Forelesning 8: Inferens

Sok-2009 h24

Eirik Eriksen Heen & ChatGPT

Invalid Date

Statistisk inferens, eller induksjon, er prosessen hvor vi trekker konklusjoner om en populasjon basert på data fra et utvalg. Dette kan inkludere testing av hypoteser, estimering av parametere, og bygging av konfidensintervaller. Eksempler på bruk av statistisk inferens inkluderer å vurdere effektiviteten av en ny vaksine eller å avgjøre om en bestemt behandling har en effekt på en populasjon.

Hypotesetesting

En sentral del av inferens er hypotesetesting, hvor vi vurderer en nullhypotese (H_0) mot et alternativ (H_1). For eksempel, ved testing av en vaksine kan vi sette opp følgende hypoteser:

H_0: Vaksinen har ingen effekt.

H_1: Vaksinen har en effekt.

Ved hypotesetesting antar vi at H_0 er sann og vurderer sannsynligheten for å observere dataene vi har samlet inn under denne antagelsen. Hvis denne sannsynligheten er lav nok, forkaster vi H_0 til fordel for H_1.

Nummersik eksempel

Når vi utfører en hypotesetest, starter vi med et spørsmål som vi ønsker å undersøke vitenskapelig. I dette eksempelet vil vi undersøke om gjennomsnittshøyden til personer som er 20 år eller eldre i USA er høyere enn 169 cm. Hypotesetesting gir oss en systematisk måte å trekke konklusjoner basert på data, og gir en objektiv vurdering av om våre observasjoner kan skyldes tilfeldig variasjon eller om de er statistisk signifikante.

Vi bruker data fra NHANES-gruppen, som inneholder informasjon om høyder og annen helseinformasjon for et representativt utvalg av den amerikanske befolkningen.

Sette opp hypotesen

Hypotesetesting innebærer alltid å formulere to konkurrerende hypoteser:

- **Nullhypotese (H_0)**: Dette er antagelsen om at det ikke finnes noen effekt eller forskjell. For vårt tilfelle vil nullhypotesen være at gjennomsnittshøyden for personer som er 20 år eller eldre er 169 cm eller lavere.

  H0: $\mu \geq 169$

- **Alternativ hypotese (H_1)**: Dette er antagelsen som vi ønsker å teste mot nullhypotesen. I dette eksempelet vil den alternative hypotesen være at gjennomsnittshøyden for personer over 20 år er høyere enn 169 cm.

  H1: $\mu < 169$

Det er viktig å forstå at vi gjennom en hypotesetest ikke beviser en hypotese som "sann". I stedet undersøker vi hvorvidt vi har tilstrekkelig bevis til å avvise nullhypotesen, altså om dataene våre er i strid med antagelsen om at gjennomsnittshøyden er 169 cm eller høyere.

Dataanalyse La oss starte med å beregne gjennomsnittshøyden i vårt utvalg for personer som er 20 år eller eldre:

```
NHANES %>%
    # Bruker kun personer som er 20 år eller eldre
    filter(Age >= 20 & !is.na(Height) ) %>%
  summarize(gjenn_hoyde = mean(Height))
```

```
# A tibble: 1 x 1
  gjenn_hoyde
        <dbl>
1        169.
```

Dette gir oss en oversikt over hva den faktiske gjennomsnittshøyden er i vårt utvalg. Anta at gjennomsnittshøyden i vårt utvalg er 168,8 cm, noe som er nær 169 cm.

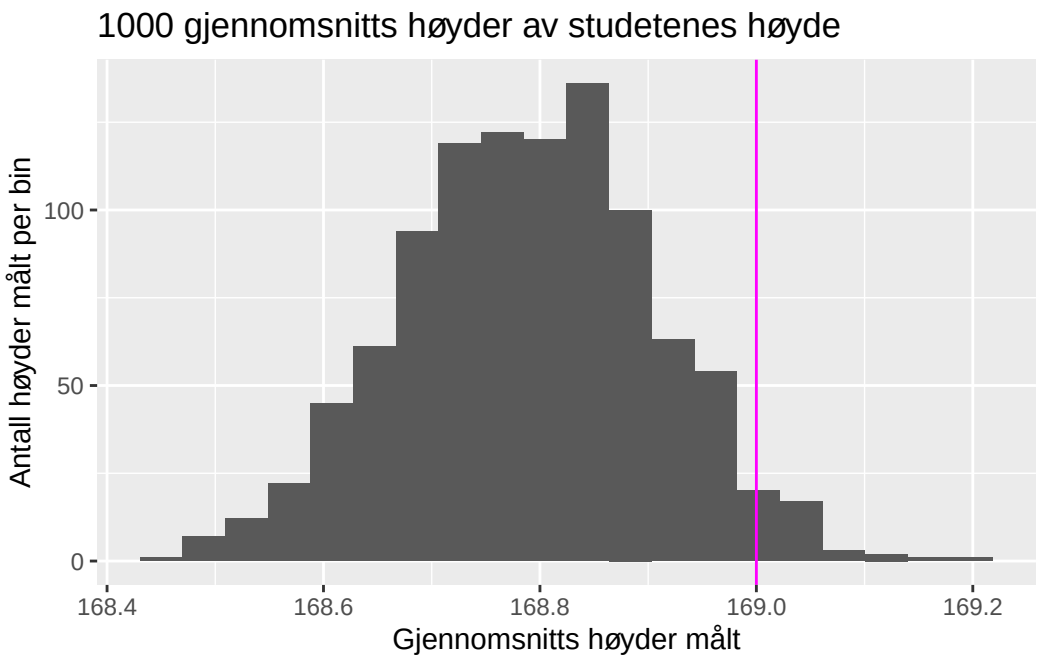
Simulering for å undersøke naturlig variasjon For å forstå hvor stor variasjon vi kan forvente fra utvalget vårt, vil vi utføre en bootstrap-simulering. Dette innebærer at vi trekker et stort antall nye utvalg (med tilbakelegging) fra vårt opprinnelige datasett, og beregner gjennomsnittet for hvert av disse utvalgene. På denne måten kan vi visualisere den naturlige variasjonen i gjennomsnittshøyden.

Vi trekker 1000 utvalg fra vårt datasett og beregner gjennomsnittshøyden for hver trekning:

```r
# Vi replikerer
Heights <- replicate(
  # tusen ganger
  n = 1000,
  # hetnter ut datasettet
  NHANES %>%
    # Bruker kun personer som er 20 år eller eldre
    filter(Age >= 20 & !is.na(Height) ) %>%
    # trekker tilfelig med tilbakelegg
    slice_sample(prop = 1, replace = TRUE) %>%
    summarize(gjenn_hoyde = mean(Height)) %>%
    pull(gjenn_hoyde)
)


Heights <- tibble(Heights)
```

Visualisering av gjennomsnittshøyder Deretter lager vi et histogram for å se fordelingen av gjennomsnittshøydene fra våre 1000 trekninger, og vi markerer gjennomsnittet på 169 cm for å se hvor ofte vi observerer verdier som er høyere eller lavere enn dette:

```r
# Plotter alle gjennomsnittene
Heights %>% ggplot(aes(x=Heights)) +
  geom_histogram(bins = 20) +
  geom_vline(xintercept = 169, color = "magenta") +
  xlab("Gjennomsnitts høyder målt") +
  ylab("Antall høyder målt per bin") +
  ggtitle("1000 gjennomsnitts høyder av studetenes høyde")
```



Her kan vi se fordelingen av gjennomsnittshøyder. Dersom veldig mange av gjennomsnittene faller under 169 cm, kan dette gi oss en indikasjon på at det er lite sannsynlig at den faktiske gjennomsnittshøyden er høyere enn 169 cm.

Resultater og konklusjon La oss nå beregne hvor mange av våre gjennomsnittshøyder som faktisk overstiger 169 cm:

```
Heights %>%
  summarise(over_169 = mean(Heights>=169))
```

```
# A tibble: 1 x 1
  over_169
     <dbl>
1    0.035
```

Fra resultatet ser vi at kun 3,2 % av våre simuleringer resulterte i gjennomsnitt som var høyere enn 169 cm. Dette betyr at det er veldig usannsynlig at vi, kun basert på naturlig variasjon, vil observere en gjennomsnittshøyde som overstiger 169 cm. Dermed har vi grunn til å tvile på nullhypotesen, og vi kan konkludere med at det er lite sannsynlig at den sanne gjennomsnittshøyden er 169 cm eller høyere. Vi har dermed støtte for å akseptere alternativhypotesen om at gjennomsnittshøyden er lavere enn 169 cm.

Viktig! Når vi gjennomfører en hyppotesetest antar vi at nullhypotesen er korrekt. Dette har vi IKKE gjort i eksemplet over. Vi har kun sett på den naturlig variasjonen i utvalget.

Hvorfor bruker vi et 5% signifikansnivå?

Når vi gjennomfører statistiske tester, setter vi ofte signifikansnivået til 5%, altså en sannsynlighet på 0,05 for å forkaste en sann nullhypotese. Men hvorfor akkurat 5%? Denne praksisen kan spores tilbake til Ronald A. Fisher, en av grunnleggerne av moderne statistikk, som foreslo 5% som en konvensjonell grense for å avgjøre om et resultat er statistisk signifikant. Fisher mente at et 5% nivå var en praktisk balanse mellom å unngå falske positive (forkaste en sann nullhypotese) og å være sensitiv nok til å oppdage sanne effekter.

For å forstå dette intuitivt, kan vi sammenligne det med å kaste et pengestykke. Hvor mange Kron på rad må du kaste før du starter å bli mistenksom på om dette pengestykke er rettferdig? Når hvis det er blir 4 eller 5 på rad så blir man kanskje litt mistenksom. Hva er sannsynligheten for dette?

Første kron:

```
pbinom(0, size = 1, prob=0.5)
```

```
[1] 0.5
```

Veldig sannsynlig

Andre kron:

```
pbinom(0, size = 2, prob=0.5)
```

```
[1] 0.25
```

Tredje kron:

```
pbinom(0, size = 3, prob=0.5)
```

```
[1] 0.125
```

Fjerde kron:

```
pbinom(0, size = 4, prob=0.5)
```

```
[1] 0.0625
```

Femte kron:

```r
pbinom(0, size = 5, prob=0.5)
```

```
[1] 0.03125
```

Vi ser at rundt fire til fem kron på rad starter stannsynlighetn å bli litt lav. Dette man kan starte å mistenke at pengestykke er manipulert.

I praksis, når vi setter signifikansnivået til 5%, sier vi at vi er villige til å akseptere en 5% sjanse for at en tilsynelatende effekt er et resultat av tilfeldigheter. Dette nivået er ikke absolutt og kan justeres avhengig av konteksten, men 5% har blitt en standard som gir en fornuftig avveining mellom følsomhet og spesifisitet i mange anvendelser.

Konfidensintervall

Så langt har vi fokusert på å avgjøre om det finnes en forskjell mellom to grupper eller variabler. Nå skal vi i stedet rette oppmerksomheten mot størrelsen på denne forskjellen, altså hvor mye de to gruppene skiller seg fra hverandre i snitt. For å gjøre dette bruker vi konfidensintervaller, som gir oss et estimat på hvor den sanne middelverdien for en populasjon sannsynligvis befinner seg, basert på et gitt utvalg.

Et konfidensintervall (KI) gir oss et spenn som vi kan være rimelig sikre på inneholder den sanne verdien av en parameter, for eksempel gjennomsnittet. Jo bredere konfidensintervallet er, desto mer usikkerhet er det knyttet til estimatet.

Eksempel

Når vi beregner et konfidensintervall for gjennomsnittet, ønsker vi å finne et intervall som med en viss sikkerhet inneholder den sanne gjennomsnittsprisen for alle boliger i populasjonen, basert på vårt utvalg. Dette intervallet gir oss en forståelse av hvor presist vi har estimert gjennomsnittsprisen og hvor mye usikkerhet som er knyttet til dette estimatet. Jo smalere konfidensintervallet er, desto sikrere kan vi være på at gjennomsnittsprisen vi har beregnet er nær den sanne verdien for hele populasjonen.

Beregning av gjennomsnitt, standardavvik og standardfeil

Vi starter med å få litt oversikt ved å beregne gjennomsnittsprisen, standardavviket (som viser variasjonen i prisene), og standardfeilen (som viser hvor mye gjennomsnittet vårt kan forventes å variere).

```r
mean_price <- house_prices %>%
  # filtrerer vekk alle prisene som er na. dette er pga det kan påvirke n() funksjonen
  filter(!is.na(price)) %>%
  summarize(
    gjenn_pris = mean(price, na.rm = TRUE),
    sd = sd(price, na.rm = TRUE),
    sd_error = sd(price, na.rm = TRUE)/ sqrt(n() ) )

mean_price
```

```
# A tibble: 1 x 3
  gjenn_pris      sd sd_error
       <dbl>   <dbl>    <dbl>
1    540088. 367127.    2497.
```

Gjennomsnittsprisen forteller oss hva den typiske prisen på en bolig er i vårt utvalg. I dette eksempelet antar vi at gjennomsnittsprisen er 540 088 kroner.

Standardavviket gir oss et mål på hvor mye de enkelte boligprisene i datasettet avviker fra gjennomsnittet. Et høyt standardavvik betyr at prisene varierer mye, mens et lavt standardavvik betyr at de fleste priser ligger nær gjennomsnittet.

Standardfeilen (standard error) beregnes som standardavviket delt på kvadratroten av antallet observasjoner (boliger i vårt utvalg). Standardfeilen gir oss en idé om hvor nøyaktig gjennomsnittet vårt er som en representasjon for hele populasjonen. I dette tilfellet er standardfeilen på 2 497, noe som antyder at variasjonen i gjennomsnittet vårt ikke er veldig stor, og at vi har en relativt høy presisjon i estimatet.

Hurtigmetode for beregning av konfidensintervall

Hvis vi ønsker en rask måte å beregne konfidensintervallet på, kan vi bruke en t-test. Selv om vi egentlig ikke er interessert i å teste om prisen er forskjellig fra null, kan vi utnytte t-testens evne til å beregne konfidensintervaller for å få resultatet raskt.

```r
# gjennomfører en t-test om prisen er forskjellig fra null og lagrer objektet
t_test_resultat <- t.test(house_prices$price,
                          # Spesifiererer konfidensintervallet vi ønsker, 95% er standard. M
                          conf.level = 0.95)
# Ved å legge til $conf.int får vi ut konfidensitnervallet til testne
t_test_resultat$conf.int
```

```
[1] 535193 544983
attr(,"conf.level")
[1] 0.95
```

Dette gir oss konfidensintervallet direkte fra t-testen. For eksempel kan vi få at konfidensintervallet for gjennomsnittsprisen er mellom 535 193 og 544 983 kroner. Dette betyr at vi med 95 % sikkerhet kan si at den sanne gjennomsnittsprisen på boliger i populasjonen ligger et sted mellom disse to verdiene, basert på dataene i vårt utvalg.

Forklaring av resultatet

Konfidensintervallet gir oss et spenn som vi kan bruke til å vurdere hvor stor usikkerhet det er knyttet til estimatet vårt. I dette tilfellet indikerer et relativt smalt konfidensintervall (535 193 til 544 983 kroner) at vi har høy tillit til at den sanne gjennomsnittsprisen i populasjonen ligger innenfor dette området. Hvis konfidensintervallet hadde vært bredere, ville det indikert større usikkerhet, og vi ville hatt mindre presisjon i estimatet vårt.

Det er også viktig å merke seg at konfidensintervallet avhenger av både størrelsen på utvalget vårt og variasjonen i prisene. Et større utvalg eller lavere variasjon ville resultert i et smalere konfidensintervall, mens et mindre utvalg eller større variasjon ville gitt et bredere konfidensintervall.

Bootstrapping

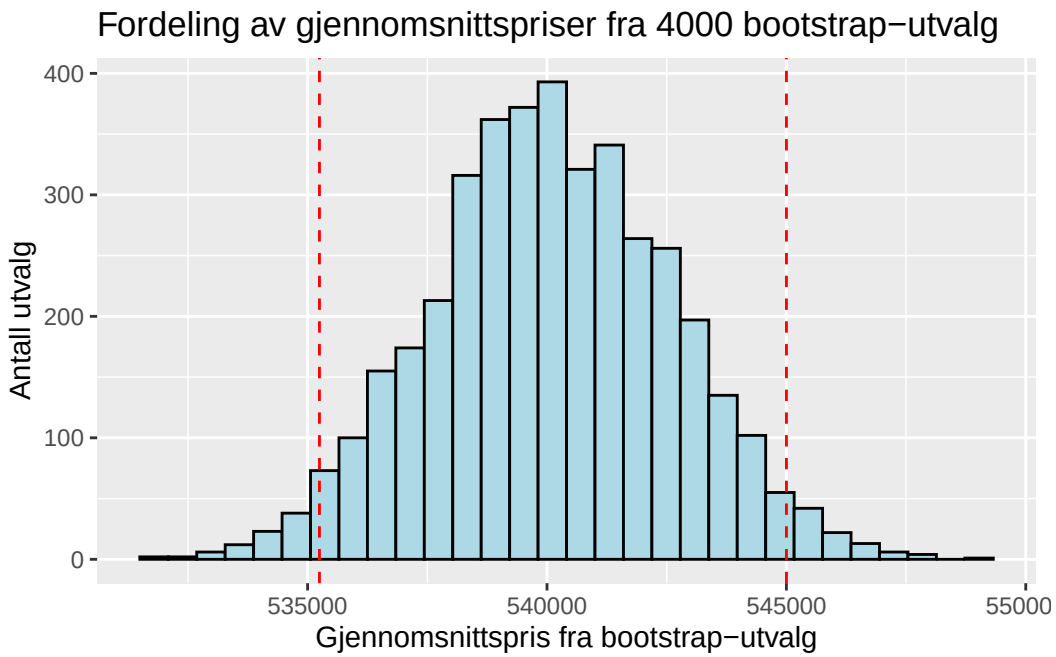
En annen metode for å beregne konfidensintervaller er bootstrapping. I motsetning til den vanlige metoden der vi antar at dataene følger en kjent fordeling (som normalfordeling), baserer bootstrapping seg på å trekke tilfeldige utvalg fra datasettet med tilbakelegging for å skape nye, simulerte utvalg. Dette lar oss numerisk beregne fordelingen av gjennomsnittet i disse simulerte utvalgene og deretter konstruere konfidensintervaller basert på denne fordelingen.

Bootstrapping er spesielt nyttig når vi har små utvalg eller når vi ikke kan anta at dataene våre følger en spesifikk fordeling. Ved å bruke denne metoden kan vi skape en empirisk fordeling av gjennomsnittet og dermed beregne et konfidensintervall uten å gjøre antagelser om fordelingen av dataene.

Fremgangsmåte Fra det opprinnelige utvalget vi har, trekker vi tilfeldige boligpriser med tilbakelegging og beregner gjennomsnittet av hver av disse trekningene. Dette gjentar vi et stort antall ganger, og til slutt beregner vi percentilene for gjennomsnittene for å finne et konfidensintervall.

Her er hvordan vi kan gjøre det i R:

```r
set.seed(200)

# Trekker 4000 tilfeldige utvalg med tilbakelegging og beregner gjennomsnittet for hvert
data_priser <- tibble(gjenn_pris = replicate(n = 4000,
                                             house_prices %>%
                                               slice_sample(prop = 1, replace = TRUE) %>%
                                               summarise(gjenn_pris = mean(price, na.rm = TRU
                                               pull(gjenn_pris)
                                               )
)

# Visualisering av fordelingen av gjennomsnittspriser fra bootstrap-utvalgene
ggplot(data_priser, aes(x = gjenn_pris)) +
  geom_histogram(bins = 30, fill = "lightblue", color = "black") +
  geom_vline(xintercept = quantile(data_priser$gjenn_pris, 0.025), color = "red", linetype =
  geom_vline(xintercept = quantile(data_priser$gjenn_pris, 0.975), color = "red", linetype =
  xlab("Gjennomsnittspris fra bootstrap-utvalg") +
  ylab("Antall utvalg") +
  ggtitle("Fordeling av gjennomsnittspriser fra 4000 bootstrap-utvalg")
```



I dette eksempelet trekker vi 4000 tilfeldige utvalg fra vårt datasett (med tilbakelegging) og beregner gjennomsnittet av prisene for hvert av disse utvalgene. Vi plotter deretter fordelingen av disse gjennomsnittsprisene for å se hvor de fleste av utvalgene befinner seg.

Beregning av konfidensintervallet Når vi har samlet alle de bootstrappede gjennomsnittene, kan vi beregne 95 % konfidensintervallet ved å finne 2,5-prosentilen og 97,5-prosentilen av disse gjennomsnittene. Dette gir oss det området hvor 95 % av bootstrap-utvalgene befinner seg.

```
# Beregner 2.5% og 97.5% kvantilene for å finne konfidensintervallet
data_priser %>%
  summarise(nedre_grense = quantile(gjenn_pris, 0.025),
            ovre_grense = quantile(gjenn_pris, 0.975))
```

```
# A tibble: 1 x 2
  nedre_grense ovre_grense
         <dbl>       <dbl>
1      535247.     545001.
```

Dette gir oss et konfidensintervall basert på fordelingen av gjennomsnittene fra de 4000 bootstrap-utvalgene.

	Øvre grense	Nedre grense
Matematisk	535 193	544 983
Numersik	535 247	545 001

Tolkning Selv om vi ikke får nøyaktig samme resultat som ved bruk av den vanlige metoden for beregning av konfidensintervaller (som bygger på fordelingsteori), er forskjellen vanligvis liten. Dette skyldes at bootstrapping-metoden gir oss en empirisk fordeling basert på dataene våre, mens den vanlige metoden gjør visse antagelser om normalitet og varians.

Bootstrapping gir oss et robust alternativ når vi ønsker å unngå slike antagelser eller når vi arbeider med data som ikke nødvendigvis følger en normalfordeling. I tillegg er denne metoden svært fleksibel og kan brukes på mange ulike typer data, noe som gjør den spesielt nyttig i mer komplekse situasjoner.

Matematisk: Eksempel på beregning av konfidensintervall for gjennomsnittsprisen på boliger (t-intervall)

Det er viktig at dere vet om hvordan man regner ut konfidensintervallet.

Når vi beregner et konfidensintervall for gjennomsnittet ved bruk av et t-intervall, tar vi høyde for at vi estimerer standardavviket fra utvalget vårt, og ikke kjenner populasjonens standardavvik. Dette er spesielt nyttig når vi jobber med små utvalg. Et t-intervall tar hensyn til usikkerheten ved å bruke utvalgets standardavvik, i motsetning til et z-intervall som antar at vi kjenner populasjonsstandardavviket.

Oversikt over de nødvendige elementene

Vi begynner med å beregne gjennomsnittsprisen, standardavviket, og standardfeilen for prisene på boliger i vårt utvalg. Standardfeilen er viktig fordi den forteller oss hvor mye gjennomsnittsprisen i vårt utvalg forventes å avvike fra den sanne gjennomsnittsprisen i populasjonen.

Oversikt over de nødvendige elementene

For å beregne et t-konfidensintervall manuelt, trenger vi følgende informasjon:

- Gjennomsnittet av boligprisene i utvalget, $\bar{x}$
- Standardavviket i utvalget, s
- Antall observasjoner i utvalget, n
- Kritisk t-verdi for det ønskede konfidensnivået og antall frihetsgrader, t^*
- Standardfeilen til gjennomsnittet, $SE = \frac{s}{\sqrt{n}}$

Formelen for t-konfidensintervallet er:

$$\text{Konfidensintervall} = \bar{x} \pm t^* \cdot SE$$

```
mean_price <- house_prices %>%
  summarize(
    gjenn_pris = mean(price, na.rm = TRUE),
    sd = sd(price, na.rm = TRUE),
    n = n(),
    sd_error = sd(price, na.rm = TRUE)/ sqrt(n() )
  )

mean_price
```

```
# A tibble: 1 x 4
  gjenn_pris      sd     n sd_error
       <dbl>   <dbl> <int>    <dbl>
1    540088. 367127. 21613    2497.
```

Finn den kritiske t-verdien (t^*)

Den kritiske t-verdien t∗t^*t∗ avhenger av konfidensnivået vi ønsker, og antall frihetsgrader ($df = n - 1$). For et 95 % konfidensnivå og $n - 1$ frihetsgrader, kan vi bruke R til å finne den kritiske t-verdien ved hjelp av `qt`-funksjonen:

```
# Finn den kritiske t-verdien for et 95 % konfidensintervall
# df = n - 1
t_value <- qt(0.975, df = mean_price$n - 1)
t_value
```

```
[1] 1.96
```

For eksempel, hvis vi har 100 observasjoner i utvalget, vil frihetsgradene være $df = 99$. Da får vi en kritisk t-verdi på omtrent 1.984 for et 95 % konfidensnivå.

Beregne konfidensintervall mauelt

```
lower_bound <- mean_price$gjenn_pris - t_value * mean_price$sd_error
upper_bound <- mean_price$gjenn_pris + t_value * mean_price$sd_error

cat("95% Konfidensintervall:", lower_bound, "til", upper_bound)
```

```
95% Konfidensintervall: 535193 til 544983
```

Dette vil gi oss et konfidensintervall som for eksempel som er [531 193, 544 983], i house_prices. Dette ser vi er det sammen som over.

Oppsummering

Her ser vi at de viktigste delene i et konfidensintervall er at intervallet er sentrert rundt gjennomsnittet. Konfidensintervallet reflekterer usikkerheten knyttet til estimeringen av gjennomsnittet i populasjonen, og bredde på intervallet påvirkes av flere faktorer:

1. **Variabilitet i dataene**: Hvis det er stor naturlig variasjon i utvalget (høy standardavvik), vil konfidensintervallet bli bredere. Dette skjer fordi det er vanskeligere å estimere gjennomsnittet presist når dataene varierer mye.
2. **Størrelsen på utvalget**: Når antall observasjoner ((n)) øker, vil standardfeilen (og dermed konfidensintervallet) bli mindre. Flere data gir et mer presist estimat av gjennomsnittet, og vi blir dermed mer sikre på hvor gjennomsnittet faktisk ligger.

3. **Sikkerhetsnivået (t-verdien)**: Størrelsen på t-verdien, som avhenger av hvor sikre vi ønsker å være på at intervallet inneholder det sanne gjennomsnittet, påvirker også intervallets bredde. Jo høyere sikkerhet vi krever (f.eks. 99 % i stedet for 95 %), desto større blir konfidensintervallet, fordi vi må utvide intervallet for å fange opp mer usikkerhet.

Med andre ord, et høyt standardavvik betyr mindre presisjon, et større utvalg gir økt presisjon, og et høyere sikkerhetsnivå (t-verdien) gir bredere intervaller for å reflektere den økte sikkerheten. Konfidensintervallet gir oss dermed en måte å kvantifisere usikkerheten i våre estimater og vise hvor presise de er basert på både data og ønsket sikkerhetsnivå.

Eksempel: Boligpriser med og uten havutsikt

La oss fortsette med eksempelet vi så på tidligere, der vi undersøkte forskjellen i pris på boliger som har havutsikt sammenlignet med de som ikke har det. Vi begynner med å se på gjennomsnittsprisene for begge grupper:

```
house_prices %>%
  group_by(waterfront) %>%
  summarise(gjen_pris=mean(price, na.rm = TRUE))
```

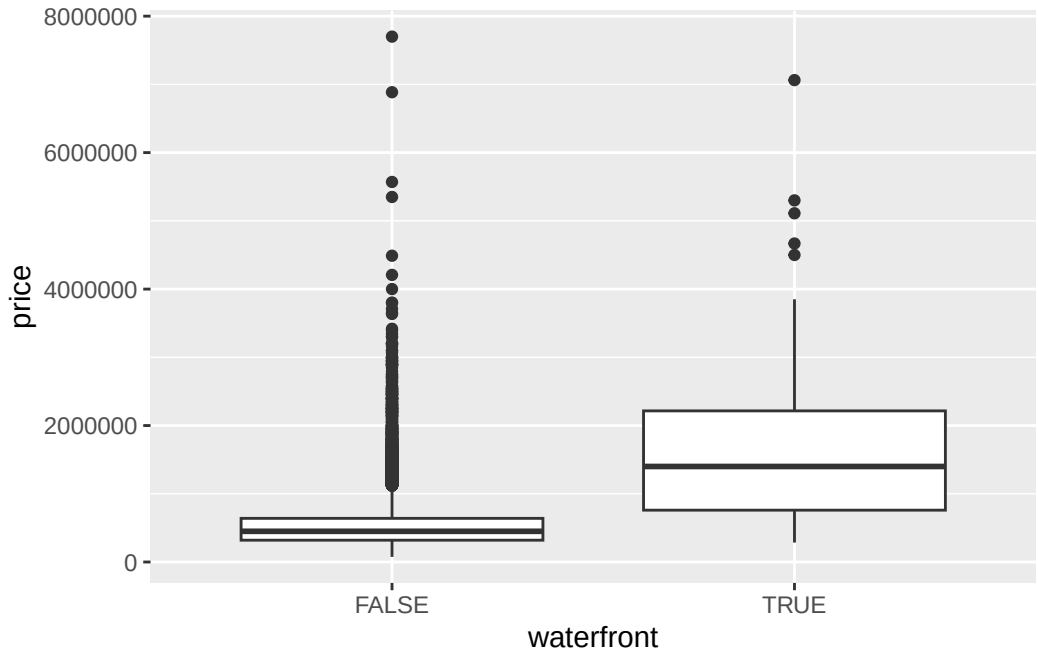
```
# A tibble: 2 x 2
  waterfront gjen_pris
  <lgl>          <dbl>
1 FALSE        531564.
2 TRUE        1661876.
```

Her ser vi at boliger med havutsikt i gjennomsnitt koster omtrent tre ganger så mye som de uten havutsikt. Dette gir oss et tydelig bilde av at det er en betydelig forskjell mellom de to gruppene, men det er også viktig å undersøke hvordan prisfordelingen ser ut.

Visualisering av fordelingen

For å få en bedre forståelse av fordelingen i prisene for disse to gruppene, kan vi bruke et boxplot som viser variasjonen i boligprisene innenfor hver gruppe:

```
house_prices %>%
  ggplot(aes(x=waterfront, y=price))+
  geom_boxplot()
```



Boxplottet viser at det er en betydelig forskjell i prisene mellom boliger med og uten havutsikt. For eksempel ser vi at prisspredningen for boliger med havutsikt er større, noe som indikerer at prisene varierer mye mer i denne gruppen enn for boliger uten havutsikt. Dette gir oss et visuelt inntrykk av forskjellen mellom de to gruppene, men for å være mer presise kan vi gjennomføre en statistisk test for å undersøke om forskjellen er signifikant.

T-test for prisforskjell

En t-test lar oss vurdere om forskjellen i gjennomsnittspris mellom de to gruppene er signifikant eller om den kan tilskrives tilfeldigheter. Vi utfører en t-test på boligprisene med og uten havutsikt:

```
#gjennomfører en t test og lagrer resultatet
t_test_resultat <- t.test(
  # tester pris som blir del in i gruppene med havutsikt eller ikke
  price ~ waterfront,
  # henter dataen fra datasettet
  data=house_prices)

# Printer reslutatet
t_test_resultat
```

```
    Welch Two Sample t-test

data:  price by waterfront
t = -13, df = 162, p-value <2e-16
alternative hypothesis: true difference in means between group FALSE and group TRUE is not e
95 percent confidence interval:
 -1303662  -956963
sample estimates:
mean in group FALSE  mean in group TRUE
             531564             1661876
```

Resultatene fra t-testen viser at vi kan forkaste nullhypotesen om at det ikke er noen forskjell i gjennomsnittlig pris mellom de to gruppene. Dette betyr at det er svært usannsynlig at forskjellen vi observerer skyldes tilfeldigheter. Med andre ord kan vi konkludere med at boliger med havutsikt generelt sett er dyrere enn de uten.

Beregning av konfidensintervaller

Selv om t-testen viser at det er en signifikant forskjell, gir den oss ikke informasjon om størrelsen på forskjellen med et visst sikkerhetsnivå. For å få et klarere bilde av dette, kan vi beregne konfidensintervallene for gjennomsnittsprisene i de to gruppene. Et 95 % konfidensintervall betyr at vi med 95 % sikkerhet kan si at den sanne gjennomsnittsprisen for en gruppe ligger innenfor det intervallet vi beregner. Dette kan vi gjøre ved hjelp av t-testens utfall eller manuelt:

```r
# Henter ut konfidensi intervallet fra t-testen
# Vi legger til $conf.int
t_test_resultat$conf.int
```

```
[1] -1303662  -956963
attr(,"conf.level")
[1] 0.95
```

Konfidensintervallet gir oss informasjon om den estimerte forskjellen i gjennomsnittspris mellom de to gruppene, sammen med en usikkerhet knyttet til denne forskjellen. For eksempel kan konfidensintervallet for prisforskjellen være noe slikt som 1 million til 3 millioner kroner, noe som betyr at vi med 95 % sikkerhet kan si at prisforskjellen mellom boliger med og uten havutsikt ligger et sted mellom disse to verdiene.

Tolkning av konfidensintervallet

Et konfidensintervall kan tolkes som et mål på hvor presist vi har klart å estimere gjennomsnittsforskjellen. Et bredt konfidensintervall indikerer større usikkerhet rundt estimatet, mens et smalere konfidensintervall betyr at vi har større sikkerhet om at gjennomsnittsforskjellen ligger nærmere den estimerte verdien. Dersom konfidensintervallet ikke overlapper null, betyr det også at forskjellen er signifikant, noe som samsvarer med resultatet fra t-testen.

For vårt tilfelle kan vi derfor konkludere med at det er en signifikant forskjell i pris mellom boliger med og uten havutsikt, og vi kan bruke konfidensintervallet til å gi et estimat på hvor stor denne forskjellen er.

Oppgave

1. Hva forteller et konfidensintervall oss? For eksempel ta for deg det 95% konfidensintervall.
2. Hvilken intervall er beredere/større (ceteris paribus), et 95% konfidensintervall eller et 99% konfidensintervall?
    1. Hvorfor er det forskjell?
3. Regn ut et konfidensintervall av hus priser for hus med oversikt over vannet, og de uten.
    1. Hvis du sammen likner dem. Hvilken er størst og hvorfor tror du det er slik?
4.

Introduksjon til «There is only one test» fra `infer`-pakken

I statistikk er inferensprosessen avgjørende for å trekke konklusjoner om en populasjon basert på et utvalg. En viktig del av inferens er hypotesetesting, som lar oss evaluere om det er en statistisk signifikant forskjell mellom grupper eller om en observasjon er resultat av tilfeldig variasjon. Tradisjonelt sett har hypotesetesting vært basert på analytiske metoder som t-tester og z-tester. Men i nyere tid har simuleringsbaserte metoder som permutasjonstester blitt mer populære, særlig når forutsetningene for de klassiske testene ikke er oppfylt.

En slik simuleringsbasert test er «There is only one test», tilgjengelig gjennom `infer`-pakken i R. Denne metoden bruker permutasjonstesting for å teste hypoteser, noe som gjør den svært fleksibel og nyttig når vi ikke kan anta normalfordeling eller når vi jobber med ikke-parametriske data.

Hva er «There is only one test»?

«There is only one test» i `infer`-pakken er en permutasjonstest som brukes til å teste forskjeller mellom grupper ved å permutere observasjonene i datasettet. Permutasjonstester bygger på ideen om å omfordele dataene mange ganger for å beregne en empirisk fordeling av teststatistikken under nullhypotesen. Dette gir oss et alternativ til klassiske tester som t-tester, spesielt når vi ikke kan anta normalfordeling eller homogen varians.

Hvorfor permutasjonstester?

Permutasjonstesting er en ikke-parametrisk metode som ikke gjør antagelser om fordelingen av dataene. Den lar oss beregne p-verdier og konfidensintervaller ved å sammenligne den observerte teststatistikken med en fordeling som genereres ved tilfeldig permutasjon. Dette gir oss en robust tilnærming for å trekke slutninger, selv når forutsetningene for vanlige tester ikke er oppfylt.

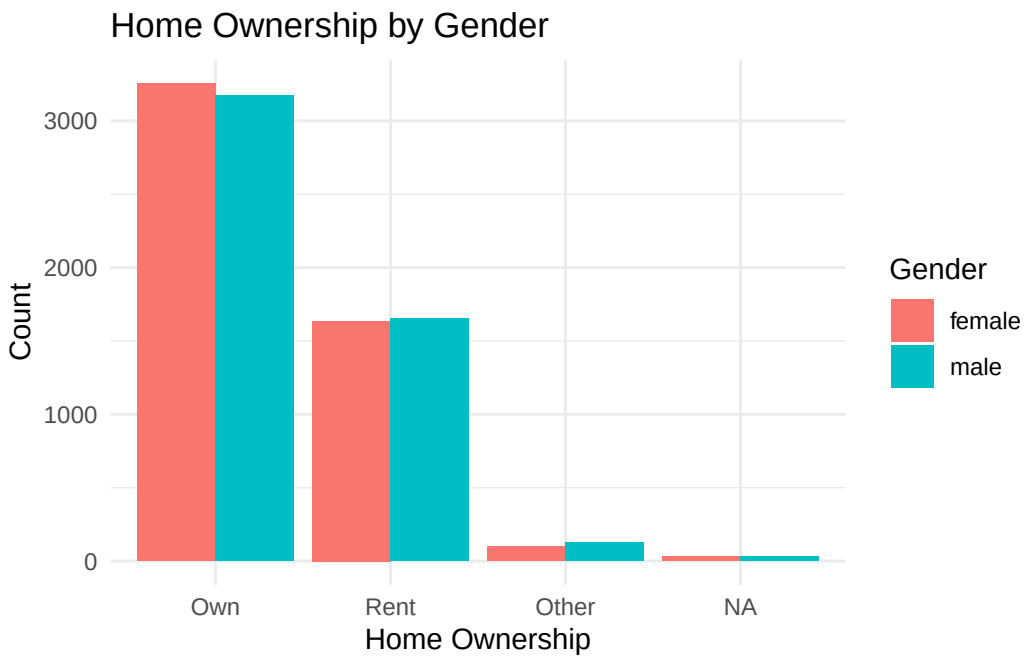
Fordelene med permutasjonstester inkluderer: - **Ingen strenge forutsetninger:** Vi trenger ikke å anta normalfordeling eller lik varians mellom grupper. - **Simuleringsbasert:** Ved å bruke simulerte fordelingen av teststatistikken, kan vi få mer realistiske p-verdier for små eller ikke-parametriske datasett. - **Fleksibilitet:** Denne tilnærmingen kan brukes på et bredt spekter av hypotesetester, inkludert tester av proporsjoner, differanser i middelverdier, eller korrelasjoner.

Hvordan virker «There is only one test»?

Vi kan illustrere «There is only one test» med et eksempel på forskjellen i andelen boligeiere mellom menn og kvinner. Målet her er å undersøke om det er en signifikant forskjell i andelen som eier bolig mellom kjønnene. For å gjøre dette bruker vi permutasjonstesting til å omfordele dataene tilfeldig for å simulere hva som ville skjedd dersom nullhypotesen var sann (at det ikke er noen forskjell mellom kjønnene).

Først visualiserer vi dataen.

```
NHANES %>%
  ggplot(aes(x = HomeOwn, fill = Gender)) +
  geom_bar(position = "dodge") +
  labs(title = "Home Ownership by Gender",
       x = "Home Ownership",
       y = "Count") +
  theme_minimal()
```



Det ser ut til at det er litt forskjell mellom menn og kvinner, flere kvinner eier og flere menn leier. Men Er det noe forskjell mellom gruppene? Med denne metoden kan vi KUN teste to grupper mot hverandre, så vi må filtrere vekk *Other* og *NA*.

```
homes <- NHANES %>%
  filter(HomeOwn %in% c("Rent", "Own") )
```

Hva er forskjellen i rate mellom gruppene?

```
homes %>%
  group_by(Gender) %>%
  summarise(Rate_Own = mean(HomeOwn == "Own"))
```

```
# A tibble: 2 x 2
  Gender Rate_Own
  <fct>     <dbl>
1 female    0.665
2 male      0.658
```

```
diff <- homes %>%
  group_by(Gender) %>%
  summarise(Rate_Own = mean(HomeOwn == "Own")) %>%
```

```
  summarise(diff_rate = diff(Rate_Own)) %>%
  pull(diff_rate)
diff
```

```
[1] -0.007829
```

Her er en typisk prosess for å gjennomføre en permutasjonstest med `infer`-pakken:

```
# Steg 1: Spesifiser data og variabler
homeown_perm <- homes %>%
  specify(HomeOwn ~ Gender, success = "Own") %>%

  # Steg 2: Formuler nullhypotesen
  hypothesize(null = "independence") %>%

  # Steg 3: Generer permuterte datasett
  generate(reps = 1000, type = "permute") %>%

  # Steg 4: Beregn teststatistikken (forskjell i proporsjoner)
  calculate(stat = "diff in props", order = c("male", "female"))
```

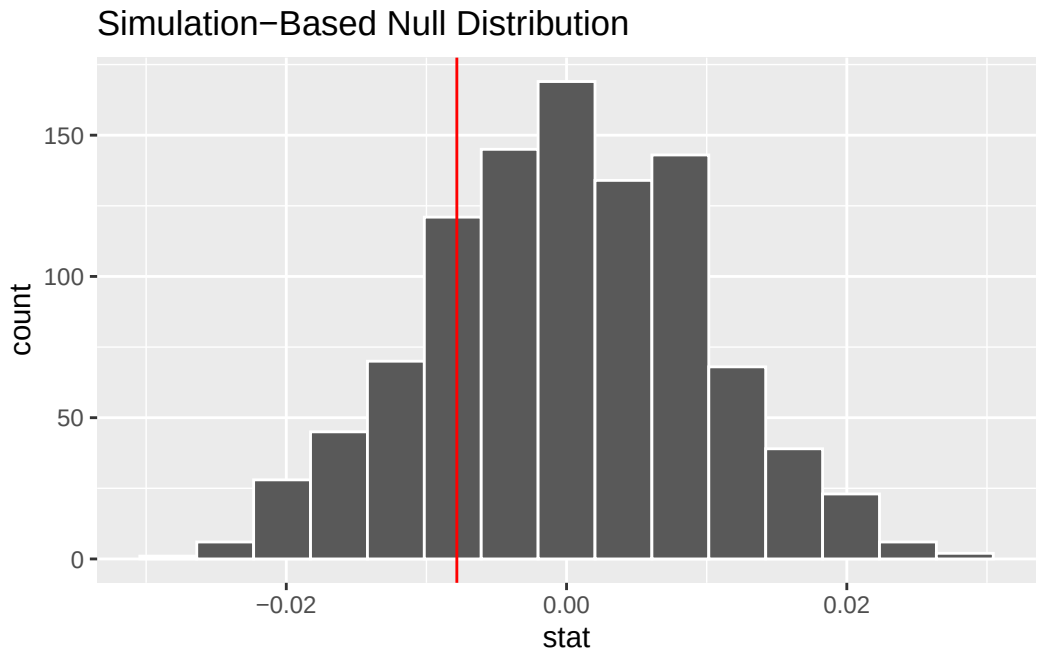
Forklaring av prosessen:

1. **Spesifiser variabler:** Først spesifiserer vi variablene vi ønsker å teste. I dette tilfellet ser vi på andelen boligeiere (`HomeOwn`) delt inn etter kjønn (`Gender`). Vi spesifiserer at "Own" er suksesskategorien, som betyr at vi ønsker å teste for andelen som eier bolig.
2. **Hypotesen:** Nullhypotesen (null = "independence") innebærer at vi antar at andelen boligeiere er uavhengig av kjønn, altså at det ikke er noen forskjell i andelen boligeiere mellom menn og kvinner.
3. **Generer permutasjoner:** Vi bruker `generate()`-funksjonen for å trekke 100 permuterte utvalg fra datasettet, der vi tilfeldig omfordeler kjønnene mens vi beholder boligeier-statusen. Dette skaper en distribusjon av teststatistikker som representerer hva vi ville forventet dersom nullhypotesen var sann.
4. **Beregning av teststatistikken:** Deretter beregner vi teststatistikken, som i dette tilfellet er forskjellen i proporsjoner mellom menn og kvinner. Dette er målet for forskjellen mellom gruppene.

Visualiser premutasjonene

Vi starter med å visualiserer distribusjonen.

```
# Visualize and calculate the p-value for the original dataset
homeown_perm %>%
  visualize() +
  geom_vline(aes(xintercept = diff), color ="red")
```




```
homeown_perm %>%
  get_p_value(obs_stat = diff, direction = "two-sided")
```

```
# A tibble: 1 x 1
  p_value
    <dbl>
1   0.426
```

```
homes <- homes %>%
  mutate(HomeOwn_d = HomeOwn == "Own")

t.test(HomeOwn_d ~ Gender, data=homes)
```

```
    Welch Two Sample t-test

data:  HomeOwn_d by Gender
t = 0.82, df = 9706, p-value = 0.4
alternative hypothesis: true difference in means between group female and group male is not e
95 percent confidence interval:
 -0.01100  0.02665
sample estimates:
mean in group female   mean in group male
              0.6654                0.6576
```

Viktigheten av «There is only one test» i inferens

«There is only one test» gir en kraftig introduksjon til simuleringsbaserte metoder innen statistisk inferens. Ved å bruke permutasjonstesting kan vi introdusere studentene til en mer intuitiv måte å tenke på hypotesetesting, som ikke krever de strenge forutsetningene til mange av de klassiske testene. Permutasjonstesting lar oss utføre realistiske analyser selv med komplekse eller små datasett.

I undervisningen kan denne metoden være svært nyttig for å hjelpe studentene med å forstå kjernen i hypotesetesting, ved å la dem eksperimentere med simuleringsbaserte metoder. Dette gir en praktisk måte å utforske forskjeller mellom grupper på, samtidig som de ser effekten av tilfeldigheter i et utvalg.

Oppsummering

Permutasjonstesting gjennom «There is only one test» i `infer`-pakken representerer en moderne, fleksibel tilnærming til hypotesetesting. Denne tilnærmingen gir studentene en intuitiv forståelse av hvordan vi kan bruke simuleringer og resampling til å evaluere hypoteser. Testen er en god innføring i simuleringsbasert inferens, som også forbereder studentene på mer avanserte metoder som bootstrap og Monte Carlo-simuleringer.

Ved å bruke `infer`-pakken kan vi gjøre inferens og testing mer tilgjengelig og forståelig for studenter, samtidig som vi beholder en høy grad av statistisk rigor.

Gjenta første eksempel

Er personene i datasettet NHANES som er 20 år eller eldre 169 eller lavere?

```r
set.seed(1337)
# Steg 1: Filtrer vekk personer under 20 år
NHANES20 <- NHANES %>%
  filter(Age >= 20) %>%
  filter(!is.na(Height))

# Steg 2: Spesifiser data og variabler
height_test <- NHANES20 %>%
  specify(Height ~ NULL) %>%       # Spesifiserer at vi tester høyde uten noen grupper
  hypothesize(null = "point", mu = 169) %>%  # Nullhypotesen: middelhøyden er 169 cm

  # Steg 3: Generer permuterte datasett
  generate(reps = 1000, type = "bootstrap") %>%  # Bootstrap for å generere nye prøver

  # Steg 4: Beregn teststatistikken (middelverdi)
  calculate(stat = "mean")        # Beregn middelverdien i hver prøve
```

Forklaring:

- **Spesifisering av data**: Vi filtrerer NHANES-datasettet for personer over 20 år og spesifiserer at vi tester høyden.
- **Hypotese**: Nullhypotesen er at den gjennomsnittlige høyden er 169 cm.
- **Generer**: Vi bruker bootstrap for å generere 1000 tilfeldige prøver fra datamaterialet.
- **Beregn teststatistikk**: Beregner middelverdien av høyden i hver bootstrap-prøve.
- **Beregn p-verdi**: Sammenligner den observerte middelhøyden med den bootstrappede fordelingen for å få en p-verdi, som indikerer om den gjennomsnittlige høyden er signifikant større enn 169 cm.

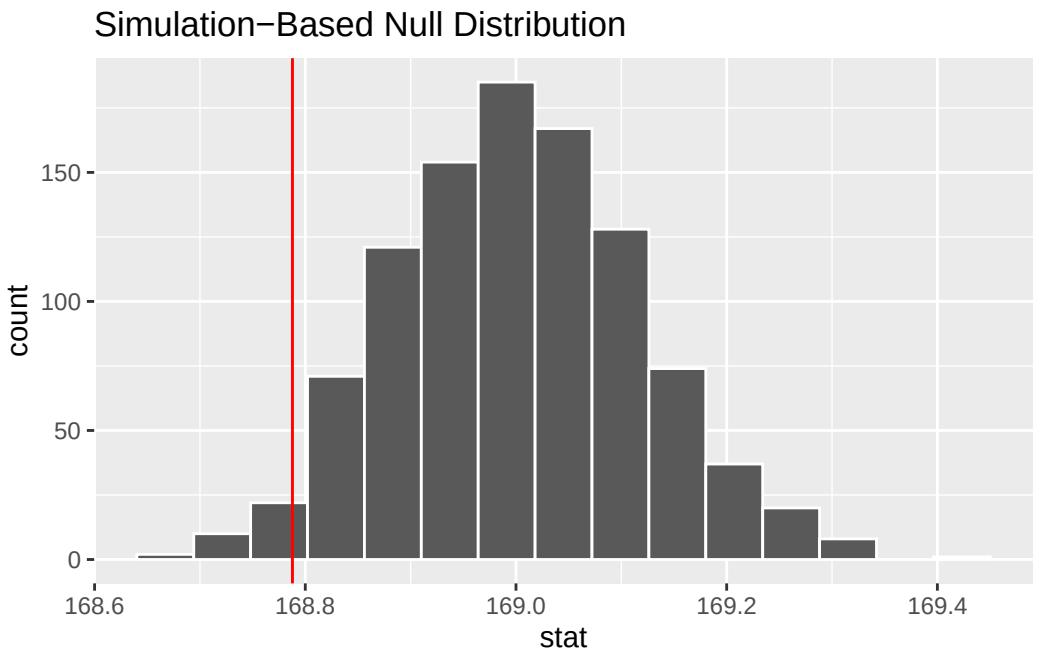
```r
# Steg 5: Sammenlign teststatistikk med den observerte verdien
# Direction her forteller oss rettningen på altelantivhyppotesen.

height_test %>%
  get_p_value(obs_stat = mean(NHANES20$Height), direction = "less")
```

```
# A tibble: 1 x 1
  p_value
    <dbl>
1   0.031
```

P-verdien er 3.1%, noe som betyr at vi kan forkaste nullhypotesen om at den gjennomsnittlige høyden er 169 cm eller høyere. Dette indikerer at vi har tilstrekkelig bevis til å støtte alternativhypotesen om at den gjennomsnittlige høyden i populasjonen er lavere enn 169 cm.

Vi kan visualisere dette resultatet med et histogram:

```
height_test %>%
  visualize() +
  geom_vline(aes(xintercept = mean(NHANES20$Height)), color ="red")
```

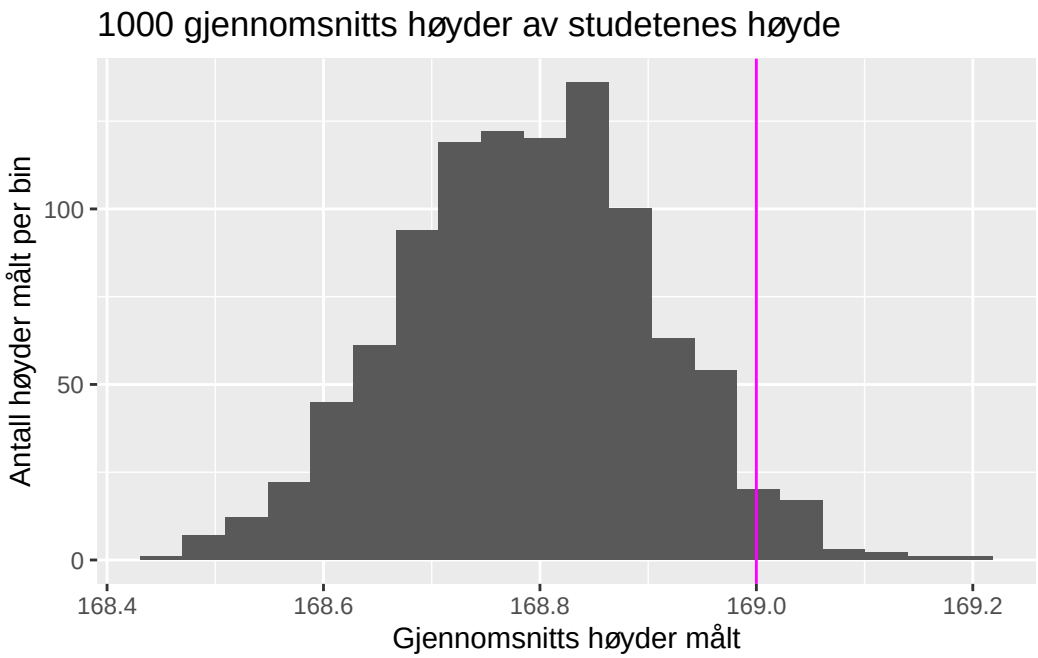


Her ser vi at det observerte gjennomsnittet i utvalget er 168.8 cm, noe som er lavere enn den antatte "sanne" gjennomsnittshøyden på 169 cm under nullhypotesen. Dette gjør det usannsynlig at vårt observerte gjennomsnitt på 168.8 cm ville oppstått hvis det faktiske gjennomsnittet i populasjonen var 169 cm. Vi legger merke til at fordelingen av de bootstrappede gjennomsnittene har en topp nær 169 cm, som er i tråd med nullhypotesen.

La oss visualisere dette nærmere med et histogram:

```
# Plotter alle gjennomsnittene
Heights %>% ggplot(aes(x=Heights)) +
  geom_histogram(bins = 20) +
  geom_vline(xintercept = 169, color = "magenta") +
  xlab("Gjennomsnitts høyder målt") +
```

```
  ylab("Antall høyder målt per bin") +
  ggtitle("1000 gjennomsnitts høyder av studetenes høyde")
```



I grafen ser vi at fordelingen av gjennomsnittene er sentrert rundt 168.8, som er gjennomsnittet vi har fra utvalget. Men når vi utfører en hypotesetest, antar vi at nullhypotesen er korrekt, og at den faktiske fordelingen av gjennomsnittene ville ligge rundt 169 cm.

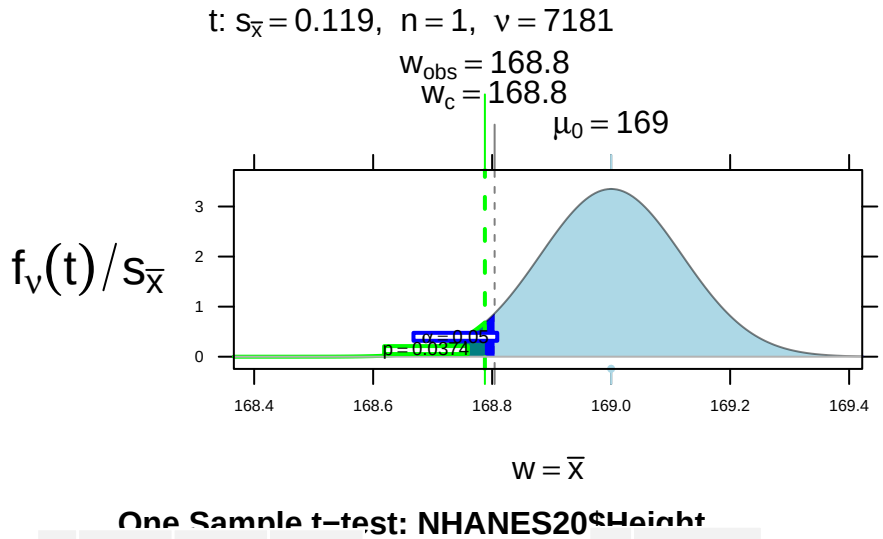
Når vi gjennomfører en t-test, får vi en visualisering som viser den samme logikken som vi ser med `infer`-pakken og "There is only one test" tilnærmingen. La oss utføre en ensidig t-test og bruke NTplot for å visualisere resultatet:

```
# Steg 1: Gjennomfør en ensidig t-test
t_test_result <- t.test(NHANES20$Height,
                        mu = 169,            # Nullhypotesens verdi for gjennomsnitt
                        alternative = "less")  # Ensidig test, høyde lavere enn 169 cm

# Se på resultatet
t_test_result
```

```
    One Sample t-test

data:  NHANES20$Height
```

```
t = -1.8, df = 7181, p-value = 0.04
alternative hypothesis: true mean is less than 169
95 percent confidence interval:
 -Inf  169
sample estimates:
mean of x
    168.8
```

```
NTplot(t_test_result)
```



One Sample t–test: NHANES20$Height

	μ_0	w_{obs}	$w_{crit.L}$
$\bar{x}$	169.00	168.79	168.80
t	0.00	−1.78	−1.65

	Probability
p	0.0374
α	0.0500

Her ser vi at NTplot gir en graf som minner om det vi får fra `infer`-pakken, med visualisering av fordelingen av teststatistikken. Dette illustrerer hvordan begge tilnærmingene demonstrerer konseptet om at under nullhypotesen vil gjennomsnittshøyden være nær 169 cm.

Oppgave

For personer som IKKE er gifg, er det noe forskjell mellom kjønnene om de eier eller leier? Gjennomfør både en "There is only one test" og en t-test.