

Simulationen

Propädeutikum im Sommersemester 2019

Eva Endres

Institut für Statistik, LMU

03.05.2019

Statistische Analyseverfahren beruhen in der Regel auf einer Annahme darüber, wie die zu analysierenden Daten entstanden sind.

Beispiele: $z \stackrel{iid}{\sim} \text{Po}(\lambda)$ oder $y_i \sim \text{N}(\mathbf{x}_i' \boldsymbol{\beta}, \sigma_\epsilon^2)$

Ziel: Etwas über die durch das Modell beschriebene Verteilung zu lernen:

- Verteilungsparameter
- Strukturparameter

Frage-/Zielstellungen:

- Unter welchen Datensituationen funktioniert ein Modell gut?
- Bei einer gegebenen Datensituation, welches ist das *optimale* Modell?
- Wie gravierend sind Verletzungen von Annahmen?

Verhalten von Modellen oft analytisch nicht berechenbar

⇒ Überprüfung der Fragestellung anhand von Simulationen

Beispielhafter Aufbau einer Simulation

- 1 Datengenerierender Prozess: (Verteilungs-)Modell zur Datenerzeugung
- 2 Statistische(s) Verfahren zur Untersuchung
- 3 Performance-Kriterium/Kriterien

Variation von Elementen aus 1. und 2. ergibt m Szenarien

Pro Szenario:

Erzeugung von k Datensätzen und Evaluierung der/des statistischen Verfahren darauf

Konkrete Wahl von k und m situationsabhängig

- Was ist als fest anzunehmen (Verfahren/Parameter/Daten)?
- Was soll/muss flexibel sein?
- Welche Strategie verwendet man für die flexiblen Parameter?

Durchführungsplan sollte vor dem Start der Simulation feststehen!

- Nicht zu viele Parameter!
Exponentielle Explosion der Komplexität
- Fairer Vergleich von mehreren Verfahren auf gleicher Datenbasis
- Umsetzung in Software: Code in Funktionen statt viel Copy-Paste;
bei Fehler(n) oder nachträglichen Änderungen muss man nur an einer Stelle etwas ausbessern
- Beim Ziehen von Zufallszahlen vorher Seed setzen
- Bei langen Simulationen: Zwischenspeichern von (Teil-)Ergebnissen;
Vorab Funktionen testen!

Kleiber, C. und Zeileis, A. (2013). Reproducible econometric simulations, *Journal of Econometric Methods* 2(1): 89–99, [[DOI](#)]

Beispielhafte Simulation in R mit Hervorhebung der folgenden Aspekte:

- Modularisierung
- Nachvollziehbarkeit
- Wiederverwendbarkeit von Code