Datorlaboration 3, SÖK

Ulf Högnäs

Översikt

Innehåll

- Hypotest och p-värden
  - med randomization
  - med sannolikhetsmodell
- Konfidensintervall
  - med the Bootstap
  - med sannolikhetsmodell
- t-fördelningen och funktionen `t.test()`

1 - Hypotestest för en andel - EUs medborgarinitiativ

Få mer att säga till om i frågor som berör dig direkt. Med ett europeiskt medborgarinitiativ kan du bidra till EU-politiken genom att uppmana EU-kommissionen att föreslå nya lagar.

medborgarinitiativ

För att få igenom ett medborgarinitiativ krävs namnunderskrifter

Du måste få stöd från minst en miljon EU-medborgare och samla in ett minsta antal underskrifter i minst sju EU-länder

För Sverige är det minsta antalet 15 120. Låt oss säga att vi driver intiativet **Stop Destroying Videogames** och att vi har samlat in 18 910 underskrifter. Vi vet att vissa av dessa underskrifter är ogiltiga. Det kan till exempel vara för att den som skrivit på inte har fyllt 18 år. Om 80% eller fler av underskrifterna är giltiga så har vi nått spärren på 15 120 eftersom:

$$18910 \cdot 0.80 = 15128$$

Vi har inte tid att kontrollera samtliga underskrifter, så vi kontrollerar istället ett slumpmässig urval på 200 underskrifter. I vårt slumpmässiga urval så finner vi att **176 underskrifter är giltiga**, medan resterande **23 är ogiltiga**.

> **i** Uppgift 1.1
>
> Använd R för att beräkna stickprovsandelen. Spara resultatet som en variabel med ett lämpligt variabelnamn, t.ex. `p_hat`.

Vi börjar med en beskrivning av hypotestest med randomization i samma stil som bokens.

1. Antag att 80% av namnunderskrifterna är ogiltiga
2. Skriv "giltig" på 160 kort och "ogiltig" på 40 kort
3. Dra ett kort med återläggning 200 gånger, blanda mellan varje dragning.
4. Beräkna andelen som blev "giltig"
5. Upprepa steg 3 och 4 tiotusen gånger.

Detta tar för lång tid, så vi använder R istället. Funktionen `rbinom()` skapar den slumpmässig vektor med dragningar av detta slag. Namnet kommer från orden *random* och *binomial*. Binomialfördelningen är en sannolikhetfördelning som vi har valt att inte ha med i kursen, trots att den är viktig. Vi nöjer oss med att säga att binomialfördelningen besvarar frågor av typen *om jag genomför ett och samma försök tio gånger, oberoende av varandra, vad är sannolikheten att jag lyckas minst åtta gånger?*

Här är en rad kod som genomför randomizeringen 10 000 gånger och sparar resultaten under namnet `MI_random`.

```
# n = antalet simuleringar
# size = antalet "dragna kort" per simulering
# prob = sannolikheten att "lyckas", i vårt fall att få "giltig"
MI_random <- rbinom(n = 1e4, size = 200, prob = .8)
# titta på de första 20 resultaten
head(MI_random, 20)
```

Nu vill vi gå från antal till andel. Andel i detta fall är ju hur stor del av 200 som blev "giltig". Vi delar däför varje antal i vår resultatvektor med 200:

```
MI_proportions <- MI_random/200
head(MI_proportions, 20)
```

Uppgift 1.2

Skapa ett histogram över de simulerade andelarna. Antingen med `hist()` i base-R eller med `histogram()` från `mosaic` i lab 2. Ändra antalet `breaks` tills du tycker att det ser bra ut.
Om du använder `hist()` så kan du dra ett vertikalt sträck vid `p_hat` med följande rader

```
# hist() kräver inte formatet data.frame
hist(MI_proportions, breaks = 30)
# v för vertikal, col för color, lty för line type
abline(v = p_hat, col = "red", lty = 3)
```

Vi har ju simulerat resultat under antagandet att 80% av underskrifterna är giltiga. Titta på histogrammet och jämför med den stickprovsandel som vi beräknade i Uppgift 1.1. Vad har vi visat med denna simulering? Förklara!

Nu ska vi skatta (uppskatta) ett p-värde. Först behöver vi hypoteser.
Låt p vara andelen ogiltiga underskrifter bland de 18 890.

$H_0 : p = 0.80$
$H_0 : p > 0.80$

Vi har också valt en gräns för statistisk signifikans. Eftersom vi vill vara nästan säkra på att vi har tillräckligt många underskrifter valde vi en låg gräns, 1%.

Vårt p-värde blir svaret på frågan *om den verkliga andelen giltiga underskrifter är 80%, hur ofta hade vi sett 176 giltiga eller fler, i ett stickprov på 200?*

Uppgift 1.3

Använd resultatet från simuleringen (`MI_proportions`) för att skatta p-värdet. Jämför p-värdet med vår gräns på 1%. Dra en slutsats.

Tips!

Du kan beräkna antalet eller andelen platser i en vektor som är större eller lika med ett visst tal på följande sätt

```r
# skapa en vektor som du kallar "numbers", att testa med
numbers <- c(0, 7, 1, 2, 1, 8, 7, 1, 0, 0)
# antalet platser som är större än 5
sum(numbers>=5)
# andelen platser som är större än 5
mean(numbers>=5)
# en tabell
table(numbers>=5)
```

Syftet med att noggrant gå igenom simulering av andel är att det ska öka er förståelse för sannolikheter och abstrakta begrepp som p-värde. I praktiken kan vi använda `prop.test()` istället. Funktionen `prop.test()` använder sig av en sannolikhetsmodell, så vi bör kontrollera the success-failure condition först.

Uppgift 1.4

Upprepa hypotestestet från uppgift 1.3 med `prop.test()`. Läs om fuktionens arguments genom att skriva`?prop.test` i Console. Du kommer att behöva använda argumenten `x, n, p` och `alternative`. Jämför p-värdet med värdet från uppgift 1.3. Fick du ungefär samma?

För att spara resurser tillåter EU ansvariga myndiheter i varje land att undersöka ett stickprov av underskrift som samlats in för merborgarinitiativ, istället för att kontrollera varje underskrift. Varje mynighet måste noggrant rapportera vilka rutiner statistiska metoder som använts.

2 - Konfidensintervall - Genomsnittsbetyg för filmer på Netflix

Tidigare har vi bara räknat konfidensintervall för andelar. Nu ska vi titta på genomsnittsbetyg för Netflixfilmer. Den data vi ska använda är hämtad från www.kaggle.com. Den användare som laddat upp datamaterialet har i sin tur hämtat det från en websida som heter The Movie Database. Websidans användare har givit filmer och tv-serier betyg mellan 0 och 10 och sidan presenterar ett genomsnittbetyg för varje film. Hela datamaterialet består av 16 000 filmer, men vi kommer att analysera ett stickprov på 100 filmer.

Uppgift 2.1

Ladda ner datamaterialet från kurshemsidan
netflix_sample.csv
Flytta filen till ditt working directory. Läs in datamaterialet med `read.csv()`, spara som `movie_sample` (t.ex.), och titta på innehållet med `head()` eller `View()`. Skapa ett

histogram över medelbetygen för filmerna.

Vi börjar med att skapa ett 95%-igt konfidensintervall med bootstrap.

```
antal_bs <- 1e3 # Välj antal bootstrapstickprov, 1000
resultat <- numeric(antal_bs) #skapa en tom vektor för att spara
# resultaten
# Det som kommer nu är en "for loop"
for (i in 1:antal_bs){
  # ta ett bootstrap sample
  bootstrap1 <- sample(movie_sample$rating, size = 100, replace = TRUE)
  resultat[i] <- mean(bootstrap1) # spara bootstrapstickprovets
  # ...medelvärde på plats 'i' i vektorn 'resultat'
}
hist(resultat, breaks = 30,
     main = "Stickprovsmedelvärden, 1000 bootstrapstickprov")
# rita röda, prickade linjer vid percentilerna 2.5% och 97.5%
abline(v = quantile(resultat, 0.025), col = "red", lty = 2)
abline(v = quantile(resultat, 0.975), col = "red", lty = 2)
quantile(resultat, c(0.025, 0.975))
```

i Uppgift 2.2

Titta på bootstrapskonfidensintervallet. Tolka resultatet med ord. Varför har vi valt percentilerna 2.5% och 97.5%?

Nu ska vi istället använda en sannolikhetmodell. Från föreläsning 6 minns vi att när vi kan säga att stickprovsmedelvärdet är approximativt normalfördelat så kan vi beräkna ett konfidensintervall med

$$\text{punktskattning} \pm z^\star \times \text{SE}$$

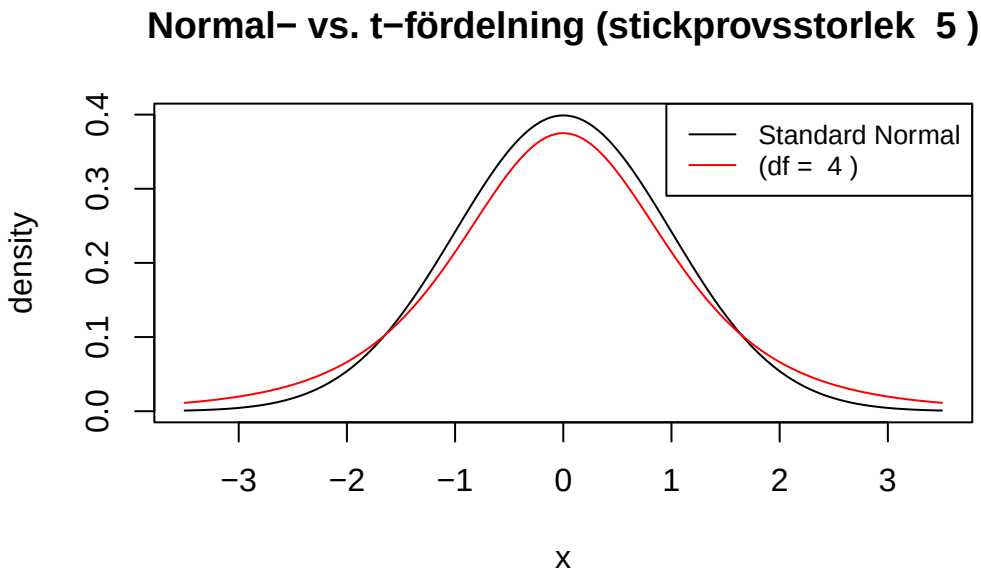
Vi behöver tre delar punktskattning, $z^\star$, som vi kommer att kalla $t^\star$ och standardfelet SE.

1. **punktskattning** Detta är stickprovets medelvärde (av betygen)
2. $t^\star$ - tidigare har vi använt 1.96 här för att så ett intervall som omattar 95%. Detta räknar dock inte med den extra osäkerhet uppstår på grund av att standardavvikelsen (från populationen) och standardfelet (beräknat från stickprovet) ofta skiljer sig åt lite. För att kompensera för detta använder man något som heter *Student's t-distribution*. Ju mindre stickprov, desto större osäkerhet. Här är en grafisk jämförelse mellan normalfördelningen och t-fördelningen, för stickprovsstorlek fem.

Vi kan få värden på $t^\star$ från R med `qt()`

```
qt(.975, df = 100) # 97.5:e percentilen, vilket ju används för 95% k.i.
```

```
[1] 1.983972
```



Det är ofta en ganska liten skillnad mellan normalfördelningen och t-fördelningen.

3. **Standardfelet SE**. För andelar har vi använt standardfelet

$$\text{SE} = \sqrt{\frac{\hat{p}(1-\hat{p})}{n}}$$

Nu kommer vi istället att använda stickprovets standardavvikelse delat på roten ur stickprovsstorleken.

$$\text{SE} = \frac{\sigma}{\sqrt{n}}$$

Detta blir sammantaget

$$\bar{x} \pm t^\star \cdot \frac{\sigma}{\sqrt{n}}$$

I denna lab kommer vi dock att använda en i R inbyggd funktion som automatiskt beräknar samtliga delar av denna formel: `t.test()`.

Först ett exempel: vi kastar en vanlig tärning 1000 gånger och vill beräkna ett konfidensintervall för medelvärdet (det borde ju vara mellan 3 och 4, helst 3.5).

```
set.seed(17) # detta gör så att vi jag får samma slumptal varje gång
# jag renderar labsidan
kast <- sample(1:6, 1000, replace = TRUE)
t.test(kast)
```

```
    One Sample t-test

data:  kast
t = 65.406, df = 999, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 3.420211 3.631789
sample estimates:
mean of x
    3.526
```

Som du ser är det lätt att få ett konfidensintervall från `t.test()`

> **Uppgift 2.3**
>
> Beräkna ett konfidensintervall för filmbetygen med `t.test()`. Kan vi anta normalfördelning? Jämför med bootstrapintervallet.

Nu är vi nyfikna på om filmer vars språk inte är engelska får bättre eller sämre betyg än engelskspråkiga filmer.

Vi börjar med att skilja på engelskspråkiga och icke-engelskspråkiga filmer.

```
table(movie_sample$language)
```

```
da de en es fr hi it ja ko no ta te th tl zh
 1  2 62  7  3  4  1  7  3  1  2  2  2  1  2
```

```
en_language <- movie_sample$rating[movie_sample$language=="en"]
print("Engelska")
```

```
[1] "Engelska"
```

```
mean(en_language)
```

```
[1] 5.949661
```

```
length(en_language) # antal med engelskt språk
```

```
[1] 62
```

```
foreign_language <- movie_sample$rating[movie_sample$language!="en"]
print("Andra språk")
```

```
[1] "Andra språk"
```

```
mean(foreign_language)
```

```
[1] 5.721421
```

```
length(foreign_language) # antal med icke-engelskt språk
```

```
[1] 38
```

Vi ser en skillnad i vårt stickprov, men kan dra några slutsatser från detta? Vi skapar ett konfidensintervall för skilnaden i medelvärde, $\mu_{en} - \mu_{foreign}$

```
t.test(en_language, foreign_language)
```

```
    Welch Two Sample t-test

data:  en_language and foreign_language
t = 0.51974, df = 52.357, p-value = 0.6054
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.6528198  1.1093002
sample estimates:
mean of x mean of y
 5.949661  5.721421
```

Uppgift 2.4

Tolka output från testet. Vad kan vi säga om skillnaden? Är det statistiskt säkerställt att engelskspråkiga filmer får högre betyg i genomsnitt?