On Multiple Comparisons in R *

by Frank Bretz, Torsten Hothorn and Peter Westfall

1   Description

The multiplicity problem arises when several inferences are considered simultaneously as a group. If each inference has a 5% error rate, then the error rate over the entire group can be much higher than 5%. This article shows practical examples of multiple comparisons procedures that control the error of making any incorrect inference.

The multcomp package for the R statistical environment allows for multiple comparisons of parameters whose estimates are generally correlated, including comparisons of k groups in general linear models. The package has many common multiple comparison procedures "hard-coded", including Dunnett, Tukey, sequential pairwise contrasts, comparisons with the average, changepoint analysis, Williams', Marcus', McDermott's, and tetrad contrasts. In addition, a free input interface for the contrast matrix allows for more general comparisons.

The comparisons themselves are not restricted to balanced or simple designs. Instead, the package is designed to provide general multiple comparisons, thus allowing for covariates, nested effects, correlated means, likelihood-based estimates, and missing values. For the homoscedastic normal linear models, the functions in the package account for the correlations between test statistics by using the exact multivariate t-distribution. The resulting procedures are therefore more powerful than the Bonferroni and Holm methods; adjusted p-values for

---

*This document is based on an article published in R News

1

these methods are reported for reference. For more general models, the program accounts for correlations using the asymptotic multivariate normal distribution; examples include multiple comparisons based on rank transformations, logistic regression, GEEs, and proportional hazards models. In the asymptotic case, the user must supply the estimates, the asymptotic covariance matrix, and the contrast matrix.

Basically, the package provides two functions. The first, `simint`, computes confidence intervals for the common single-step procedures. This approach is uniformly improved by the second function (`simtest`), which utilizes logical constraints and is closely related to closed testing. However, no confidence intervals are available for the `simtest` function. For testing and validation purposes, some examples from Westfall et al. (1999) are included in the package.

2 Details

Assume the general linear model

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon},$$

where $\mathbf{Y}$ is the $n \times 1$ observation vector, $\mathbf{X}$ is the fixed and known $n \times p$ design matrix, $\boldsymbol{\beta}$ is the fixed and unknown $p \times 1$ parameter vector and $\boldsymbol{\epsilon}$ is the random, unobservable $n \times 1$ error vector, distributed as $N_n(\mathbf{0}, \sigma^2\mathbf{I}_n)$. We assume the usual estimates

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^t\mathbf{X})^-\mathbf{X}^t\mathbf{Y}$$

and

$$\hat{\sigma}^2 = (\mathbf{Y} - \mathbf{X}\hat{\boldsymbol{\beta}})^t(\mathbf{Y} - \mathbf{X}\hat{\boldsymbol{\beta}})/\nu,$$

where $\nu = n - \text{rank}(\mathbf{X})$. Our focus is on multiple comparisons for parameters of the general form $\mathbf{c}^t\boldsymbol{\beta}$. Its variance is given through

$$\text{Var}(\mathbf{c}^t\hat{\boldsymbol{\beta}}) = \hat{\sigma}^2\mathbf{c}^t(\mathbf{X}^t\mathbf{X})^-\mathbf{c}.$$

2

In simultaneous inferences we are faced with a given family of estimable parameters $\{\mathbf{c}_1^t\boldsymbol{\beta}, \ldots, \mathbf{c}_k^t\boldsymbol{\beta}\}$. We thus use the pivotal test statistics

$$T_i = \frac{\mathbf{c}_i^t\hat{\boldsymbol{\beta}} - \mathbf{c}_i^t\boldsymbol{\beta}}{\hat{\sigma}\sqrt{\mathbf{c}_i^t(\mathbf{X}^t\mathbf{X})^-\mathbf{c}_i}}.$$

For a general account on multiple comparison procedures we refer to Hochberg and Tamhane (1987). The joint distribution of $\{T_1, \ldots, T_k\}$ is multivariate t with degrees of freedom ν and correlation matrix $\mathbf{R} = \mathbf{D}\mathbf{C}(\mathbf{X}^t\mathbf{X})^-\mathbf{C}^t\mathbf{D}$, where $\mathbf{C}^t = (c_1, \ldots, c_k)$ and $\mathbf{D} = \mathrm{diag}(\mathbf{c}_i^t(\mathbf{X}^t\mathbf{X})^-\mathbf{c}_i)^{-1/2}$. In the asymptotic case $\nu \to \infty$ or if σ is known, the corresponding limiting multivariate normal distribution holds. The numerical evaluation of the multivariate t and normal distribution is available with the R package `mvtnorm`, see Hothorn et al. (2001).

Note that the contrast matrix C is defined in a slightly different way as usual in S. The `contrasts` function assumes a matrix S defining a set of independent linear combinations of the dummy variables coding the factor levels whereas C defines linear combinations, maybe dependent, of the parameter vector β. The following connection can be established:

$$\begin{aligned} C\hat{\beta} &= CX^+y \\ &= (XC^+)^+y = (XS)^+y. \end{aligned}$$

Therefore, the Moore-Penrose-Inverse C^+ (or any other G-inverse) of the contrast matrices offered by the `contrMat` function can be passed to the `contrasts` function.

The function `simint` provides simultaneous confidence intervals for the estimable functions $\mathbf{c}_i^t\boldsymbol{\beta}$ in the (two-sided) form

$$\left[\mathbf{c}_i^t\hat{\boldsymbol{\beta}} - c_{1-\alpha}\hat{\sigma}\sqrt{\mathbf{c}_i^t(\mathbf{X}^t\mathbf{X})^-\mathbf{c}_i}; \mathbf{c}_i^t\hat{\boldsymbol{\beta}} + c_{1-\alpha}\hat{\sigma}\sqrt{\mathbf{c}_i^t(\mathbf{X}^t\mathbf{X})^-\mathbf{c}_i}\right],$$

where $c_{1-\alpha}$ is the critical value at level $1-\alpha$, as derived under the distributional assumptions above. If lower or upper tailed tests are used, the corresponding interval bounds are set to $-\infty$ and ∞, respectively.

The second function `simtest` provides more powerful test decisions than `simint` yet it does not provide simultaneous confidence intervals. It uses the

stepwise methods of Westfall (1997), which take the logical constraints between the hypotheses into account and which are closely related to the closed testing principle of Marcus et al. (1976). In addition, the stochastic dependencies of the test statistics are incorporated, thus allowing imbalance, covariates and more general models. Again, any collection of linear combinations of the estimable parameters is allowed, not just pairwise comparisons. We refer to Westfall (1997) for the algebraic and algorithmic details.

3 Example

We illustrate some of the capabilities of the `multcomp` package using the `recovery` dataset. Three different heating blankets b_1, b_2, b_3 for post-surgery treatment are compared to a standard blanket b_0. The variable of interest in this simple one-way layout was recovery time in minutes of patients allocated randomly to one of the four treatments. The standard approach for comparing several treatments against a control is the many-to-one test of Dunnett (1955). The Dunnett test is one of the "hard-coded" procedures available for one-factor models in `multcomp`. To obtain simultaneous confidence intervals for the comparisons $\beta_i - \beta_1$ on simply calls:

```
>library(multcomp)

Loading required package: mvtnorm

>data(recovery)
>Dcirec <- simint(minutes ~ blanket, data = recovery,
+     conf.level = 0.9, alternative = "less")

>print(Dcirec)

        Simultaneous confidence intervals: Dunnett
        contrasts


        90 % confidence intervals


                Estimate   --   90 %
```

```
blanketb1-blanketb0   -2.133 -Inf  0.823
blanketb2-blanketb0   -7.467 -Inf -4.510
blanketb3-blanketb0   -1.667 -Inf -0.036
```

Thus, blankets b_2 and b_3 lead to significant lower recovery times in comparison to the standard b_0, since the respective upper confidence bounds are less than 0. In particular, the output above indicates that at the designated confidence level of 90% the average recovery time for b_2 is more than 7 minutes shorter than it is for b_0.

A second way to obtain the same results is to define the contrast matrix $\mathbf{C}$ explicitly:

```
>C <- matrix(c(0, 0, 0, -1, -1, -1, 1, 0, 0, 0,
+     1, 0, 0, 0, 1), nc = 5)
>rownames(C) <- paste("C", 1:nrow(C), sep = "")
>Ccirec <- simint(minutes ~ blanket, data = recovery,
+     conf.level = 0.9, alternative = "less", eps = 1e-04,
+     cmatrix = C)

>print(Ccirec)

        Simultaneous confidence intervals: user-defined
        contrasts


        90 % confidence intervals


   Estimate   --    90 %
C1  -2.1333 -Inf  0.8225
C2  -7.4667 -Inf -4.5108
C3  -1.6667 -Inf -0.0360
```

The first column of $\mathbf{C}$ stands for the intercept β_0, the remaining columns are reserved for the 4 levels $\beta_1, \ldots, \beta_4$ of the single factor. Each row defines a particular linear combination $\mathbf{c}_i^t \boldsymbol{\beta}$. Note that the *eps* argument specifies the accuracy of the numerical results (see pmvt in package mvtnorm for more details). This

5

is the reason why the confidence bounds are now printed with four significant digits instead of the former three digits.

More detailed output is available by using the summary method:

```
>summary(Ccirec)

        Simultaneous 90% confidence intervals: user-defined
        contrasts


        user-defined contrasts for factor blanket


Contrast matrix:
    (Intercept) blanketb0 blanketb1 blanketb2 blanketb3
C1            0        -1         1         0         0
C2            0        -1         0         1         0
C3            0        -1         0         0         1


Absolute Error Tolerance:  1e-04


 90 % quantile:  1.843


Coefficients:
   Estimate    --     90 % t value Std.Err.  p raw p Bonf
C1  -2.1333 -Inf  0.8225 -1.3302    1.6038 0.0958 0.2874
C2  -7.4667 -Inf -4.5108 -4.6556    1.6038 0.0000 0.0001
C3  -1.6667 -Inf -0.0360 -1.8837    0.8848 0.0337 0.1012
    p adj
C1 0.2412
C2 0.0001
C3 0.0924
```

Note that column names have been added to the contrast matrix, i.e. the names of the raw parameters. It is wise to check if the names correspond to the correct column since the columns of the design matrix may have been interchanged. This output prints the user defined contrast matrix $\mathbf{C}$ and the quantile $c_{1-\alpha}$. In addition, simulta-

neous confidence intervals, the estimates $\mathbf{c}_i^t \hat{\boldsymbol{\beta}}$ and their standard errors are given as well as the raw p-values (computed from the marginal t distributions) and multiplicity adjusted p-values (using either the multivariate t distribution or the Bonferroni correction). The simultaneous confidence intervals and the adjusted p-values based on the multivariate t distribution are compatible in the sense that if $p_{adj} < 0.05$, then the associated confidence interval does not contain the 0.

By default, the lexicographically smallest factor level is used as baseline, this can be altered by using the base argument passed to contrMat:

```
>simint(minutes ~ blanket, data = recovery, conf.level = 0.9,
+     alternative = "less", base = 2)

        Simultaneous confidence intervals: Dunnett
        contrasts


Call:
simint.formula(formula = minutes ~ blanket, data = recovery,
    conf.level = 0.9, alternative = "less", base = 2)


        90 % confidence intervals


                    Estimate   --    90 %
blanketb0-blanketb1    2.133 -Inf  4.824
blanketb2-blanketb1   -5.333 -Inf -1.785
blanketb3-blanketb1    0.467 -Inf  3.215
```

A more powerful approach is available using the simtest function. The call remains essentially the same, also no simultaneous confidence intervals are available:

```
>Ctrec <- simtest(minutes ~ blanket, data = recovery,
+     conf.level = 0.9, alternative = "less", eps = 1e-04,
+     cmatrix = C)

>summary(Ctrec)
```

```
         Simultaneous tests: user-defined contrasts


         user-defined contrasts for factor blanket


Contrast matrix:
   (Intercept) blanketb0 blanketb1 blanketb2 blanketb3
C1           0        -1         1         0         0

C2           0        -1         0         1         0

C3           0        -1         0         0         1



Absolute Error Tolerance:  1e-04


Coefficients:
   Estimate t value Std.Err.  p raw p Bonf  p adj
C2  -7.4667 -4.6556   1.6038 0.0000 0.0001 0.0001

C3  -1.6667 -1.8837   1.6038 0.0337 0.0675 0.0640

C1  -2.1333 -1.3302   0.8848 0.0958 0.0958 0.0958
```

It transpires that the adjusted p-values are indeed uniformly lower in comparison to those from `simint`.

A final example call illustrates the use of the `multcomp` package, if the estimates $\hat{\beta}_i$ and their covariances are passed by hand. In such cases, the core functions `csimint` and `csimtest` have to be called without using the `sim{int,test}` interfaces. The call

```
>parm <- c(14.8, 12.6667, 7.3333, 13.1333)
>N <- c(20, 3, 3, 15)
>contrast <- contrMat(N, type = "Dunnett")
>nu <- 37
>mse <- 6.7099
>covm <- mse * diag(1/N)
>csimint(estpar = parm, df = as.integer(nu), covm = covm,
+     cmatrix = contrast, conf.level = 0.9, alternative = "less")
```

```
        Simultaneous confidence intervals: user-defined
        contrasts


        90 % confidence intervals


    Estimate    --    90 %
2-1   -2.133 -Inf  0.822
3-1   -7.467 -Inf -4.511
4-1   -1.667 -Inf -0.036
```

yields the same result as the first call above. The sample size vector N and the mean square error *mse* are only required for a convenient computation of the covariance matrix. Note that the contrast matrix can either be entered by hand or by using the availability of standard contrast matrices in the `contrMat` function.

The generic `simint` implements methods for linear models and generalized linear models. Simultaneous confidence intervals for a subset of parameters of a linear model, given by an integer or character vector argument `psubset` or via a contrast matrix `cmatrix`, are computed as follows.

```
>require(MASS)

Loading required package: MASS
[1] TRUE

>data(anorexia)
>lmod <- lm(Postwt ~ Prewt + Treat, data = anorexia)
>confint(lmod)

                  2.5 %      97.5 %
(Intercept) 23.0498681 76.4923499
Prewt        0.1128268  0.7560955
TreatCont   -7.8754712 -0.3186599
TreatFT      0.3060571  8.8200682

>simint(lmod, psubset = 2:4)
```

```
        Simultaneous confidence intervals: model contrasts


        95 % confidence intervals


          Estimate  2.5 % 97.5 %

Prewt          0.434  0.042  0.827

TreatCont    -4.097 -8.708  0.514

TreatFT       4.563 -0.632  9.758

>simint(lmod, cmatrix = cbind(0, diag(3)))

        Simultaneous confidence intervals: model contrasts


        95 % confidence intervals


   Estimate  2.5 % 97.5 %

C1    0.434  0.042  0.827

C2   -4.097 -8.708  0.514

C3    4.563 -0.632  9.758
```

And for a generalized linear model (using the same example as above):

```
>glmod <- glm(Postwt ~ Prewt + Treat, data = anorexia,

+     family = gaussian)

>confint(glmod)

Waiting for profiling to be done...
                2.5 %      97.5 %

(Intercept) 23.5253133 76.0169047

Prewt        0.1185495  0.7503728

TreatCont   -7.8082428 -0.3858882

TreatFT      0.3818011  8.7443242

>simint(glmod, psubset = 2:4)

        Simultaneous confidence intervals: model contrasts


        95 % confidence intervals
```

```
          Estimate   2.5 % 97.5 %

Prewt        0.434  0.042  0.827

TreatCont   -4.097 -8.708  0.514

TreatFT      4.563 -0.632  9.758
```

Just for illustrative purposes we compute many–to–one confidence intervals by passing the contrast matrix to `lm` directly:

```
>require(MASS)

[1] TRUE

>data(anorexia)

>lmod <- lm(Postwt ~ Prewt + Treat, data = anorexia,

+    contrasts = list(Treat = mginv(contrMat(table(anorexia$Treat)))))

>simint(lmod, psubset = 3:4)

        Simultaneous confidence intervals: model contrasts


        95 % confidence intervals


              Estimate  2.5 % 97.5 %

TreatCont-CBT   -4.097 -8.392  0.198

TreatFT-CBT      4.563 -0.276  9.402

>simint(Postwt ~ Prewt + Treat, data = anorexia)

        Simultaneous confidence intervals: Dunnett
        contrasts


Call:

simint.formula(formula = Postwt ~ Prewt + Treat, data = anorexia)


        95 % confidence intervals


                    Estimate  2.5 % 97.5 %

TreatCont-TreatCBT    -4.097 -8.392  0.198

TreatFT-TreatCBT       4.563 -0.276  9.402
```
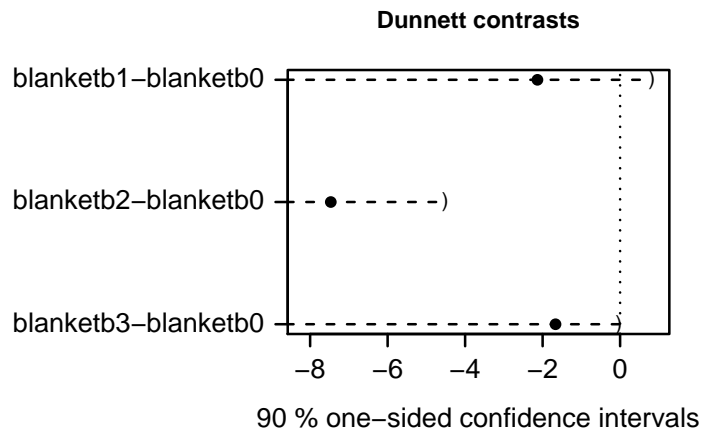
Dunnett contrasts

Figure 1: A graphical representation of one-sided Dunnett confidence intervals. The intervals are plotted as horizontal lines where the limits of the intervals are given by round brackets and the estimates by a point.

4   Graphical Representation

The method `plot.hmtest` is available for a graphical inspection of the simultaneous confidence intervals. For each contrast, the confidence interval is plotted, for example `plot(Dcirec)` can be used for plotting the one-sided Dunnett confidence intervals for the `recovery` example from the first code snippet.

5   Conclusion

This article addressed the application of multiple comparisons using the `mult-comp` package. The present methods cover several standard test procedures and allow for user specified type of comparisons. Also the discussion has been devoted to general linear models, the package is also applicable to more general linear and nonlinear mixed models as long as the covariances between the estimates are known.

Currently, the quantiles of the multivariate t or normal distribution are computed using `uniroot` on the p-value functions. This is time consuming and will be improved in future versions of the `mvtnorm` package.

We would like to thank Doug Bates for corrections and suggestions improving the readability.

References

Y. Hochberg and A. Tamhane. *Multiple comparison procedures*. Wiley, New York, 1987.

T. Hothorn, F. Bretz, and A. Genz. On multivariate t and Gauss probabilities in R. *R News*, 1(2):27–29, 2001.

R. Marcus, E. Peritz, and K.B. Gabriel. On closed testing procedures with special reference to ordered analysis of variance. *Biometrika*, 63:655–660, 1976.

P. Westfall. Multiple testing of general contrasts using logical constraints and correlations. *Journal of the American Statistical Association*, 92:299–306, 1997.

P. H. Westfall, R. D. Tobias, D. Rom, R. D. Wolfinger, and Y. Hochberg. *Multiple Comparisons and Multiple Tests Using the SAS System*. SAS Institute Inc., Cary, NC, 1999.