

# Response to Reviewers

*Eric Hare*

*04/10/2015*

1. It would be best to avoid using `.` in the names of functions except when defining methods. In particular, `make.RV()` is badly named, since “RV” is the class of the main objects created in this package, so this looks like a method for a `make()` function. `RV()` would be a better name. (`as.RV()` is less bad since this is a common paradigm in R and it is unlikely that anyone will define a generic names `as(.)`)  
*We have renamed the `make.RV` function to `RV`, and updated the paper accordingly.*
2. We have renamed the `mult` and `multN` functions to `joint` and `iid` respectively, and updated the paper accordingly. We have also clarified the descriptions of these functions in the table.
3. The `+` and `*` operators are now overloaded. One may call `X + Y` to form the sum of two independent random variables. Similar functionality now exists for joint distributions, where you may call `X * Y` to form the joint of `X` and `Y`. Note that if `X * X` is called, it will simply return the variable obtained by squaring each outcome, and not the joint of two independent realizations of `X`, so that computations like `E(X^2)` still make sense.
4. All random variables used as examples in both the documentation of `discreteRV` and in the paper now begin with a capital letter.
5. We did consider enforcing that random variables have numeric outcomes, as mathematically they must. But we did away with this requirement so that the use cases for `discreteRV` could be expanded to realms where it is easier to refer to factor outcomes. Of course, this greatly reduces the functionality of the package as pointed out. We have updated the paper to clarify that numeric values are preferred for outcomes, not strictly integers.
6. We have attempted to improve the support for random variables with an infinite number of outcomes. This is not perfect in every instance right now, and more work is being done. However, it will handle the cases listed in the paper and be more stable in general.
7. The undesirable inherited properties of vectors pointed out have now been solved by overloaded the `+` and `*` operators as described above.
8. A print method for an `RVsim` object has been added.
9. The arguments to `rsim` have been reversed, as suggested.
10. We agree that storing the outcomes multiple times is an issue. We are striving towards removing this limitation. The `outcomes` attribute is only used in the case of a random variable with infinite support, to store the bounds for printing purposes. The other two are used, one to make syntax more friendly and the other is stored numerically to allow for the precision needed. We have corrected issues with, for instance, `X^2` not updating these outcomes properly, and will work towards a better future solution which isn’t as likely to produce issues.
11. We have added support for three familiar distributions, which are documented in the `RV` function documentation. For instance, to construct a poisson random variable with mean parameter 3, you would call `RV("poisson", lambda = 3)` We will work on adding support for more in the future.
12. Extensive work has been done with regards to joint distributions and the power of `discreteRV` in this realm:
  - A simple interface for defining a joint random variable has been created. Outcomes can be specified as a list of length `n`, where `n` is the number of variables involved. For instance, to create the variable `XandY` used in the review example, one would call `XandY <- jointRV(list(1:3, 0:2), 1:9/sum(1:9))`.
  - Marginal distributions now keep reference to their joint distribution, allowing for the computations explained in the review. To derive the marginals, one would call `X <- marginal(XandY, 1)`, and `Y <- marginal(XandY, 2)`. Then probability statements such as `P(X == 1 | Y > 1)` can be made. Further, `X | (Y > 1)` will return the random variable obtained. The functionality described in the review has been implemented, and it should support more than two random variables, although there are certainly edge cases that likely will need to be addressed in future updates.

13. The statements in the extended example have been revised according to the review. In particular, outcomes are no longer generated using `paste`, but instead by exploiting the new joint random variable functionality.
14. We agree that the simpler notation for the craps game simulation is preferred. We initially had constructed the simulation differently, and did not update the code properly. It has now been updated accordingly.