Datorlaboration 6

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 paket som ska användas

- Läsa in följande paket i R:

```
library(survey)
library(maptools)
library(rmeta)
library(hexbin)
library(sweSCB)
```

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 exempelvis följande kod.

```
install.packages("survey")
```

Om det inte går att läsa in paketet `sweSCB` (ex. du har en egen dator) behöver du installera paketet först. Om det inte går att läsa in paketet (ex. du har en egen dator) behöver du installera paketet först.

```
install.packages("sweSCB")
library(sweSCB)
```

Mer information finns här: **sweSCB**.

1.2 Ladda ned och läs in Survey 2010

- I denna laboration ska vi börja analysera riktiga surveydata (med alla problem det innebär). Vi har fått tillgång till det datamaterial som ligger till grund för boken *Den svenska väljaren* Hagevi (2011). Ett mindre urval av variablerna i studien har sparats som en R-fil. Ladda ned filen från kurshemsidan och läs in den i R med följande funktion

```
load("svy2010.Rdata")
```

- Det går också att ladda in filen direkt från webben med `repmis`-paketet:

```
library(repmis)
source_RData(url = "https://raw.github.com/MansMeg/KursSvyMeth/master/Labs/DataFiles/svy2010.Rdata")

## Downloading data from: https://raw.github.com/MansMeg/KursSvyMeth/master/Labs/DataFiles/svy2010.R
##
## SHA-1 hash of the downloaded data file is:
## d11fe835c8306f4746b9f950bc128985049815e2
```

- Du ska nu ha läst in en fil med 1613 observationer och 70 variabler. Infomation om respektive variabler finns i dokumentet KodbokSurvey2010.pdf som finns på samma ställe som datamaterialet, dock finns inte alla variabler med i datasetet.
- Vi ska nu skapa två kontinuerliga variabler i datasetet `fathAge` och `mothAge` på följande sätt:

```
svy2010$fathAge <- svy2010$FR37 - svy2010$FR40_1
svy2010$fathAge[abs(svy2010$fathAge) > 100 | svy2010$fathAge < 0 ] <- NA
svy2010$mothAge <- svy2010$FR37 - svy2010$FR40_2
svy2010$mothAge[abs(svy2010$mothAge) > 100 | svy2010$mothAge < 0] <- NA
```

1.3 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=",")
```

- Även i detta fall kan vi självklart använda `repmis`-paketet.

```
agpop <- source_data(url = "https://raw.github.com/MansMeg/KursSvyMeth/master/Labs/DataFiles/agpop.da

## Downloading data from: https://raw.github.com/MansMeg/KursSvyMeth/master/Labs/DataFiles/agpop.dat
##
## SHA-1 hash of the downloaded data file is:
## fdd78ace764a7b61254f073564ab50968cdd9375
```

- 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 Grafik

Vi ska nu skapa grafik baserat på surveydata. Eftersom vikter är något som påverkar våra variabler och vår grafik väljer vi därför att dels använda en stratifierad urvalsdesign som exempel och dels data från Survey 2010. Visualisering är en konstform så försök göra din grafik så snygg och tilltalade som möjligt. Använd gärna färger.

Börja med att skapa de två surveyobjekten vi ska använda, dels baserat på agpop och dels baserat på Survey 2010.

Dra ett (Neymanallokerat) stratifierat urval från `agpop` och skapa ett surveyobjekt på samma sätt som gjordes i laboration D4. Jag döper mitt objekt till `agSTRAT`.

Skapa sedan ett surveyobjekt baserat på datamaterialet i Survey 2010, jag kommer kalla detta objekt `svy2010design`.

2.1 Kategoriska variabler

a) Vi börjar med att producera enklare grafik baserat på surveyresultaten. Denna grafik är inte specifik för surveyundersökningar, men nu skapar vi grafer baserat på våra skattningar av populationen istället på hur data ser ut i vårt urval.

```
bys <- svyby(formula=~FARMS92, by=~REGION, design=agSTRAT, FUN=svytotal)
barplot(bys)
dotchart(bys)
```

Vi kan också inkludera både antalet observationer och vår osäkerhet med en så kallad `forestplot()` från `rmeta`-paketet.

```
labelMatrix <- matrix(rownames(bys), nrow=4)
forestplot(labeltext=labelMatrix,
                   mean = coef(bys),
                   lower= coef(bys) - 1.96 * SE(bys),
           upper = coef(bys) + 1.96 * SE(bys))
```

b) Vi ska nu gå in på de mer specifika funktioner som finns för surveydata. Exempelvis kan vi använda `svyboxplot()` för att skapa boxplots för populationen vi vill estimera. Funktionen estimerar kvantilerna i populationen och skapar en boxplot baserat på dessa estimat.

```
svyboxplot(ACRES92 ~ REGION, design=agSTRAT)
```

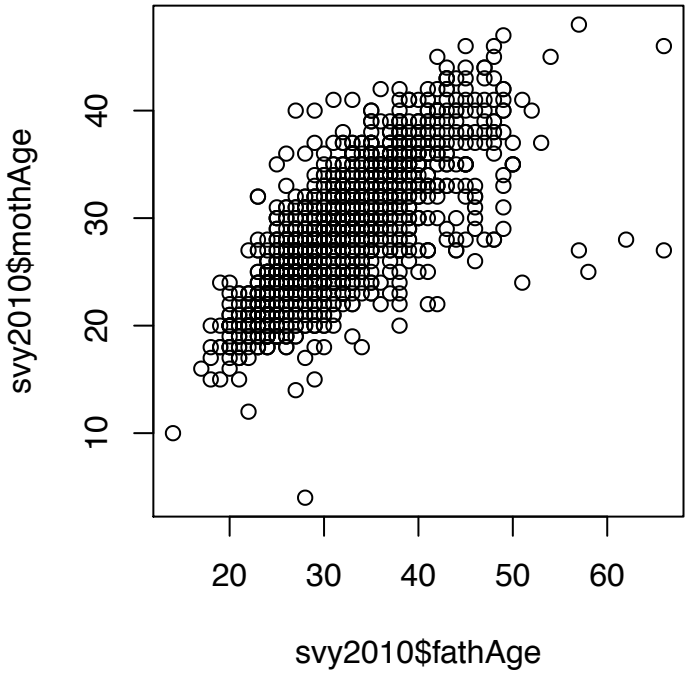
På ett liknande sätt fungerar funktionen `svyhist()` som skapar ett histogram baserat på de viktade observationerna.

```
svyhist(~ ACRES92, design=agSTRAT)
```

2.2 Scatterplots i surveyer

a) Som nämnts tidigare är ett (delikat) problem i surveyer ofta att antalet observationer är mycket stort. Det gör att scatterplots ofta kan bli helt övertäckta med datapunkter vilket gör det svårt att se resultaten. Nedan är ett exempel på scatterplot där det är mycket svårt att se fördelningen då punkterna ligger "ovanpå" varandra.

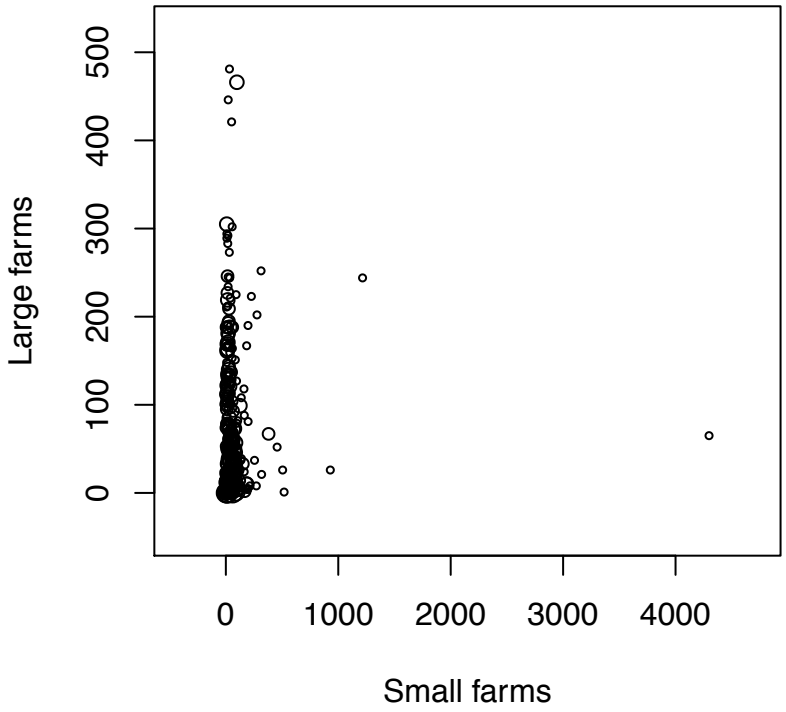
```
plot(svy2010$fathAge, svy2010$mothAge)
```



Vi ska nu pröva olika former av alternativ för surveydata.

Dessa plottar kan visas på olika sätt. Dels kan vikterna tas med i beaktande och explicit användas i en scatterplot. Detta görs med funktionen `svyplot()` på följande sätt. Nedan skappas en "bubbleplot" men där storleken på bublorna är baserade på vikterna i datamaterialet.

```
svyplot(LARGEF92~SMALLF92,design=agSTRAT, style="bubble",xlab="Small farms", ylab="Large farms")
```



Nedan är exempel på andra metoder för att producera plottar för surveydata (eller generellt stora data).

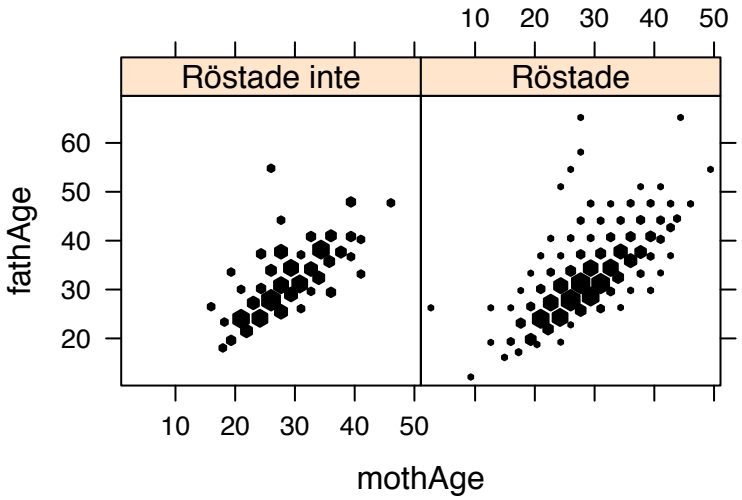
```
svyplot(fathAge~mothAge, design = svy2010design, style="transparent", pch=19, alpha=c(0,.1))
svyplot(fathAge~mothAge, design = svy2010design, style="hex")
svyplot(fathAge~mothAge, design = svy2010design, style="grayhex")
```

Skapa plottarna ovan och inkludera dem i din rapport. Gör dem så snygga du kan (se hjälpen) med labels på x- och y-axeln. Vilka plottar föredrar du och varför?

b) Till sist ska vi pröva att skapa så kallade "conditioning plots" d.v.s. scatterplots för olika grupper. Dessa plots påminner om funktionerna ovan, dock med tillägget | som indikerar för vilken kategorisk variabel ska plottarna delas upp. Nedan är ett exempel.

```
svycoplot(fathAge~mothAge | Valdeltagande, design = svy2010design, style="hex")

## Loading required package: lattice
```



Välj ut en variabel och skapa en så snygg conditioning plot du kan. Använd hjälpen för att se hur kan styra olika delar.

2.3 Kartor

a) I surveyer och liknande undersökningar är det inte sällsynt att det finns ett intresse av producera kartor av det material som samlas in. Vi kommer därför gå igenom den mest grundläggande funktionalitet i R-paktetet `maptools` för att skapa kartor i R.

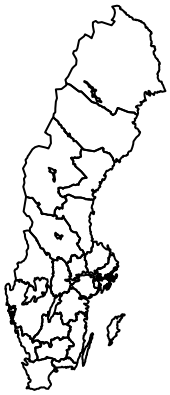
För att skapa kartor behöver vi en så kallad shapefil. Dessa innehåller geografisk data i vektoriserat format vilket vi kan läsa in och modifiera i R. Mer information om shapefiler finns **[här]** och om vektoriserad GIS-information finns **[här]**.

Shapefiler som är aktuella för just era problem beror på detaljeringsgrad, men församlingsgränser eller kommunindelningsgränser av intresse finns på de flesta kommuner och länsstyrelser. Ofta finns informationen att tillgå gratis som shapefiler. På kurshemsidan ligger shapefiler som innehåller kommun och länsgränser för Sverige. Ladda ned dem och läs in dem i R på följade sätt (**Obs!** paketet `maptools` måste vara inläst i R):

```
sweMap <- readShapePoly("Lan_SCB_07.shp")
```

För att skapa en karta över den fil du precis läst in används `plot()`:

```
plot(sweMap)
```



b) Vi ska nu skapa statistik på karta baserat på surveyestimat. Det första steget är att skapa den variabel vi vill beskriva med kartan. Jag vill analysera andelen som vill skära ned på den offentliga sektorn i olika områden. Först skapar jag variabeln nedan, observera att vårt dataset ligger i designobjektet och har namnet `variables`.

```
svy2010design$variables$minskaOff <- as.numeric(svy2010$FR28_1 %in% c("Ganska bra", "Mycket bra"))
```

Nästa steg är att skapa skattningar för respektive län, detta görs med `svyby()`.

```
minskaOffEst <- svyby(~minskaOff, by=~LKF, design=svy2010design,  svymean)
```

Precis som surveyobjektet har ett eget element där datasetet är lagrat har data baserat på shapefiler ochså ett ingbyggt dataset. Det är till detta vi ska lägga på våra estimat, vi lägger till / sammanfogar våra data till denna fil. För att titta på det data som finns den geografiska filen:

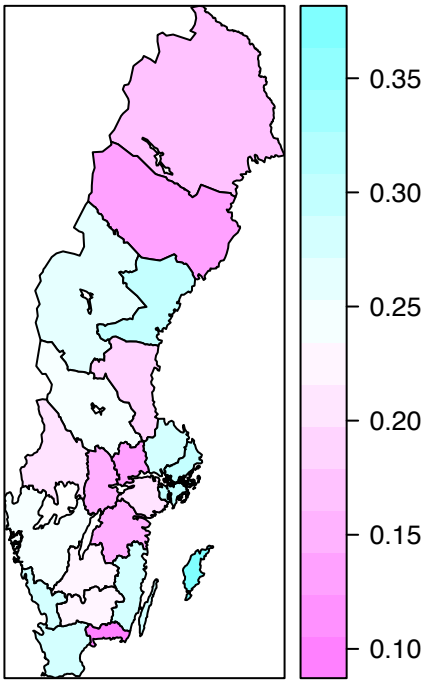
```
sweMap@data
```

För att lägga till våra estimat används följande kod:

```
sweMap@data <- merge(sweMap@data, minskaOffEst, by.x="LNKOD", by.y="LKF")
```

Nu har vi lagt till våra estimat och kan använda `spplot()` för att skapa en karta baserad på vår statistik.

```
spplot(sweMap, "minskaOff")
```



Vi har nu gjort en karta över andelen som vill minska den offentliga sektorn i olika landsting. Välj nu en annan fråga och gör en så snygg karta som möjligt. Gör också en mindre analys kring de geografiska skillnaderna ni ser.

c) Som ni kanske upptäckt är ett problem med estimaten att de är olika säkra för olika län. Exempelvis i Stockholms län finns många observationer medan på Gotland är antalet observationer mycke mindre Det som då kan vara av intresse är att kombinera län till större områden. Det finns i Survey 2000 så kallade riksområden som vi kan använda.

Lägg till variabeln `romr` i det spatiala objektets dataset på liknande sätt som ovan. Efter att det är gjort kan vi använda funktionen `unionSpatialPolygons` för att lägga samman länsområdena (polygoner) till större områden.

```
sweMapLarge <- unionSpatialPolygons(SpP = sweMap,IDs = sweMap@data$romr)
```

På detta sätt får vi ett nytt spatialt objekt med de nya geografiska områdena:

När vi lägger ihop olika polygoner i en karta kommer objektet att reduceras till ett `SpatialPolygons`-objekt istället för ett objekt av klassen `SpatialPolygonsDataFrame` (som krävs för att visualisera data). Vi behöver därför aggregera upp data på den nya indelningsnivån och skapa ett objekt av klassen `SpatialPolygonsDataFrame`.

```
aggData <- aggregate(x=sweMap@data$BEF05,by=list(sweMap@data$romr),FUN=sum)
names(aggData) <- c("romr", "BEF05")
sweMapLargeWithData <- SpatialPolygonsDataFrame(Sr=sweMapLarge, data=aggData)
```

Med detta objekt kan vi sedan visualisera data med de nya områderna med hjälp av följande kod:

```
spplot(sweMapLargeWithData, "BEF05")
```

Använd dessa nya större områden för att igen skapa en karta baserad på dessa områden med statistik för den variabel du använt. Gör en så snygg karta du kan. Skiljer sig resultaten från varandra?

d) Som ett sista steg ska vi plocka ut endast delar av en karta för att skapa kartor över mindre delområden. Börja med att läsa in shapefilen som innehåller kommungränser. Pröva att skapa en karta med samtliga kommungränser.

```
sweMunipMap <- readShapePoly("Kommun_SCB_07.shp")
```

Titta på kartan med följande kod:

```
plot(sweMunipMap)
```

Som ett nästa steg ska vi välja ut ett enskild län och plotta de olika kommunerna i detta län. Detta gör på ett liknande sätt som för dataset i R. För att välja ut länet "01" (Stockholms län) används följande kod.

```
sweMunipMap01 <- sweMunipMap[substr(sweMunipMap$KNKOD,1,2)=="01",]
```

Studera kartan med följande kod:

```
plot(sweMunipMap01)
```

e) Vi ska nu skapa en karta över Östergötland för en godtycklig statistisk storhet. Ta reda på vilken länskod Östergötland har. Gå in på SCB och hämta ned data på kommunnivå för Östergötlands län. Du kan välja vilken statistik du vill (ex. valresultat finns på kommunal nivå) och skapa en karta över denna variabel för Östergötland.

För att snabbt ladda ned data till R kan du använda paketet `sweSCB`.

```
library(sweSCB)
myData <- findData()
```

3 Analys

Precis som när det gäller att skatta medelvärden, proportioner eller andra populationsparametrar innebär designperspektivet att vi betraktar populationsparametrarna som fixa "sanna" värden i den ändliga populationen. Vi drar sedan ett urval för att kunna dra statistiska slutsatser om denna ändliga population. Dessutom kan vi på grund av att urvalet är ett stratifierat urval eller ett klusterurval få en designeffect i våra studier som behöver hanteras när vi gör våra tester.

Denna laboration kommer endast att beröra det specifika med just surveysituationen, inte kring dessa tester och modeller i sin helhet.

3.1 t-test

Många gånger finns det ett intressa av att jämföra olika redovisningegrupper efter en variabel som kan vara av intresse. Finns det skillnader i vår **ändliga population** avseende redovisningsgrupper som inte beror på slumpen vi skapat i vårt urval. Vill vi jämföra två grupper med varandra (eller med ett fast värde) avseende en kontinuerlig variabel använder vi funktionen `svyttest()`. För att göra testet behövs dels ange en formula, vilket består av den kontinuerliga variabeln vi vill testa skillnader avseende (ex. y) och en kategorisk variabel med två klasser (ex. `grupp`). Sedan krävs designobjektet vi vill testa.

Exempel:

```
svyttest(formula = y ~ grupp, design = mittDesignObjekt)
```

Nedan har jag gjort ett t-test. Vad har jag testat? Vad är slutsatsen från testet?

```
svyttest(formula=mothAge~Valdeltagande, design=svy2010design)


Design-based t-test

data:  mothAge ~ Valdeltagande
```

```
t = -0.2186, df = 1611, p-value = 0.827
alternative hypothesis: true difference in mean is not equal to 0
sample estimates:
difference in mean
           -0.1138
```

Använd surveyobjektet du skapat för Survey 2010 och testa om det finns skillnader mellan moderata och socialdemokratiska väljare när det gäller moderns ålder. Vad får du för resultat? Vad är din slutsats?

Obs! Först behöver du plocka ut de personer i undersökningen som röstade på moderaterna eller socialdemokraterna. Det enklaste är att göra detta med `subset()`. Är du osäker på hur du använder `subset()` - se laboration D4 och avsnittet om redovisningsgrupper.

3.2 Ändlighetskorrektionens betydelse för designbaserade test

Pröva nu att göra om surveyobjektet `svy2010design` så att de svarande utgör hela populationen. D.v.s. ändra i argumentet `fpc` som du använde för att skapa surveyobjektet så att storleken på populationen blir densamma som urvalsstorleken. Gör nu om testet i 3.1. Får du ett annat resultat? Vad beror detta på? Förklara.

3.3 χ^2 - test

Ett annat vanligt test för kategoriska variabler är det klassiska χ^2-testet. Precis som för t-testet behöver vi korrigera våra tester för att ta hänsyn till dels vår ändlighetskorrektionsfaktor och dels för att beakta potentiella designeffekter.

För att genomföra ett test om valdeltagande är oberoende av vilket parti en person föredrar kan ett χ^2-test användas på följande sätt.

Exempel:

```
svytable(formula = ~FR15 + Valdeltagande, design = svy2010design)

                                    Valdeltagande
FR15                                   R<U+00F6>stade inte   R<U+00F6>stade
  Centerpartiet                                     18672         322100
  Feministiskt initiativ                             9336          28009
  Folkpartiet                                       37345         606855
  Kristdemokraterna                                  9336         228738
  Milj<U+00F6>partiet                                102699         723558
  Moderaterna                                       168052        1969945
  Piratpartiet                                       28009          32677
  Socialdemokraterna                                121371        1708531
  Sverigedemokraterna                                37345         252078
  V<U+00E4>nsterpartiet                                18672         340773
  Annat parti (v.g. ange vilket)...?:               42013          65354

svychisq(formula = ~FR15 + Valdeltagande, design = svy2010design)


Pearson's X^2: Rao & Scott adjustment

data:  svychisq(formula = ~FR15 + Valdeltagande, design = svy2010design)
F = 6.497, ndf = 10, ddf = 16120, p-value = 4.321e-10
```

Vad är din slutsats av testet ovan?

Som standard används Rao-Scott-korrektion av testet. Vi kan välja andra korrektioner som exempelvis Wald-korrektion genom att sätta argumentet `statistic` till `''Wald''`.

Exempel:

```
svychisq(formula = ~FR15 + Valdeltagande, design = svy2010design, statistic="Wald")
```

```
Design-based Wald test of association

data:  svychisq(formula = ~FR15 + Valdeltagande, design = svy2010design,     statistic = "Wald")
F = 2.312, ndf = 10, ddf = 1612, p-value = 0.0107
```

Välj nu två kategoriska variabler du vill testa om de är oberoende av varandra och testa dem med ett designbaserat χ^2-test. Vad är dina slutsatser?

3.4 Linjär regression

Till sist ska vi pröva att använda linjär regression i surveysammanhang, d.v.s. att i en regression ta hänsyn till urvalsdesign och ändlighetskorrketion. Precis som vid testerna ovan innebär att detta att vi skattar dessa värden i den ändliga populationen.

Vi ska nu pröva att anpassa en regressionsmodell på materialet avseende åkerytor i USA.

```
myModel <- svyglm(formula = ACRES92 ~ FARMS92 + REGION, design=agSTRAT)
myModel

Stratified Independent Sampling design
svydesign(~1, strata = ~REGION, fpc = fpc.strata, data = agSTRATAdata)

Call:  svyglm(formula = ACRES92 ~ FARMS92 + REGION, design = agSTRAT)

Coefficients:
(Intercept)      FARMS92     REGIONNE      REGIONS      REGIONW
     253372          100      -229384      -101516       444257

Degrees of Freedom: 299 Total (i.e. Null);  292 Residual
Null Deviance:     4.62e+13
Residual Deviance: 3.42e+13  AIC: 8550
```

Vad har jag gjort för analys ovan? Vad är dina slutsatser baserat på denna analys?

Obs! Precis som vid vanlig regression kan vi använda `summary()` för att få ytterligare information från vår modell (hypotestester m.m.).

Pröva sedan att anpassa en vanlig linjär regression på samma data (men utan att ta hänsyn till designen).

Exempel:

```
myLmModel <- lm(ACRES92 ~ FARMS92 + REGION, agSTRATAdata)
```

Vad är skillnaden i resultat mellan de två metoderna? Vad beror denna skillnad på?

Skapa nu ett nytt designobjekt för hela populationen `agpop`. Se till att ange en korrekt ändlighetskorrektion för denna population (d.v.s ange att det är hela populationen vi analyserar). Gör om samma linjära regressionsanalys som vi gjorde på urvalet `agSTRAT` tidigare.

Vad får du medelfel på koefficienterna? Täcker konfidensintervallen du fick baserat på urvalet `agSTRAT` de sanna värdena i populationen?

Referenser

Hagevi, M., 2011. Den svenska väljaren, 1st Edition. Boréa, Umeå.

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