G-computation for causal inference

Part 1 : Understanding G-computation

1) Data preparation :
   a) Load the database dataCOHORT (*https://github.com/chupverse/gcomputation/*)

```
library(gcomputation)
```

```
data(dataCOHORT)
```

   b) Check the positivity assumption[1].
      For the remainder of the practical assignment, we will exclude patients under the age of 30.
   c) Choose the prognostic factors to adjust for in the G-computation model using the DAG.
      For the remainder of the practical assignment, we will choose to adjust on the age, the BMI, the gender, the Glasgow Coma Scale, the injury and the PAO2/FIO2.

2) Creating the outcome model[2] :
   a) Define numeric variables for the outcome and exposure.

```
dataCOHORT$VAP_num <- ifelse(dataCOHORT$VAP == "Yes", 1, 0)
```

```
dataCOHORT$GROUP_num <- ifelse(dataCOHORT$GROUP == "Untreated", 0, 1)
```

   b) Construct the outcome model using a multivariate logistic regression on complete cases.

3) Calculate the counterfactual predictions :
   a) Create two counterfactual datasets, `data0` and `data1` where all subjects are set as untreated and treated respectively.
   b) Use the outcome model to predict the individual probability of VAP for every subject in `data0` and in `data1`.

4) Estimate the marginal proportions and ATE :
   a) Calculate the estimated marginal proportion of VAP (`P0` and `P1`) in `data0` and `data1`, respectively, by taking the mean of the predicted probabilities in each dataset.
   b) Calculate the ATE as the risk difference : `ATE = P1 – P0`

5) Interpret the resulting ATE.

Part 2 : Bootstrapping for variance estimation

6) Estimate the uncertainty of the ATE using bootstrapping[3] :
   a) Write a function that performs steps 2 to 4 from Part 1 and returns the estimated ATE.
   b) Run a bootstrap loop (e.g., 100 iterations), resampling the data with replacement at each iteration to compute and store the ATE.
   c) Calculate the mean ATE and the 95% confidence interval using the 2.5th and 97.5th percentiles and interpret the results.

Part 3 : Using the gcomputation package

From here on we will use `gc_binary` function of the `gcomputation` package, which combines these steps, allows for different outcome models and includes bootstrapping. See Table 1 at the end of the document for a description of the available parameters.

7) Running the `gc_binary` function :
   a) Define a formula `.f1` for the binary outcome `VAP_num`, adjusted for the `GROUP_num` and the variables selected in step 1)c).
   b) Run `gc_binary` using the "all" model, matching the manual steps from Part 1.

```
gc_bin_all <- gc_binary(formula = .f1, data = dataCOHORT,
                        group = "GROUP_num", model = "all",
                        boot.number = 100, effect = "ATE",
                        progress = TRUE, seed = 5186)
```

8) Display and interpret the results :
   a) Print the results of `gc_bin_all`.
   b) Use summary to obtain the detailed output, including confidence intervals derived from the bootstrap results with the `ci.type = “perc”` parameter.
   c) Compare results with the manual bootstrap results from Part 2.

9) Verifying the model integrity :
   a) Create a subpopulation including only patients with `LEUKO ≥ 20000`
   b) Define the following two formulas and run the `gc_binary` function on each formula with the “all” model :

```
.f2 <- VAP_num ~ GROUP_num * DIABETES + PAO2FIO2 +
                 GLASGOW + TIME_INTUBATION
.f3 <- VAP_num ~ GROUP_num + AGE + BMI + GLASGOW + INJURY + PAO2FIO2
```

   c) Plot each of the models and compare the two calibration plots by plotting the object returned by the `gc_binary` function.

Bonus questions

10) Calculate the marginal effect of the group on the death (censored outcome) using the `gc_times` function.

11) Run the `gc_binary` function of the following formula with the “all” model and compare the results obtained with and without multiple imputation[4] (`boot.mi = TRUE`).

```
.f4 <- VAP_num ~ GROUP_num * (AGE + SEX + BMI + DIABETES + ALCOOL +
                              SMOKING + INJURY + GLASGOW + PAO2FIO2 +
                              LEUKO + TIME_INTUBATION)
```

12) Run the `gc_binary` function with the `.f4` formula defined previously with the “lasso” model and compare the results obtained with the “all” model.[5,]

Table 1 : Arguments of the `gc_binary` function.

Argument	**Description**
formula	A regression formula related to the Q-model, with the variable group among the predictors.
data	A data frame in which to look for the variables related to the outcome, the studied exposure (group), and the predictors included in the model.
group	The name of the variable related to the exposure/treatment. This variable must have two modalities, encoded as 0 for untreated/unexposed patients and 1 for treated/exposed ones.
effect	The type of marginal effect to be estimated. Three types are possible: "ATE" (default), "ATT", and "ATU".
model	The modelling method used to create the Q-model. Implemented methods are: "all", "lasso", "ridge", "elasticnet", "aic", and "bic".
param.tune	Optional argument to specify the tuning parameters for the Q-model. If NULL (default), the tuning parameters are estimated by cv-fold cross-validation. Otherwise, the user can provide a tuning grid or specific values for each method.
cv	The number of splits for cross-validation. Default is 10.
boot.type	The type of bootstrap to use. Two types are available: "bcv" (default) and "boot".
boot.number	The number of bootstrap resamples. Default is 500.
boot.tune	Logical value indicating whether tuning parameters should be estimated within each bootstrap iteration.
boot.mi	Logical value indicating whether multiple imputation should be applied using the mice package before G-computation.
progress	Logical value indicating whether to print a progress bar in the R console. Default is TRUE.
seed	Random seed to ensure reproducibility during cross-validation. If NULL, a seed is randomly assigned.
m	The number of multiple imputations to perform if boot.mi = TRUE.
...	Additional arguments to be passed to the mice function for customizing the multiple imputation process (e.g., method, maxit, diagnostics).

References

(1) Léger, M.; Chatton, A.; Borgne, F. L.; Pirracchio, R.; Lasocki, S.; Foucher, Y. Causal Inference in Case of Near-Violation of Positivity: Comparison of Methods. *Biometrical Journal* **2022**, *64* (8), 1389–1403. https://doi.org/10.1002/bimj.202000323.

(2) Snowden JM, Rose S, Mortimer KM. Implementation of G-computation on a simulated data set: demonstration of a causal inference technique. American Journal of Epidemiology. 2011;173(7):731–738. doi: 10.1093/aje/kwq472.

(3) Giordani, P.; Kiers, H. A. L. Bootstrap Confidence Intervals for Principal Covariates Regression. Br J Math Stat Psychol 2021, 74 (3), 541–566. https://doi.org/10.1111/bmsp.12238.

(4) Schomaker, M.; Heumann, C. Bootstrap Inference When Using Multiple Imputation. Stat Med 2018, 37 (14), 2252–2266. https://doi.org/10.1002/sim.7654.

(5) Chatton, A.; Le Borgne, F.; Leyrat, C.; Gillaizeau, F.; Rousseau, C.; Barbin, L.; Laplaud, D.; Léger, M.; Giraudeau, B.; Foucher, Y. G-Computation, Propensity Score-Based Methods, and Targeted Maximum Likelihood Estimator for Causal Inference with Different Covariates Sets: A Comparative Simulation Study. *Sci Rep* **2020**, *10* (1), 9219. https://doi.org/10.1038/s41598-020-65917-x.

(6) Shortreed, S. M.; Ertefaie, A. Outcome-Adaptive Lasso: Variable Selection for Causal Inference. *Biometrics* **2017**, *73* (4), 1111–1122. https://doi.org/10.1111/biom.12679.