Genomgång sdcApp Steg-för-steg

Dessa instruktioner är till stöd för den som håller i workshopen och innehåller anteckningar om vad som kommer att gås igenom i sdcApp programmet innan deltagarna själva får prova på egen hand. I presentationen finns en slide med rubriken 'Genomgång sdcApp', vilken markerar när genomgången är tänkt att ske under workshoppen. Genomgången är strukturerad så att den börjar med huvudmenyn och ger översiktlig information om vad varje menyval används till. Därefter följer en genomgång av anonymiseringsprocessen och varje undermeny.

Översikt menyer

* Hur man startar sdcApp
* Hur man ställer in **storage path** innan man laddar in data och varför (dvs om man vill ha en rapport efteråt)
* Hur data laddas in i sdcApp - vilka olika dataformat som går att använda
* Visa att det finns ett informationstecken ovanför de flesta funktioner med förklaringar och instruktioner vilket är väldigt pedagogiskt.
* Kort om de olika alternativen som finns innan man laddar in data
* **Explore variables** innan man börjar pseudonymisera och omkoda data
* Gå igenom de olika menyerna kortfattat och vad man gör under dessa:
* **About/help** - kortfattat om programmet samt där du ställer in storage path, dvs dit du exporterar rapporter och pseudonymiserade datafiler.
* **Microdata** - här väljer du format och datafil som du vill använda.

- Under Microdata kan du också konvertera variabler till faktorvariabler eller numeriska variabler beroende på hur du vill använda dem senare i analysen i sdcApp, eller om du bedömer att en variabel automatiskt har importerats på ett felaktigt sätt och behöver ändras. Variabeltypen styr vilka funktioner som praktiskt går att använda.

* **Anonymize** - här väljer du de nyckelvariabler eller bakgrundsvariabler som du vill ha med i din analys, och om de skall tolkas som kategoriska eller kontinuerliga variabler. Variabler som står som “No” kommer inte att tas någon hänsyn till.
* Under **Risk/Utility** kommer du, efter att du laddat upp data, få fram en översikt av hur hög risk observationer i dina data löper för identifiering, dvs hur unika de är i det dataset du laddat in. Här kan du också se enskilda observationer som inte når upp till den gräns för

k-anonymitet som du satt som tröskel.

* Under **Export data** får du en överblick av det modifierade datasetet som du har skapat. Du kan också få ut en så kallad Anonymization report som är en slags loggbok över data och de ändringar som du genomfört.
* Under **Reproducibility** så får du ut all R-kod som ligger till grund för de kommandon som du genomfört i användargränssnittet. Detta kan användas som en mer detaljerad logg över de ändringar som du genomfört, för att upprepa samma ändringar eller för att kommunicera ändringar med en forskare som kan R.
* **Undo** kan användas till att ångra ditt senaste kommando.

Genomgång steg-för-steg

1. *Microdata*
   * Under Microdata laddar du in dina data.
   * Här får du möjlighet att välja hur dina data ska laddas in.
   * Du kan exempelvis välja vilken typ av variabeltyp dina variabler ska anta, om första raden ska tolkas som variabelnamn (i t.ex. en .csv), om missing values ska kodas om till NA som är R:s symbol för missing values samt hur du vill att data ska struktureras. I de flesta fall räcker det att ha defaultinställningarna.
   * Välj datafil och ladda in.
   * Ögna igenom data så att du är säker på att filen ser ut att ha laddats in på rätt sätt, dvs ligger variabelnamnet på översta raden, ligger variablerna i varsin kolumn osv.
   * Du kan också gå igenom enskilda variabler genom vänstermenyn Explore variables. Här kan du exempelvis titta på fördelningen på enskilda variabler, vilket kan vara bra att ha koll på inför eventuella omkodningar.
   * Om du vill konvertera variabler så behöver du göra det innan du sätter upp ett problem. Det gör du i vänstermenyn under **Convert numeric to factor** och **Convert variables to numeric**.
   * Det finns även en **Reset variables** meny där du kan återställa variabler som du omkodat.
   * Du kan också skapa stratifieringsvariabler, dvs kombinera två eller fler variabler för att skapa en klass, vilket kan vara användbart om du har väldigt många nyckelvariabler i din analys och vill skapa olika större strata. Inte gå in på djupet med detta.
   * Här kan du också ta bort specifika värden under **Set specific values to NA**.
   * *Notera att du behöver göra detta innan du skapar ett sdc Problem. När du väl satt upp ett problem kan du endast se data och titta på enskilda variabler.*
2. *Anonymize*
   * Här skapar du ditt så kallade **sdc Problem** med den data du laddat in. Här är det upp till dig att bedöma om några variabler bör raderas helt ur dina modifierade data och vilka bakgrundsvariabler som bör ingå i din röjandekontroll. I det här fallet ligger IP-adress, löpnummer och e-post med vilket kan klassas som direkta identifierare och bör raderas.
   * **Notera** att det endast är de variabler som du inkluderar i analysen som också tas hänsyn till i programmets bedömning av k-anonymitet etc. Detta steg är därför kanske det viktigaste i hela processen och kräver att du själv har koll på vilka variabler som utgör risk för identifiering i verkligheten. En utgångspunkt är att inkludera de variabler som du vet kan länkas till andra kompletterande datakällor såsom kön, ålder, kommun etc.
   * Under **Show summary** får du information kring de variabler som du valt ut. Du ser också hur många/få “datatvillingar” som kombinationen av dessa variabler har, samt hur stor andel av dina data som uppfyller/inte uppfyller k-anonymitet på olika nivåer.
   * Under **Recoding** kodar du om faktorvariabler. Det är ett av de vanligare sätten att modifiera tabulära data för att minska detaljrikedomen. Det är huvudsakligen den här funktionen som vi kommer att använda oss av idag.
   * Det finns dock andra metoder som PRAM (Post Randomization) som randomiserar värden, men det är överkurs idag.
   * **Notera** att vi inte har valt några numeriska eller kontinuerliga variabler för vår analys. Det kan man absolut göra och då finns en rad metoder som man kan använda sig för under menyn **Anonymize numerical values**, bland annat **Top-bottom-coding** som är ett sätt att få bort så kallade outliers, dvs värden som är speciellt unika från mängden och skapa stora standardavvikelser.
   * **Notera** att exempelvis ålder i det här fallet lika gärna hade kunnat hanteras som en numerisk variabel. I det här fallet har vi valt att behandla den som en kategorisk variabel för att det ger oss fler möjligheter att skapa specifika kategorier. Hade vi valt den som en numerisk/kontinuerlig variabel i problemformuleringen hade vi inte kunnat koda om den på det sätt som vi vill i det här fallet utan endast ta bort outliers eller använda

mikro-aggregering. Micro aggregation grupperar ihop de närmsta värdena (du väljer själv hur många) för en variabel i data och redovisar medelvärdet för den grupp den skapat. Det är inte speciellt användbart när det kommer till ålder men kan vara det i andra fall.

* + Under **Recoding** kodar du som sagt om faktorvariabler eller kategoriska variabler. Du väljer vilken/vilka variabler du vill koda om samt hur du vill koda om dem. Att koda om betyder i regel att du klumpar ihop värden i bredare kategorier i en variabel för att skapa fler “tvillingar”, vilket i sin tur gör det svårare att urskilja enskilda värden ur mängden.
  + En utgångspunkt när du kodar om är såklart att de nya klasser eller kategorier du skapar är någorlunda logiska, dvs du klumpar inte ihop 18-åringar med 85-åringar eller lågutbildade med högutbildade, utan de bör efterlikna originaldata i så stor utsträckning som möjligt.
  + Fördelningen av värden i en variabel kan ge viss vägledning i hur man bör koda om. Exempelvis om det är väldigt få äldre i dina data kan det vara en idé att göra en större kategori och inkludera fler värden i denna kategori etc.
  + När du kodar ihop värden, var också vaksam på fördelningen mellan de klasser du skapar. Om en klass eller kategori är mycket mindre än de övriga klasser du skapar är risken större för att individer i denna kategori går att urskilja. Med andra ord är det en balansgång mellan att skapa så användbara kategorier som möjligt samtidigt som du vill skapa så säkra data med breda kategorier som möjligt.
  + **Notera** att du efter varje omkodning får en realtidsuppdatering i högerspalten kring hur stor andel av dina data som inte uppfyller tröskeln för k-anonymitet. Du kan med andra ord följa vilken effekt som dina omkodningar får på data.
  + Under menyn **k-Anonymity** finns ett verktyg för att selektivt ta bort datavärden som är problematiska för att uppnå k-anonymitet. Funktionen kallas **Establish k-anonymity**. Med detta verktyg maskas (eller eg raderas) värden i en eller flera variabler bland de observationer som inte uppfyller den grad av k-anonymitet som du satt. Du väljer först tröskel för

k-anonymitet som du vill ha. Du kan sedan välja hur programmet ska prioritera maskningen om det är så att någon av dina variabler är viktigare att ha kvar värden i än andra. Att använda detta verktyg rekommenderas inte initialt men kan i vissa fall användas i slutskedet om det endast är ett fåtal problematiska observationer kvar som inte uppnår dina tröskelvärden för

k-anonymitet.

* + När du är klar med dina omkodningar går du över till menyn **Risk/utility**.

1. *Risk/utility*

Risk

* + Under denna meny kan du utvärdera hur dina modifieringar av data får för konsekvenser för risken för identifiering samt hur det påverkar datas användbarhet.
  + Du kan exempelvis titta i detalj på de observationer som inte när upp till ditt tröskelvärden för k-anonymitet. Detta är ett ganska användbart verktyg för att snabbt få en överblick.
  + Under **Information of risk** ser du vilken risk som observationer i dina data hade innan du modifierade data och hur de ser ut nu. Värdet går från 0-1. De observationer som ligger på

1.0 är helt unika, dvs de löper 100% risk att urskilja. De som ligger på 0,0 har tillräckligt många “tvillingar” för att inte kunna urskiljas från mängden.

* + Under menyn **Obs. violating k-anonymity** finns ett användbart verktyg för att visa vilka observationer som inte uppnår ditt tröskelvärde. Här kan du ställa in vilken nivå av

k-anonymitet som du vill ha och programmet visar då alla observationer som inte uppnår

k-anonymitet. I det här fallet har jag valt k-3-anonymitet och får fram x observationer som jag kan titta på i detalj.

* + **Notera** att sdcApp endast tar hänsyn till huruvida observationerna i dina data är unika eller inte. Det analyserar med andra ord endast de data som du laddar in i programmet och kan inte ta hänsyn till något annat.
  + Måttet på k-anonymitet säger alltså endast något om hur många observationer som saknar tvillingar i data, inget annat.
  + Det är med andra ord *upp till dig* att göra en bedömning hur säkra data är utifrån de andra utomliggande och kontextuella faktorer som föreligger.
  + Du behöver därför titta noggrant på de modifierade data du har skapat för att själv göra en bedömning. I vissa fall kan omständigheterna vara så att k-2 anonymitet kan vara fullgott och i andra fall behöver du ha en mycket högre nivå.

Utility

* + Under Visualizations -> **Information loss** kan du utvärdera hur dina omkodningar påverkar detaljrikedomen i dina data. Du får en tabell med en jämförelse hur dina bakgrundsvariabler såg ut innan du kodade om dem och hur de ser ut nu. T.ex. kan du se hur många observationer som ingår i de nya klasser som du omkodat jämfört med i de ursprungliga.
  + Detta ger en indikation på hur mycket information som går förlorad efter omkodningar. Du får inte ut en specifik siffra utan *behöver själv* göra en bedömning kring förlusten av detaljrikedom och eventuellt kan påverka senare analys.
  + Ett annat sätt att göra detta är att faktiskt göra körningar på dina omkodade data för att se om de samband som du tidigare belagt består. Det är så man hade behövt göra i verkligheten men inget vi har tid för idag.
  + **Notera** att det även finns en rad sätt att utvärdera numeriska variabler. Vi kommer inte att gå igenom det idag, men här kan man få fram ett korrelationsvärde mellan ursprungliga variabeln och modifierade variabler. Värdet går mellan 0-1 och ger en indikation på hur lika/olika dina modifierade data är i jämförelse med originaldata, där 0 betyder att dina modifierade variabler avviker totalt från originaldata och 1 betyder att de är detsamma som innan.
  + **Take-home-point:** Återigen, det går inte fullt ut att utvärdera balansen mellan risk och användbarhet genom att endast titta på data.
  + Detta måste man göra genom att själv fundera kring hur de kategorier man har skapat begränsar möjligheten till analys och hur data är tänkt att användas.
  + Det enda vi får svar på genom sdcApp är hur data såg ut innan/efter omkodningar.
  + **Exempel**: I det här fallet har vi exempelvis begränsat möjligheten att analysera hur människor inom länen röstar och tycker om olika politiska frågor.
  + Vi vet exempelvis att det ofta föreligger viktiga skillnader i röstningsbeteende mellan de som bor i storstadskommuner och de som bor i mindre tätorter eller på ren landsbygd. Efter våra omkodningar mister vi möjligheten att analysera hur personer utanför Stockholm, Göteborg och Malmö röstar och kan bara analysera hur man röstar på länsnivå.
  + **Take-home-point:** Detta är en av de viktigaste poängerna med hela workshopen, dvs även om den här typen av program är väldigt användbara för att bedöma risk och användbarhet, är de begränsade av det material du matar in, dvs de kan inte ta någon som helst hänsyn till utomliggande faktorer och kontexter eller hur data är tänkt att användas.

1. *Export data*
   * Under menyn Export data kan du gå igenom dina modifierade data och spara ut dem.
   * Du kan också få ut en s.k. Anonymization report som är en slags loggbok med information om alla steg som du genomfört. Detta kan vara väldigt användbart i dokumentationssyfte och för att kommunicera till en forskare eller sekundäranvändare hur data har behandlats.
2. *Reproducibility*
   * Här får du som sagt ut det R-skript som ligger till grund för alla modifieringar. Det är väldigt användbart om du exempelvis vill visa vad du har gjort och om en “vanlig” R-användare snabbt vill köra skriptet på sin dator eller bara se vad som har gjorts i detalj.