$, the probability of subgroup $s \in S(n,k)$ of cardinality k is chosen, is shown to be small and independent of n.

The performance of the calibration procedure was the subject of Cotton (2021a). Here we focus on the subsequent use in determining combinatorial outcome probabilities.

As also noted in Cotton (2021a) the iterative method in the Python winning package, presented herein, differs markedly from prior art and scales to $n \gg 100000$. Methods that rely on multidimensional optimization cannot do the same, and this applies even when dimension reduction is employed, as with Cotton (2013).

2. Models and software

Figure 1 depicts some hypothetical performance distributions for five contestants. It is assumed that variables can be approximated by random variables $\tilde{X}_1, \tilde{X}_2, \ldots$ where the $\tilde{X}_i$ are supported on an evenly spaced lattice. Expected prices of rank-determined contingent claims (paying 1 in the event of no tie, but proportionately less otherwise) will serve as an approximation to rank-probabilities for the continuous variables.

The key insight borrowed from Cotton (2021a) is the characterization of a group of contestants by two quantities: the density of the first order statistic, but also the multiplicity (expected score-contingent number of winning ties).

2.1. Review of multiplicity calculus

Dropping tildes we use F to denote the (cumulative) distribution function and f the density of variables taking values on a lattice. Both are considered vectors. This is not a serious loss of generality, provided the algorithm supports reasonably sized lattices, which it does. Nor is there loss of generality in assuming the distributions are supported on integers.

Following Cotton (2021a) we define, for any $f(\cdot) : \mathcal{Z} \to \mathcal{R}$ and any $a \in \mathcal{R}$ a shifted distribution $f^{\to a}(\cdot)$ also supported on the integers $\mathcal{Z}$:

$$f^{\to a}(j) := (1-r)f^{\to a}(j) + rf^{\to a+1}(j) \tag{1}$$

where $r = a - a$ is the fractional part of the shift a. This extends the obvious right shift operator applicable when a is an integer. Formerly $f^{\to a}(j) := f(j-a)$. If $f(\cdot)$ is the distribution for X then $f^{\to a}$ approximates the distribution $X + a$.

The quantity

$$p_i = E\left[\frac{\iota_{X_i = X^{(1)}}}{\sum_{j=1}^{n} \iota_{X_j = X^{(1)}}}\right] \tag{2}$$

is, ignoring ties, a winning probability for the i'th item (probability it is selected first). It is the winning state price. Here ι is the indicator function and the denominator counts ties.

Consider a subgroup A minimally characterized by $\Upsilon_A = (S_A(), m_A())$ where $S_A(j) = Prob(X^{(1)} > j) = 1 - F_A(j)$ is the survival function for the first order statistic, and here F_A is the cumulative distribution for the same. Obviously we can quickly determine f_A, the density of the first order statistic, from F_A by prepending zero and differencing the vectors.

As was justified in Cotton (2021a) the combination of two item groups A and B into a single group relies on the following estimate for combined multiplicity:

$$m_{A\cup B}(j) = \frac{m_A(j)f_A(j)S_B(j) + (m_A(j) + m_B(j))f_A(j)f_B(j) + m_B(j)f_B(j)S_A(j)}{f_A(j)S_B(j) + f_A(j)f_B(j) + f_B(j)S_A(j)} \tag{3}$$

The corresponding operation for $S's$ is mere multiplication, whereupon we recover winning distribution F_A also and density $f_{A\cup B}$. It is clear, at noted in that paper, that the density of the first order statistic and also the multiplicity can be determined for the entire race by repeated application.

Furthermore, the multiplicity relations can be inverted to enable us to remove the i'th contestant. The multiplicity with item i left out is related to the multiplicity with i left in as follows:

$$m_{\hat{i}}(j) = \frac{m(j)f_i(j)S_{\hat{i}}(j) + m(j)f_i(j)f_{\hat{i}}(j) + m(j)f_{\hat{i}}(j)S_i(j) - m_i(j)f_i(j)S_{\hat{i}}(j)}{f_{\hat{i}}(f_i + S_i)} \tag{4}$$

as also discussed in Cotton (2021a). The corresponding operation for survival functions is mere division, and thus we can characterize a race with one contestant removed.

Losing probability for asset i by removing it from the race, using formula 4. Then an approximate win state price can be estimated by considering all possible values j taken by X_i and also the lowest value j' taken by the other item performances. The density of the latter is $f_{\hat{i}}$ and

$$\begin{aligned} p_i &= \sum_{j,j'} f_i(j)f_{\hat{i}}(j')E\left[\frac{W}{M}|X_i = j, X_{\hat{i}}^{(1)} = j'\right] \\ &\approx \sum_{j} f_i(j)\left\{\frac{f_{\hat{i}}}{1 + m_{\hat{i}}(j)} + S_{\hat{i}}(j)\right\} \end{aligned} \tag{5}$$

as shown in more detail in Cotton (2021a).

2.2. The illusive quinella calculator, and generalization.

Although the calculus above ostensibly applies when comparing one horse against the rest, it is also possible to compare the worst performance amongst a subgroup A to the best performance of the complement $B = \{1, \ldots, n\} \setminus A$. The accounting for multiplicity in the discrete variables brings their state prices into close alignment with the probabilities for their continuous counterparts.

So as an example, a fast numerical method for estimating quinella probabilities under arbitrary performance distribution assumptions, for independent performances, can proceed as follows:

1. Remove two horses i and j from the race using repeated application of Equation 4.
2. Compute winning distribution and multiplicity from *reversed* performance distributions for the group comprising two horses i and j only. This characterizes their group's worst performance and tie probability.
3. Compare the worst performance of these two horses to the rest of the runners, represented by the density and multiplicity achieved in the first step.

Clearly the quinella example generalizes to any subgroup of cardinality $k > 2$ by repeated application of Equation 4. Contests with hundreds of thousands of participants are amenable to this procedure.

2.3. Rank probabilities

Let $S(k, n)$ denote the set of all of k-combinations and $p(s)$ the probability, computed as above, that a particular k-combination corresponds to the first k items chosen. Also let $p_{i,j}$ denote the probability that item i is chosen j'th (i.e. that X_i is the j'th smallest value. Then we have

$$\sum_{j=1}^{k} p_{i,j} = \sum_{s \in S(j,k) | i \in s} p(s) \tag{6}$$

and both equal the probability that item i is selected when all items compete for the top k spots. We can use the right hand side to compute the left, and since this holds for every k we can then compute $p_{i,j}$ by subtraction.

To place this approach in horseracing terms for $k = 2$: the probability that a horse finishes second is equal to the probability it finishes in the top two minus the probability it wins, and the former is a sum of quinella probabilities.

By symmetry, the probability of a k-combination of items being drawn last can also be computed, merely by reversing the performance distributions. Working from the front and the rear, we can compute all group selection probabilities subject only to the combinatorial increase in the number of terms on the right hand side of Equation 6 - a burden that represents the limitation of this approach for large k.

This computational caveat aside, it is clear that for moderate n *all* rank probabilities can be computed if the sum on the right hand side extends to all sets of cardinality $\frac{n-1}{2}$, since in the case of odd n it is only necessary to compute up to $k = (n-1)/2$. (Again speaking loosely if we know the probabilities that a horse finishes first, second, though seventh and also last through seventh-last, we clearly know the probability it finishes eighth in a 15 horse race. The same rationale applies with ties and state prices.)

2.4. Dependence

Having established a route to the computation of group and rank probabilities in the case of independent performances, we now consider the case where performances are dependent. Depending on the computational budget and structure assumed, quadrature can be used. The example of a Gaussian copula with common off-diagonal correlation $0 \leq \rho < 1$ is given.

The performance X_i of the i'th contestant can be assumed to satisfy

$$X_i = F_i^{-1}\left(\Phi(Z_i)\right)$$

where F_i is the cumulative distribution on the lattice (the 1-margin), Φ is the standard cumulative normal distribution, and Z_i is an auxiliary random variable with standard normal distribution. (As written this is merely a restatement of the distributional transform, or the definition of F_i.)

In the independent case all assets are assumed to satisfy this generative model, with corresponding Z_i's that are independent and identically distributed. A simplistic dependence structure now introduces a connection between the Z_i's, such as

$$Z_i = \rho Z + \sqrt{1-\rho^2}\epsilon_i$$

where $Z \sim N(0,1)$ is a common factor, ρ is a correlation parameter and the ϵ_i's are independent N(0,1). All state prices can be viewed as iterated expectations, first conditioning on a choice of Z. If we let $F_i(\cdot; z)$ denote the cumulative distribution of the i'th asset return knowing $Z = z$ then rearranging we have

$$F_i(x; z) = \Phi\left(\frac{\Phi^{-1}(F_i(x)) + \rho z}{\sqrt{1-\rho^2}}\right) \tag{7}$$

Thus given any vector valued linear functional G acting on collections of densities, we can define $g(z)$ as the action of G on the set of densities transformed according to Equation 7 with parameter z. Then, the integral

$$\int_{z=-\infty}^{\infty} g(z)\phi(z)dz$$

can be estimated using standard methods such as Gaussian quadrature.

3. Examples

Five-way rank probabilities are shown in Table 2. This table uses a correlation parameter $\rho = 0.1$, whereas when $\rho = 0.25$ is considered instead slightly different rank probabilities are computed, reported in Table 3.

The results are quite similar. Because the common factor influences conditional probabilities in monotone fashion, the results are not overly sensitive to choice of correlation. The reader will observe that the rounded answers are almost identical to two significant digits.

The dependence structure can be generalized by allowing more than one factor to vary. The accuracy and computational convenience is then subject to the efficacy of two, three or higher dimensional quadrature schemes.

	Asset 1	Asset 2	Asset 3	Asset 4	Asset 5
location	-0.50	-0.25	0.00	1.00	1.50
scale	1.00	1.50	1.20	1.30	2.00

Table 1: Location and scale parameters for skew-normal return distributions used to illustrate the rank probability approach.

4. Examples and application to crowd-sourced forecasting

The repository Cotton (2021) contains the core algorithms and numerous examples. In this

	Rank 1	Rank 2	Rank 3	Rank 4	Rank 5
Asset 1	0.37	0.33	0.20	0.08	0.02
Asset 2	0.32	0.24	0.21	0.15	0.08
Asset 3	0.20	0.26	0.28	0.19	0.07
Asset 4	0.04	0.10	0.19	0.37	0.31
Asset 5	0.07	0.08	0.12	0.21	0.52

Table 2: Example five-way rank probabilities, $\rho = 0.1$.

	Rank 1	Rank 2	Rank 3	Rank 4	Rank 5
Asset 1	**0.38**	0.33	0.20	0.08	0.02
Asset 2	0.32	0.24	0.21	0.15	**0.07**
Asset 3	0.20	0.26	0.29	**0.18**	0.07
Asset 4	0.04	**0.09**	0.19	0.37	0.31
Asset 5	**0.06**	**0.08**	0.12	0.21	0.53

Table 3: Example rank probability computation, $\rho = 0.25$.

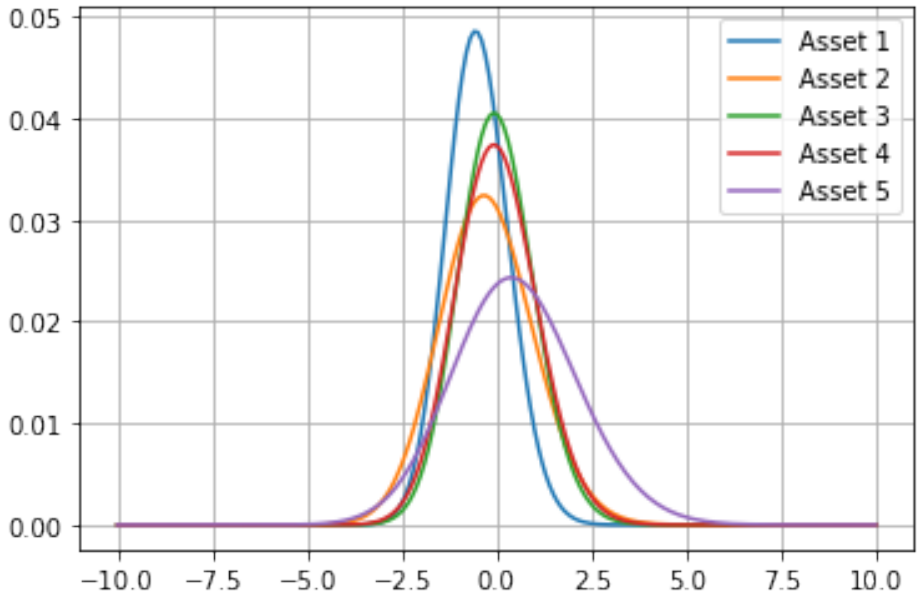

Figure 1: Skew-normal return distributions for five assets, used as an example for rank probability calculations.

	Rank 1	Rank 2	Rank 3	Rank 4	Rank 5
Asset 1	0.25	?	?	?	?
Asset 2	0.15	?	?	?	?
Asset 3	0.20	?	?	?	?
Asset 4	0.22	?	?	?	?
Asset 5	0.18	?	?	?	?

Table 4: A partially completed competition entry showing only worst-performer probabilities.

overview an example specific to the M6 Financial Forecasting competition is provided. Notebooks replicating the figures and tables herein are found in Cotton (2021b).

Prediction markets, financial markets and betting markets represent a crucial piece of information that can and should assist forecasting - but often this information represents only a slice, or a projection, or a margin, of some more important "full" picture worthy of forecasting.

The performances take on a financial interpretation - the realized return of a stock (the market, personified, chooses companies).

4.1. Relative location parameter inference

While group selection and rank probabilities may be desirable, they can be awkward to elicit from survey respondents. Elsewhere, rank probabilities might be only partially revealed by market prices.

Another example is provided by the M6 Financial Forecasting Duathlon. Participants are invited to submit five-way rank probabilities for stock returns. But it is certainly plausible that some would-be participants might shy away from this task, yet possess valuable information.

Similarly, published market share statistics for trading venues represent partial information, from which the aggressiveness of participants in their bidding and offering might need to be be inferred. An estimate of whether a particular bid or offer will be competitive can lean on the same calculus, as discussed in Cotton (2021a).

Similarly, betting markets for winning events are often more liquid, and might contain more information, then relatively illiquid markets for lower placings. A market might exist for the winner of an election, for example, leaving open the task of forecasting relative vote counts.

In some of these settings, imputation can exploit the fast algorithm provided. To illustrate with an example, Figure 4 depicts a partial rank probability table. Here an opinion has been expressed as to the probability that a stock will be the worst performing of the five, but no other information is supplied.

Suppose, for illustration, that we assume all five assets have similar volatilities and, for that matter, identical return profiles up to a translation. The imputation proceeds by calibrating the relative location parameters of the return distributions. Figure 2 depicts return distributions consistent with Table 4.

In other context, those distributions might represent the "ability" of an item to attract the attention of a consumer.

Using the procedure provided in Section 2 the corresponding five-way rank probabilities (state prices, to be pedantic) are reported in Table 5. Again, purely for illustration, the parameter $\rho = 0.25$ has been used. The penultimate column repeats the assumed probabilities taken

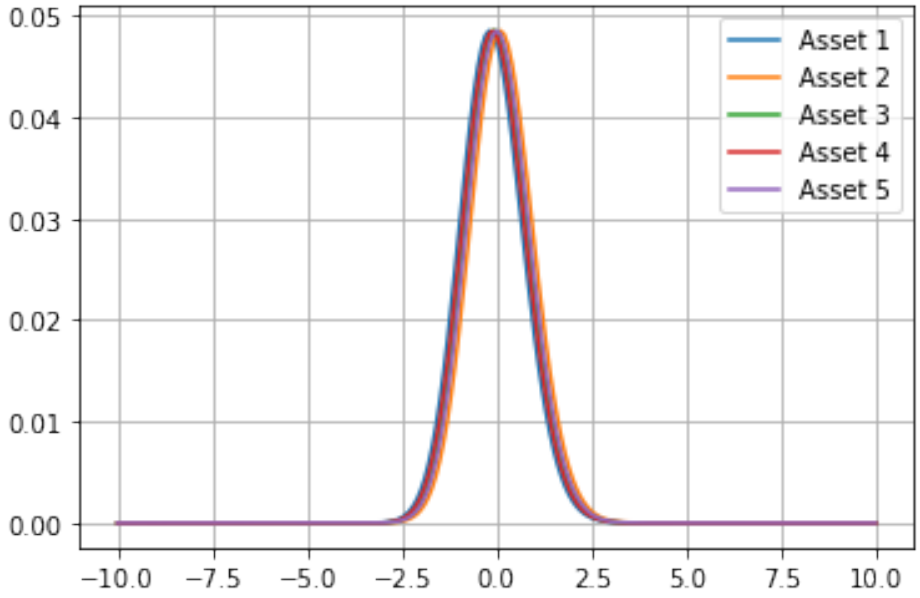

Figure 2: Implied relative return distributions consistent with Table 4.

	Rank 1	Rank 2	Rank 3	Rank 4	Rank 5	Assumed 1	Discrepancy
Asset 1	**0.252**	0.218	0.197	0.178	0.156	**0.250**	0.002
Asset 2	**0.149**	0.176	0.198	0.222	0.255	**0.150**	-0.001
Asset 3	**0.200**	0.203	0.202	0.200	0.195	**0.200**	0.000
Asset 4	**0.220**	0.210	0.201	0.191	0.178	**0.220**	0.000
Asset 5	**0.179**	0.193	0.202	0.209	0.216	**0.180**	-0.001

Table 5: Imputed five-way rank probabilities taking Table 4 as a starting point. A gaussian correlation of $\rho = 0.25$ has been assumed, and this leads to a small differential in Rank 1 probabilities.

from Table 4 and the last column reports the discrepancy after calibration.

The small discrepancy noted arises from two sources. One is due to numerical issues discussed in Cotton (2021a) that, in our example, modify the losing probabilities by roughly one part in a million.

By far the largest component in the last column is due to inconsistency in the calibration approach. Technically speaking, one should calibrate assuming $\rho = 0.25$ whereas we have calibrated using $\rho = 0$ implicitly. However the errors are still quite small, as can be seen, and so a more complex procedure might not be warranted.

5. Summary

The Python winning package provides numerical procedures for estimating the probability that k specific contestants will fill the top k placegettings in a contest among n participants. It also provides a fast calibration procedure whereby location parameters for performance can be inferred from winning probability.

Putting those two together, a fast computational alternative to the Harville (1973) model is provided and then by summation, imputation of rank probabilities from winning (or losing) probability only. Numerous applications are discussed in Cotton (2021a).

```
R> data("quine", package = "MASS")
```

Computational details

If necessary or useful, information about certain computational details such as version numbers, operating systems, or compilers could be included in an unnumbered section. Also, auxiliary packages (say, for visualizations, maps, tables, ...) that are not cited in the main text can be credited here.

The results in this paper were obtained using R 3.4.1 with the **MASS** 7.3.47 package. R itself and all packages used are available from the Comprehensive R Archive Network (CRAN) at https://CRAN.R-project.org/.

Acknowledgments

All acknowledgments (note the AE spelling) should be collected in this unnumbered section before the references. It may contain the usual information about funding and feedback from colleagues/reviewers/etc. Furthermore, information such as relative contributions of the authors may be added here (if any).

References

Bradley RA, Terry ME (1952). "Rank Analysis of Incomplete Block Designs: I. The Method of Paired Comparisons." *Biometrika*. ISSN 00063444. doi:10.2307/2334029.

Cane V, Luce RD (1960). "Individual Choice Behavior: A Theoretical Analysis." *Journal of the Royal Statistical Society. Series A (General)*. ISSN 00359238. doi:10.2307/2343282.

Cotton P (2013). "The horse race problem." URL http://finmathblog.blogspot.com/2013/09/the-horse-race-problem-general-solution.html.

Cotton P (2021a). "Inferring Relative Ability from Winning Probability in Multientrant Contests." *SIAM Journal on Financial Mathematics*, **12**. doi:10.1137/19M1276261.

Cotton P (2021b). "The Python M6 Package." URL https://github.com/microprediction/m6.

Cotton P (2021). "Winning Python Library." URL https://pypi.org/project/winning/.

Harville DA (1973). "Assigning probabilities to the outcomes of multi-entry competitions." *Journal of the American Statistical Association*. ISSN 1537274X. doi:10.1080/01621459.1973.10482425.

Mashaal S (2021). "Horse Package." URL https://github.com/stumash/HorsePackage.

Turner HL, van Etten J, Firth D, Kosmidis I (2020). "Modelling rankings in R: the PlackettLuce package." *Computational Statistics*, **35**(3). ISSN 16139658. doi:10.1007/s00180-020-00959-3.

A. More technical details

Appendices can be included after the bibliography (with a page break). Each section within the appendix should have a proper section title (rather than just *Appendix*).

For more technical style details, please check out JSS's style FAQ at https://www.jstatsoft.org/pages/view/style#frequently-asked-questions which includes the following topics:

- Title vs. sentence case.
- Graphics formatting.
- Naming conventions.
- Turning JSS manuscripts into R package vignettes.
- Trouble shooting.
- Many other potentially helpful details...

B. Using BibTeX

References need to be provided in a BibTeX file (`.bib`). All references should be made with `\cite`, `\citet`, `\citep`, `\citealp` etc. (and never hard-coded). This commands yield different formats of author-year citations and allow to include additional details (e.g., pages, chapters, ...) in brackets. In case you are not familiar with these commands see the JSS style FAQ for details.

Cleaning up BibTeX files is a somewhat tedious task – especially when acquiring the entries automatically from mixed online sources. However, it is important that informations are complete and presented in a consistent style to avoid confusions. JSS requires the following format.

- JSS-specific markup (`\proglang`, `\pkg`, `\code`) should be used in the references.
- Titles should be in title case.
- Journal titles should not be abbreviated and in title case.
- DOIs should be included where available.
- Software should be properly cited as well. For R packages `citation("pkgname")` typically provides a good starting point.

Affiliation:

Peter Cotton
Intech Investments
E-mail: peter.cotton@microprediction.com
URL: https://www.microprediction.org/

Journal of Statistical Software http://www.jstatsoft.org/
published by the Foundation for Open Access Statistics http://www.foastat.org/
MMMMMM YYYY, Volume VV, Issue II *Submitted:* yyyy-mm-dd
doi:10.18637/jss.v000.i00 *Accepted:* yyyy-mm-dd