

ReadMe – HAR Modelling

Az egész alapvetően úgy működik, hogy a configHAR függvénnyel létre kell hozni egy config struktúrát, ami tartalmazza, hogy pontosan mit is szeretnénk futtatni és milyen beállításokkal. Ezután már csak át kell adni a fitHAR függvénynek az adatokat és a config-ot, amit pedig elvégzi a szükséges számításokat. Az egészben az a jó, hogy ha végzett egy futtatás, akkor elég csak 1 sort átírni a config-ban és máris lehet újra futtatni az új beállításokkal.

configHAR function: itt lehet megadni a beállításokat. Teleraktam olyan függvényekkel, ami lecsekkolja, hogy valóban a várt inputot kapja e.

- Config: egy előző config beállítás, ez akkor érdekes, ha csak egy előzőt szeretnénk módosítani
- Type: in sample vagy out of sample
- Models: a modellek listája (pl.: [„HAR”, „HAR_RS”]), olyanokat lehet megadni, amik az estimateHAR függvényben felsorolva vannak, megadható string vektorként és cell formátumban is
- ModelNames: milyen néven mentse a modelleket (pl ha a HAR modell log formáját szeretnénk becsülni, akkor a Models-be a „HAR”-t kell megadni és a Transform-ban a „log”-ot, viszont elmenthetjük „logHAR” néven, hogy ne keveredjen össze az eredetivel)
- h: előrejelzési horizon (egyszerre több is megadható)
- k: lag struktúra (1 5 22)
- T: ha nem az egész időtartamra szeretnénk, hanem csak egy szeletére
- w: window size
- Overlap: legyen e a változók között overlap amikor számolja
- Transform: itt adható meg hogy „log”, „sqrt” (realized volatility), „logsqrt”
- TransformType: itt adható meg hogy egyes változókat, hogy transzformáljon különleges módon (pl a 0-t tartalmazó változóknál nem lehet logaritmust venni)
- TransformSum: ez egy apróság, hogy amikor több időszakra számol realizált volatilitás átlagot akkor először összegezzem és aztán vonjon gyököt vagy fordítva (jensen egyenlőtlenség miatt különbözik)
- NormVar: szintén a transzformációhoz tartozik, ha olyan módon szeretnénk pl, hogy $\log(1+x/RV)$
- Intercept: legyen-e konstans a modellekben
- Display: a modellek futtatása közben kiírja e hogy hol jár

fitHAR function: ez pedig az a függvény ami valójában a modell futtatásokat végzi. Bekéri az adatokat (struktúrában ahol minden változó hossza egyforma) illetve az configot.

createVar, lagVar, leadVar: ezek segédfüggvények, amik arra szolgálnak, hogy a regresszióhoz elkészítsék a megfelelő változókat (leadVar a függő változót, lagVar a magyarázóváltozókat). A kódban megtalálhatók azok a transzformáció típusok is amik a TransformType-ban megadhatók (‘+’, ‘+X’, stb)

Ha ezek elkészültek akkor a fitHAR függvény végig megy az összes megadott előre jelzési horizonton és modellen és annak függvényében hogy in sample vagy out of sample nézzük, visszac dob egy struktúrában megfelelő adatokat:

- in sample output: maga a modell, a newey west t statisztikák és p értékek
- out of sample output: az előrejelzett érték (f_{hat}), a realizált érték (f), az SSE és a szabadságfok (ami a log vissza transzformációhoz kell), és a becsült koefficiensek a különböző ablakokon (coeff)

estimateHARX function: ez a függvény, ami magát az OLS becslést végzi. A felsorolásban ott vannak, hogy milyen modellek adhatók meg, de megadható egyedi modell így, pl: ‘RV~RV1+J1’. Ha esetleg negatív RV-t jelezne előre akkor ahelyett egy átlagot dob vissza (ezt lehet, hogy átállítom inkább majd a legutolsó értékre).

Ha ezek lefutottak akkor visszkapunk egy struktúrát az eredményekkel, amiből szeretnénk valamilyen eredményeket kiolvasni:

getEstimates function: ez a függvény visszadobja az in sample elemzésből a koefficienseket, t és p értékeket, valamint az R2 és adjR2 értékeket rendezettebb formában. a 'h' és 'Models' paraméterekkel lehet szűrni a modelleket, ha nem akarjuk mindet visszkapni.

evalForecast function: ezzel pedig a különböző out of sample előre jelzéseket lehet összehasonlítani különböző módszerekkel. Itt is lehet szűrni előre jelzési horizontra (h) és modellekre (Models), valamint be lehet állítani, hogy QLIKE vagy MSE loss function legyen, illetve hogy a test statisztikákat vagy a p értékeket dobja vissza (Pval), illetve a MCS eljárásnak is lehet a paramétereit állítani (MCSp, MCSiter).

- loss: simán az egyes előre jelzésekre vonatkozó loss értékeket dobja vissza minden időpontra
- meanloss: az loss értékek átlaga az egyes modellekre különböző h értéknél
- MZ: a modellek Mincer Zarnowitz R2 mutatója
- DM: Diebold Mariano teszt az összes modell közt az összes időszakra
- DMpair: DM teszt, de csak 2 modell közt az összes időszakra
- GW: Giacomini White teszt az összes modell közt az összes időszakra
- GWpair: GW teszt, de csak 2 modell közt az összes időszakra
- MCS: Model Confidence Set számolása az összes időszakra

Illetve van még néhány segédfüggvény:

backTransform function: ez csinálja azt, hogy a log modellből származó előre jelzéseket visszatranszformálja $\exp(\mu + \sigma^2/2)$ módon

selectFit: ez csak azt csinálja, hogy lehet szűrni bizonyos modellekre és előre jelzési horizontokra

mergeFit: ez két eredményeket tartalmazó struktúrát von össze egybe