

# Supplement to “The sensitivity of linear regression coefficients’ confidence limits to the omission of a confounder”

Carrie A. Hosman, Ben B. Hansen, and Paul W. Holland

November 25, 2009

## 1 Introduction

This supplement details how to obtain the data and perform the calculations in the corresponding paper. The .pdf file describes and provides the key pieces of code necessary to obtain the data, prepare the data for analysis, compute benchmarked values of treatment confounding and the partial correlation between omitted variable and the response, and compute the sensitivity intervals. To find all estimates and compute the values in all tables contained in the paper, reference the corresponding .Rnw file. Note that all computations in this supplement were performed with R version 2.8.0. For more information about using the code in .Rnw files with Sweave and R, see a reference such as <http://www.stat.uni-muenchen.de/~leisch/Sweave/> or another similar reference.

## 2 Obtaining and preparing data and first pass analysis

### 2.1 The data

In this section, we show how to obtain and prepare the data. The SUPPORT data analyzed in Connors et al. (1996) can be obtained from the `Hmisc` package. The data is then prepared with some recoding and imputing of missing values as well as computation of the variable `stay` denoting length of ICU stay in the hospital.

```
> library(Hmisc)
> library(boot)
> library(xtable)
> library(ggm)
> sessionInfo()
```

```
R version 2.8.0 (2008-10-20)
i386-pc-mingw32
```

```
locale:
```

```
LC_COLLATE=English_United States.1252;LC_CTYPE=English_United States.1252;LC_MONETARY=English_United States.1252;
```

```
attached base packages:
```

```
[1] tools      stats      graphics  grDevices  utils      datasets  methods
[8] base
```

```
other attached packages:
```

```
[1] ggm_1.0.2    xtable_1.5-5 boot_1.2-37  Hmisc_3.5-2
```

```
loaded via a namespace (and not attached):
```

```
[1] cluster_1.12.0 grid_2.8.0    lattice_0.17-25
```

```
> getHdata(rhc)
```

```
> rhc = data.prep(rhc)
```

## 2.2 First pass regression results and stepwise data reduction

We can fit a simple OLS model to obtain our initial estimate of the effect of RHC on the (log of) the length of hospital stay. With the `step` function, we can reduce the data to the set of 19 covariates discussed in the paper and create a new data frame with this data.

```
> rhcmod = lm(log(stay) ~ swang1 + age + sex + edu + race + income +
+   ninsclas + cat1 + cat21 + ca + cardiohx + chfhx + dementhx +
+   psychhx + chrpulhx + renalhx + liverhx + gibledhx + malighx +
+   immunhx + transhx + amihx + aps1 + scoma1 + meanbp1 + wblc1 +
+   hrt1 + resp1 + temp1 + pafi1 + alb1 + hema1 + bili1 + crea1 +
+   sod1 + pot1 + paco21 + ph1 + wtkilo1 + dnr1 + urin1 + resp +
+   card + neuro + gastr + renal + meta + hema + seps + trauma +
+   ortho, data = rhc)
> mod.reduc = function(rhc, rhcmod) {
+   redmod = step(rhcmod)
+   listt = c(attr(summary(redmod)$terms, "term.labels"))
+   dat = rhc$stay
+   for (i in 1:length(listt)) {
+     a = rhc[which(names(rhc) == listt[i])]
+     dat = cbind(dat, a)
+   }
+   dat$stay = dat$dat
+   dat = as.data.frame(dat[, -1])
+ }
```

```

+     return(dat)
+ }
> dat = mod.reduc(rhc, rhcmod)
> newmod = lm(log(stay) ~ ., data = dat)

```

### 3 Obtaining the necessary quantities for the sensitivity intervals

#### 3.1 Bootstrapping the t-quantile

Rather than using the standard t-quantile for a 95% confidence interval, a bootstrapping procedure provides a more accurate t-quantile. This follows a recommendation in the literature (Faraway, 1992) for when the stepwise reduction procedure may have had an effect on standard errors.

```

> get.tboot2 = function(dat, n) {
+   boot.rhc = function(data, indices) {
+     data = data[indices, ]
+     newmod = lm(log(stay) ~ ., data = data)
+     summary(newmod)$coef[2, 1:2]
+   }
+   bstrap = boot(dat, boot.rhc, n)
+   tboot = rep(NA, n)
+   for (i in 1:n) {
+     tboot[i] = (bstrap$t[i, 1] - bstrap$t0[1])/bstrap$t[i,
+       2]
+   }
+   return(tboot)
+ }
> tb2 = sort(abs(get.tboot2(dat, 5000)))
> t95 = tb2[ceiling(0.95 * 5000)]

```

#### 3.2 Calculating hypothesized values of $t_W$ and $\rho_{y \cdot w|z\mathbf{x}}$

In order to calculate the sensitivity intervals described in the paper, we must first calculate the values of  $t_W$  and  $\rho_{y \cdot w|z\mathbf{x}}$  typical of the variables included in the model of treatment effect. Each variable is omitted in turn and the values are calculated.

The data inputs of this function are the original data, `rhc` ; the reduced data, `dat` ; the index of variable to be omitted; and `type` , the indicator of whether the omitted variable is a numeric or factor variable (We define `type=1` if the variable is a numeric variable and `type=2` if it is a factor variable. Refer to Section 4 of the paper to see the derivation of the code for factor variables). The function outputs a matrix in which each row contains useful values pertaining to each of the variables in the reduced data frame:  $\rho_{y \cdot w|z\mathbf{x}}$ , the value of  $t_W$ , the estimated

regression coefficient for the RHC without the row variable in the model, the corresponding estimated standard error of the regression coefficient of RHC, the degrees of freedom of this regression, and also the estimated regression coefficient for RHC with the variable in question in the model as well as its corresponding standard error.

```
> specpars = function(rhc, dat, var.index, type) {
+   newdat = dat[, -var.index]
+   dat$stay = rhc$stay
+   trtmod = lm(log(stay) ~ ., data = dat)
+   r.with = summary(trtmod)$r.squared
+   b.add = summary(trtmod)$coef[2, 1]
+   se.b.add = summary(trtmod)$coef[2, 2]
+   trtmod2 = lm(log(stay) ~ ., data = newdat)
+   r.without = summary(trtmod2)$r.squared
+   b = summary(trtmod2)$coef[2, 1]
+   se.b = summary(trtmod2)$coef[2, 2]
+   r.par = (r.with - r.without)/(1 - r.without)
+   m = length(dat)
+   newdat = dat[, -m]
+   newdat = newdat[, -j]
+   newdat2 = cbind(newdat, dat[, j])
+   if (type == 1) {
+     bigmod = lm(swang1 ~ ., data = newdat2)
+     tw.v = as.vector(summary(bigmod)$coef[, 3])
+     t.w = tw.v[length(tw.v)]
+     df = summary(bigmod)$df[2]
+   }
+   if (type == 2) {
+     smmod = lm(swang1 ~ ., data = newdat)
+     bigmod = lm(swang1 ~ ., data = newdat2)
+     Fw = ((deviance(smmod) - deviance(bigmod))/(deviance(bigmod)/df.residual(bigmod)))
+     df = summary(bigmod)$df[2]
+     k = length(levels(dat[, j])) - 1
+     t.w = sqrt((k * df/(df + 1 - k)) * Fw)
+   }
+   specpars = cbind(r.par, t.w, b, se.b, df, b.add, se.b.add)
+   return(specpars)
+ }
```

Using this key function, we can obtain the values shown in Table 1, which are used to produce sensitivity intervals in manner similar to the method in the next section.

An unformatted version of Table 1 can be found easily:

```
> xtable(spectab)
```

	tw2	rpar
1	6.80	0.33
2	48.10	3.40
3	20.20	0.77
4	2.10	0.24
5	0.70	0.10
6	5.10	0.14
7	0.50	0.05
8	2.50	0.05
9	2.30	0.14
10	15.40	0.07
11	2.30	0.71
12	3.30	0.89
13	2.20	0.11
14	3.10	0.09
15	6.80	0.25
16	3.70	0.31
17	6.50	3.33
18	5.90	0.21
19	3.60	0.07

## 4 Computation of sensitivity intervals

In this section, we focus on the six variables in Table 3 of the paper. The columns in this table present the absolute value of the estimated confoundedness with treatment ( $t_W$ ) and three sensitivity intervals: the first two use potential values of  $\rho_{y \cdot w | z \mathbf{x}}$  of 1% and 10%, respectively, while the third provides a sensitivity interval when the value of  $\rho_{y \cdot w | z \mathbf{x}}$  is unrestricted (an option in the complete absence of information to help one speculate about the value of  $\rho_{y \cdot w | z \mathbf{x}}$ ). The value of  $t_W$  for each variable is computed in a similar manner to those in Section 3.2 in this supplement.

These methods of calculating the sensitivity intervals would require modification if the value of  $t_W$  is guided by informed speculation rather than calculation with existing data.

With the following functions, sensitivity intervals can be obtained. The main function calls two other functions to make sensitivity intervals depending on the relationship of  $\rho_{y \cdot w | z \mathbf{x}}$  to the value of  $g$  (see Proposition 3.1 of the paper). The function requires the data frame containing the known and measured variables, a benchmarked value of treatment confounding (a hypothesized  $t_W$ ), a t-quantile, a value of  $\rho_{y \cdot w | z \mathbf{x}}$  under consideration (such as 1% or 10% or if unrestricted, simply use a value of 1), and a variable degrees of freedom value (1 for numeric covariates or  $l - 1$  where  $l$  is the number of levels for a factor variable).

```
> make.ci = function(dat, tw, tquant, r.par, k) {
+   mod = lm(log(stay) ~ ., data = dat)
```

```

+   df = summary(mod)$df[2]
+   g = (tw^2 * (df - k))/(tw^2 * (df - k) + tquant^2 * (tw^2 +
+     df))
+   if (r.par <= g) {
+     ci = cor2(dat, tw, tquant, r.par, k)
+   }
+   if (r.par > g) {
+     ci = cor1(dat, tw, tquant, k)
+   }
+   return(ci)
+ }
> cor2 = function(dat, tw, tquant, r.par, k) {
+   mod = lm(log(stay) ~ ., data = dat)
+   b = summary(mod)$coef[2, 1]
+   se.b = summary(mod)$coef[2, 2]
+   T = abs(tw)
+   df = summary(mod)$df[2] - 1
+   adj1 = T * sqrt(r.par)
+   adj2 = tquant * sqrt(((T^2) + df - k + 1)/(df - k)) * sqrt(1 -
+     r.par)
+   adj = adj1 + adj2
+   lb = b - adj * se.b
+   ub = b + adj * se.b
+   ci = cbind(lb, ub)
+   return(ci)
+ }
> cor1 = function(dat, tw, tquant, k) {
+   mod = lm(log(stay) ~ ., data = dat)
+   b = summary(mod)$coef[2, 1]
+   se.b = summary(mod)$coef[2, 2]
+   T = abs(tw)
+   df = summary(mod)$df[2] - 1
+   adj = sqrt((T^2) + (tquant^2) * (((T^2) + df - k + 1)/(df -
+     k)))
+   lb = b - adj * se.b
+   ub = b + adj * se.b
+   ci = cbind(lb, ub)
+   return(ci)
+ }

```

Then, a sensitivity interval for weight and a hypothesized value of  $\rho_{y \cdot w|z\mathbf{x}}$  of 10% can be computed:

```
> kiloci = make.ci(dat, as.numeric(spec[4, 1]), t95, 0.1, 1)
```

Sensitivity intervals for the other variables and other values of  $\rho_{y \cdot w|z\mathbf{x}}$  can be computed similarly.

The last column of Table 3 in the paper allows  $\rho_{y \cdot w|z\mathbf{x}}$  to be unrestricted. In this case, we use a value of  $\rho_{y \cdot w|z\mathbf{x}} = 1$ . The intervals can be computed in precisely the same manner.

## 5 Extensions

In Section 3.2, the variation of the computation of the values of  $t_W$  and  $\rho_{y \cdot w|z\mathbf{x}}$  when the omitted variable is a factor variable is presented. In this section, we provide the code to compute intervals using the extensions of our method presented in Sections 4.2 and 4.3 in the paper. These extensions examine treatment effects differing by subgroups and sensitivity intervals in combination with propensity score subclassification.

### 5.1 Treatment effects differing by subgroup

This section relies on different methods of benchmarking the treatment confounding to arrive at the values of  $t_W$ . The sensitivity intervals are also based on slightly different model fits. For details, refer to Section 4.2 of the original paper.

### 5.2 Sensitivity intervals with propensity scores

The code provided in this section is very similar to that in Section 4, but it accomodates a propensity score. It requires specification of a propensity score and also requires a choice of  $n$ , which is the number of strata into which the data will be divided on the basis of the estimated propensity score. One could use the rule of thumb of  $n = 5$ , but some experimentation could be done with balance assessed on various choices of strata. We chose six strata because it was the smallest number of strata with no significant imbalance on the covariates included in the propensity model.

```
> pmod = glm(swang1 ~ age + sex + race + ninsclas + income + cat1 +
+   cat21 + ca + edu + cardiohx + chfhx + dementhx + psychhx +
+   chrpulhx + renalhx + liverhx + giblethx + malighx + immunhx +
+   transhx + amihx + aps1 + scoma1 + meanbp1 + wblc1 + hrt1 +
+   resp1 + temp1 + pafi1 + alb1 + hema1 + bili1 + crea1 + sod1 +
+   pot1 + paco21 + ph1 + wtkilo1 + dnr1 + resp + card + neuro +
+   gastr + renal + meta + hema + seps + trauma + ortho + urin1,
+   family = "binomial", data = rhc)
> pcores = pmod$linear.predictor
```

## References

CONNORS, A. J., SPEROFF, T., DAWSON, N., THOMAS, C., HARRELL, F. E. J., WAGNER, D., DESBIENS, N., GOLDMAN, L., WU, A., CALIFF, R.,

- FULKERSON, W. J., VIDAILLET, H., BROSTE, S., BELLAMY, P., LYNN, J. and KNAUS, W. (1996). The effectiveness of right heart catheterization in the initial care of critically ill patients. support investigators. *J. Amer. Med. Assoc.*, **276** 889–97.
- FARAWAY, J. J. (1992). On the cost of data analysis. *Journal of Computational and Graphical Statistics*, **1** 213–229.