Datorlaboration 4

Måns Magnusson

VT 2014

Instruktioner

- **Allmänt**
  Vid tidigare laborationer har vi använt SAS för att dra urval, studier av teoretiska egenskaper hos estimatorer och allokera urval. Nu ska vi fokusera på att med R göra analyser, bortfallhantering och estimation då vi fått in data från en surveyundersökning.
- **Datamaterial**
  Vilket datamaterial som ska användas framgår av respektive uppgift. Allt datamaterial finns att tillgå **här** om inte annat anges. För att ladda ned datan, klicka på den datafil du vill ladda ned och klicka sedan på "Raw" med högra musknappen och klicka "Spara länk som...".
- **Hjälpmaterial**
  Behöver ni hjälp med att använda R-paketet survey finns, utöver dokumentationen, extra material här. Det finns också en bok *Complex surveys : a guide to analysis using R* som behandlar analyser med surveypaketet i R.
- Det är tillåtet att diskutera med andra men att plagiera andra grupper är **inte tillåtet**.
- Utgå från mallen för laborationsrapporter som går att ladda ned som **LyX** eller **PDF**. Samtliga labbrapporter ska lämnas in **i PDF-format via LISAM**.
- Godkänd laboration ger **0.5 p** på tentamen. Extrapoängen är endast giltiga vid denna kursomgång.
- Deadline för labben framgår på **kurshemsidan**.
- **Laborationsrapport**
  Rapporten ska innehålla den kod ni kört, eventuella resultat samt svara på de frågor som finns i laborationen.

Innehåll

1 Förberedelser

1.1 Läsa in `survey` paketet

- Börja med att läsa in **`survey`**-paketet

```
library(survey)
```

Om det inte går att läsa in paketet (ex. du har en egen dator) behöver du installera paketet först. Det kräver internetanslutning och då använder du följande kod.

```
install.packages("survey")
https://www.skatteverket.se/download/18.353fa3f313ec5f91b951151/1370005553373/Skv_medborgarunders_20
```

1.2 Ladda ned och läs in `agpop.dat`

- Ladda ned filen (se instruktionen). Identifiera den mapp där du har sparat filen **`agpop.dat`**. Använd funktionen **`setwd()`** för att ställa in den korrekta sökvägen .
- Läs in **`agpop.dat`** i R, vilket kan göras med följande kod:

```
agpop <- read.table("agpop.dat", header = TRUE, sep = ",")
```

- Glöm inte bort att ta bort bortfallet för variablerna ACRES92 och ACRES87 i **`agpop`**. För att ta bort saknade värden kan följande kod användas:

```
agpop <- agpop[agpop$ACRES92 > 0 & agpop$ACRES87 > 0, ]
```

2 Obundet slumpmässigt urval

a) Vi ska börja med att använda oss av R för att dra ett OSU från populationen **`agpop`** som vi läst in. För detta använder vi funktionen **`sample()`** (det finns mer avancerade metoder för att dra urval i R-paketet **`sampling`**). Med hjälp av denna funktion kan vi dra ett slumpmässigt urval, antingen med eller utan återläggning (vi kommer bara dra utan återläggning i denna laboration). Detta görs genom att först dra ett antal index som vi sedan använder för att välja ut våra urvalsenheter från populationen.

Exempel:

```
OSUindex <- sample(size = 300, 1:3042, replace = FALSE)
agOSUdata <- agpop[OSUindex, ]
```

En av de stora skillnaderna (och fördelarna) med R jämfört med andra statistikprogram är att R är objektorienterat. Vi kommer därför att använda vårt urval för att skapa ett surveyobjekt. Detta objekt innehåller sedan all information som R behöver för att genomföra de populationsskattningar vi är intresserade av.

För att skapa ett surveyobjekt behöver vi dels en **`data.frame`** (vårt urval) och information om populationsstorleken (för att beakta ändlighetskorrektionen i variansberäkningarna). Det som krävs är en vektor med populationstotalen för respektive urvalsenhet. Det kan tyckas konstigt att ändlighetskorrektionen anges på detta sätt, men när vi sedan studerar stratifierade urval blir det (förhoppningsvis) tydligare.

Exempel:

```
fpc.srs <- rep(3042, 300)
```

Med denna vektor med N kan vi nu skapa vårt första surveyobjekt. För detta använder vi funktionen **`svydesign()`**.

Exempel:

```
agOSU <- svydesign(ids = ~1, data = agOSUdata, fpc = fpc.srs)
```

Det argument som funktionen behöver är `ids` (som indikerar kluster och eftersom vi inte har kluster i denna design anges bara `~1`), sedan anges vår urvalsfil under `data` och sist anges vektorn med populationsstorleken per urvalselement för i argumentet `fpc` (finite population correction).

Vi kan nu använda oss av funktionen `summary()` för att få mer information om vårt surveyobjekt. Vilken information får du om surveyobjektet `agOSU`? Är det någon information du saknar?

b) Med hjälp av surveyobjektet är det (relativt) enkelt att skatta populationsmedelvärdet och populationstotalen. Detta görs med funktionerna `svymean()` och `svytotal()`. För att välja ut vilka variabler i surveyobjektet som ska användas för skattningar används tecknet `~`. Utöver vilken variabel som är intressant behöver vi också ange vårt surveyobjekt som vi vill använda för skattningen av `ACRES92`.

Exempel:

```
svymean(~ACRES92, design = agOSU)

          mean    SE
ACRES92 331502 23799

svytotal(~ACRES92, design = agOSU)

             total       SE
ACRES92 1008429317 72396151
```

Som du kanske märkt beräknar inte funktionen automatiskt ett konfidensintervall utan bara estimatet med tillhörande medelfel. För att beräkna konfidensintervall används funktionen `confint()` och funktionen ska användas på ett `svystat`-objekt (som exempelvis skapas av `svymean()` och `svytotal()`. Det gör att vi behöver spara ned vår skattning innan vi beräknar konfidensintervallet.

Exempel:

```
medel <- svymean(~ACRES92, design = agOSU)
confint(medel)

         2.5 % 97.5 %
ACRES92 284857 378147
```

Vad får du för konfidensintervall? Täcker intervallet det sanna värdet i populationen du dragit ditt urval från?

2.1 Estimation i redovisningsgrupper

a) Funktionerna `svymean()` och `svytotal()` skattar två vanliga estimat av intresse. Ofta finns också ett intresse av att skatta dessa storheter i olika **redovisningsgrupper**. Detta görs med funktionen `svyby()` och för att använda denna funktion behöver du dels ange vilken variabel du är intresserad av att skatta i respektive redovisningsgrupp och sedan vilken variabel som indikerar redovisningsgrupperna (med argumentet `by`), vi behöver också ange vilken skattning vi vill göra (med argumentet `FUN`) och vilket surveyobjekt vi vill använda (med argumentet `design`).

Vi ska nu skatta medelvärdet (med `svymean()`) för variabeln `ACRES92` i respektive region.

Exempel:

```
svyby(~ACRES92, by = ~REGION, design = agOSU, FUN = svymean)

   REGION ACRES92     se
NC     NC  367148  37379
NE     NE  122769  14845
S       S  237869  24402
W       W  611597 102230
```

Är det någon skillnad mellan de olika regionerna när det gäller ytan som används för jordbruk? Vilken region har störst yta för jordbruk? Varför?

b) Ett alterntativ för att producera estimat i enskilda grupper är att använda funktionen `subset()`. Den kan användas för att plocka ut enskilda observationer från ett surveyobjekt och skapa ett nytt surveyobjekt. Detta kan sedan användas precis som vanligt för estimation.

Exempel:

```
agOSUne <- subset(agOSU, REGION == "NE")
svymean(~ACRES92, design = agOSUne)

          mean    SE
ACRES92 122769 14845
```

2.2 Estimation av design effect

Vi kan också vara intresserade av vilken designeffekt vi har för en enskild skattning. För att beräkna designeffekten lägger vi bara till argumentet `deff` i den estimator vi vill använda.

Exempel:

```
svymean(~ACRES92, design = agOSU, deff = TRUE)
```

Vad får du för design effect? Varför får du detta resultat?

2.3 Kvotestimation

Vi vet sedan tidigare att det går att använda hjälpinformation för att öka precisionen i våra skattningar. Med hjälp av kvotestimation kan vi ta hjälp av en hjälpvariabel för att skatta variabeln `ACRES92` med bättre precision. Vi gör detta i R i två steg (precis som om vi skulle göra beräkningen för hand). Som ett första steg först skattar vi kvoten med funktionen `svyratio()`.

För att använda kvotskattningen tillsammans med vår kända populationstotal använder vi sedan funktionen `predict()`. Denna funktion använder vår kvotskattning tillsammans med en total vi känner för hela populationen.

Exempel:

```
ratio.ACRES92 <- svyratio(numerator = ~ACRES92, denominator = ~ACRES87, design = agOSU)
predict(object = ratio.ACRES92, total = sum(agpop$ACRES87))

$total
          ACRES87
ACRES92 947275128

$se
        ACRES87
ACRES92 6902420
```

Vad får du för resultat om du använder en kvotskattning och varför får du detta resultat? Hur mycket tjänar vi på att använda en kvotskattning i detta exempel? Hur skulle du göra om du skulle vilja skatta medelvärdet med hjälp av kvotestimatorn?

3 Stratifierat urval

a) Att dra ett stratifierat urval i R är inte mycket mer komplicerat än att dra ett urval med OSU. För att dra ett stratifierat urval använder vi funktionen `stratsample()` i `survey`paketet. Denna funktion skapar på liknande sätt som `sample()` en vektor av index som sedan kan användas för att välja ut urvalsenheterna från populationen `agpop`. Dock kräver denna funktion att en stratifieringsvariabel anges och att urvalsstorleken anges för respektive strata.

Vi utgår från tidigare datorlaborationer och använder oss av Neymanallokering för att allokera urvalet mellan strata.

Exempel:

```
STRATAindex <- stratsample(agpop$REGION, c(NC = 69, NE = 7, S = 122, W = 102))
agSTRATAdata <- agpop[STRATAindex, ]
```

Precis som i fallet med OSU behöver vi skapa en vektor med populationsstorleken (`fpc`) för att möjliggöra ändlighetskorrektion av varianserna. Nu avser `fpc` istället populationstotalerna för respektive strata.

Exempel:

```
fpc.strata <- numeric(300)
fpc.strata[agSTRATAdata$REGION == "NC"] <- 1049
fpc.strata[agSTRATAdata$REGION == "NE"] <- 209
fpc.strata[agSTRATAdata$REGION == "S"] <- 1370
fpc.strata[agSTRATAdata$REGION == "W"] <- 414
```

Nu har vi den information som behövs för att kunna skapa ett surveyobjekt baserat på ett stratifierat urval. Att skapa ett surveyobjekt för ett stratifierat urval kräver utöver de argument som behövdes i OSU-fallet även att argumentet `strata` anges där variabel som använts för stratifiering anges.

Exempel:

```
agSTRAT <- svydesign(~1, strata = ~REGION, fpc = fpc.strata, data = agSTRATAdata)
```

Använd funktionen `summary()` för att få information om det stratifierade surveyobjektet. Vilka skillnader finns mot surveyobjektet då designen var OSU? Är det någon information du saknar?

b) I den tidigare uppgiften där vi hade ett fall med vanligt OSU är vikterna av mindre intresse (eftersom alla vikter $w_i = \frac{N}{n}$). Nu har inte längre samtliga element samma designvikt. I flera fall kan det vara så att vi ska leverera ett dataset med tillhörande vikter. Vi kan då behöva plocka ut vikterna från ett surveyobjekt. Detta gör vi med funktionen `weights()`. Vi behöver dock ange vilken typ av vikter vi är intresserade av. Just nu är vi bara intresserade av att plocka ut designvikterna och då anger vi argumentet `type = ‘‘sampling’’` i funktionen `weights()`.

Exempel:

```
weights(agSTRAT, "sampling")
```

Hur stor är den största vikten och hur stor är den minsta vikten som du har plockat ut ur det stratifierade surveyobjektet?

c) Använd funktionerna `svymean()`, `svytotal()` och `svyby()` för att besvara följande frågor. Vilka är de stora skillnaderna jämfört med OSU? Vad får du för designeffekt? Pröva att skapa ett surveyobjekt bara för en region (ex. `NC`) och skatta totalen för detta strata som att det vore en egen undersökning. Jämför med resultatet från `svyby()` där du försöker skatta samma region som en redovisningsgrupp, får du samma resultat?

d) Pröva att kategorisera variabeln `ACRES87` i klasserna 0-100 000, 100 000-400 000 och 400 000 +. För detta kan funktionen `cut()` användas med fördel.

Exempel:

```
agpop$myStrata <- cut(agpop$ACRES87, c(0, 100000, 400000, Inf), include.lowest = TRUE)
```

Gör (med R eller för hand) en Neymanallokering av urvalet till de tre strata (se Lohr 2009). Dra ett stratifierat urval från `agpop` men stratifierat efter denna nya kategorivariabel istället och gör en ny totalskattning av `ACRES92`. Vad får du för resultat, vad får du för design effect? Blir det bättre eller sämre än att använda `REGION` som stratifieringsvariabel? Varför? Finns det några sätt som man skulle kunna förbättra stratifieringen i övrigt?
[**Tips!** för att beräkna standardavvikelsen i respektive strata kan `aggregate()` användas]

Referenser

Lohr, S., 2009. Sampling: design and analysis, 2nd Edition. Thomson.

Lumley, T., 2010. Complex surveys : a guide to analysis using R. Wiley-Blackwell, Oxford.