

02402 Statistik (Polyteknisk grundlag)

Uge 7: Simulering og fejlophobning

DTU Compute
Danmarks Tekniske Universitet
2800 Kgs. Lyngby

Projekt 1

Project 1 afleveres i dag (kl 23.59)

Du afleverer på Learn.

- 1 Opsamling: Python til t-test
- 2 Opsamling: Python til styrkeberegninger
 - Én stikprøve
 - To stikprøver
- 3 Introduktion til simulering
- 4 Fejlophobning

02402 Statistik (Polyteknisk grundlag)

Opsamling: Python til t-test

DTU Compute
Danmarks Tekniske Universitet
2800 Kgs. Lyngby

Python kommandoer for t-test

t-test med én stikprøve:

- `stats.ttest_1samp(data, pop_mean = mu0)`

Welch t-test med to stikprøver:

- `stats.ttest_ind(dataA, dataB, equal_var = False)`

Pooled t-test med to stikprøver:

- `stats.ttest_ind(dataA, dataB, equal_var = True)`

Parret t-test med to stikprøver:

- `stats.ttest_rel(dataA, dataB)`
- `stats.ttest_1samp(diff_data, pop_mean = 0)`

Desuden kan konfidens interval beregnes med (her vises 95% CI):

- `stats.ttest_****(...).confidence_interval(0.95)`

Kahoot!

(x5)

02402 Statistik (Polyteknisk grundlag)

Opsamling: Python til styrkeberegninger

- Én stikprøve
- To stikprøver

DTU Compute
Danmarks Tekniske Universitet
2800 Kgs. Lyngby

02402 Statistik (Polyteknisk grundlag)

- Én stikprøve
- To stikprøver

DTU Compute
Danmarks Tekniske Universitet
2800 Kgs. Lyngby

Forsøgsplanlægning med krav til præcisionen

Fejlmarginen ("margin of error") er defineret som

$$ME = t_{1-\alpha/2} \frac{\sigma}{\sqrt{n}}$$

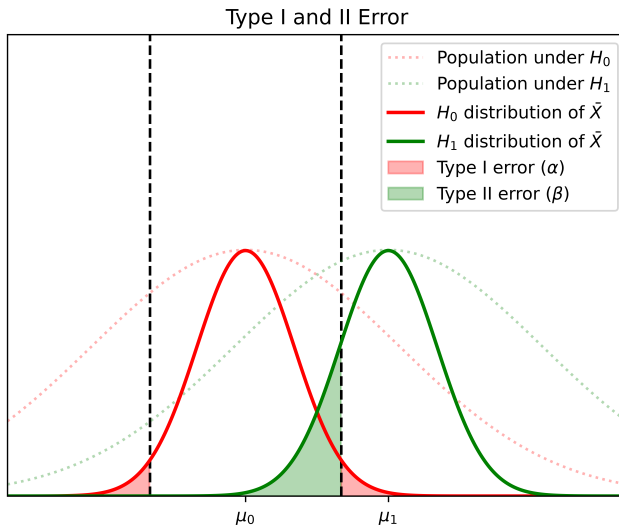
Vi approximerer $t_{1-\alpha/2}$ (fra t -fordeling med $\nu = n - 1$ frihedsgrader) med $z_{1-\alpha/2}$ (fra standard normalfordeling).

|||| Method 3.63 The one-sample CI sample size formula

When σ is known or guessed at some value, we can calculate the sample size n needed to achieve a given margin of error, ME , with probability $1 - \alpha$ as

$$n = \left(\frac{z_{1-\alpha/2} \cdot \sigma}{ME} \right)^2. \quad (3-59)$$

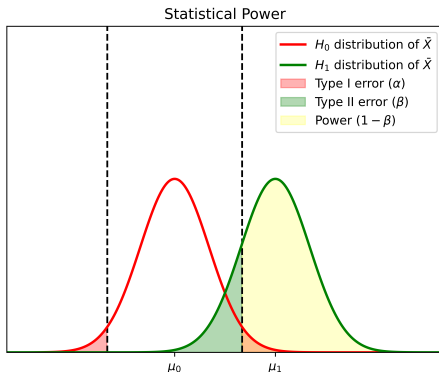
Type I og Type II fejl



α : det røde areal

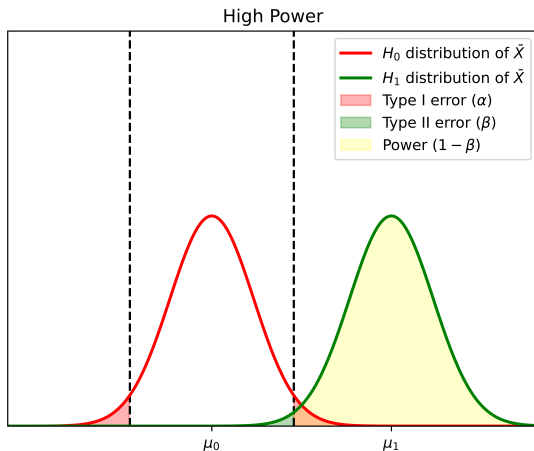
β : det grønne areal

Statistisk Styrke

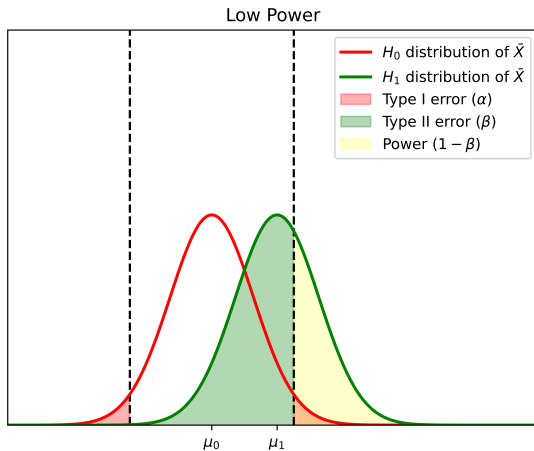


- **Styrke** = $1 - \beta$
- $P(\text{Type II fejl}) = \beta$
- Styrke = "Sandsynligheden for en korrekt afvisning af H_0 ".
- Styrke = "Sandsynligheden for at detektere en (påstået) effekt".
- Typiske værdier for ønsket styrke er 80%, 90%, .. (noget relativt stort)

Eksempel med høj styrke

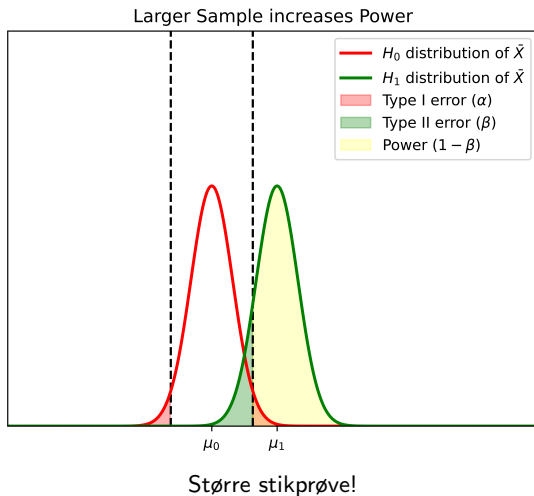


Eksempel med lav styrke



Hvad kan vi gøre?

Eksempel med større stikprøve



Forsøgsplanlægning: Stikprøvestørrelsen n

Det store spørgsmål: Hvor stort skal n være?

Vi skal have nok observationer til at kunne detektere en relevant effekt med høj styrke $1 - \beta$ (typisk mindst 80%):

|||| Method 3.65 The one-sample sample size formula

For the one-sample t -test for given α , β and σ

$$n = \left(\sigma \frac{z_{1-\beta} + z_{1-\alpha/2}}{(\mu_0 - \mu_1)} \right)^2,$$

where $\mu_0 - \mu_1$ is the difference in means that we would want to detect and $z_{1-\beta}$, $z_{1-\alpha/2}$ are quantiles of the standard normal distribution.

(Formlen i Method 3.65 kan udledes ud fra tegningen på forrige slide - du kan prøve selv hvis du har mod på en udfordring)

Forsøgsplanlægning: Styrke

Hvis vi kender (eller antager) tre ud af de fire følgende størrelser, så kan vi finde den manglende:

- Stikprøvestørrelsen (sample size): n
- Signifikansniveauet: α
- Forskellen i middelværdi relativt til populationsstandardafvigelsen ("effekt-størrelsen"): $\frac{\mu_0 - \mu_1}{\sigma}$
- Styrken (power): $1 - \beta$

Python

Vi skal nu bruge nogle funktioner fra pakken `statsmodels.stats.power`

```
(import statsmodels.stats.power as smp)
```

Man kan bruge:

- `smp.TTestPower().solve_power(effect_size=.., alpha=.., power=.., nobs=..)`
- fx:
`smp.TTestPower().solve_power(effect_size=delta/sd, alpha=0.05, power=0.80, nobs=n)`

Python kommandoer til styrke-beregninger

Én stikprøve (udklip fra dokumentation):

```
TTestPower.solve_power(  
    effect_size=None,  
    nobs=None,  
    alpha=None,  
    power=None,  
    alternative='two-sided'  
)
```

[\[source\]](#)

solve for any one parameter of the power of a one sample t-test

for the one sample t-test the keywords are:

effect_size, nobs, alpha, power

Exactly one needs to be `None`, all others need numeric values.

02402 Statistik (Polyteknisk grundlag)

- Én stikprøve
- To stikprøver

DTU Compute
Danmarks Tekniske Universitet
2800 Kgs. Lyngby

Forsøgsplanlægning: To stikprøver

Det store spørgsmål: Hvor store skal n_1 og n_2 være?

Vi kan generalisere forsøgsplanlægningen til den situation hvor vi sammenligner to stikprøver. Stikprøvestørrelserne er nu givet ved n_1 og n_2 . Vi vil antage at variansen (σ^2) er ens i de to populationer.

Formel for stikprøvestørrelse med to stikprøver

For en t-test med to stikprøver, hvor α , β og σ er givet:

$$n_1 = (k + 1) \left(\sigma \frac{z_{1-\beta} + z_{1-\alpha/2}}{\mu_1 - \mu_2} \right)^2$$

Her er $\mu_1 - \mu_2$ den forskel i middelværdier, som vi ønsker at måle, medens $z_{1-\beta}$ og $z_{1-\alpha/2}$ er fraktiler i standardnormalfordelingen. Desuden er $k = n_1/n_2$ forholdet mellem det to stikprøvestørrelser, således at vi har $n_2 = n_1/k$.

(Denne formel er IKKE med i bogen, dvs det er ikke pensum)

Python

Vi skal bruge nogle funktioner fra pakken `statsmodels.stats.power`

```
(import statsmodels.stats.power as smp)
```

Til stikprøvestørrelser for test med to stikprøver kan man bruge:

- `smp.TTestIndPower().solve_power(effect_size=.., alpha=.., power=.., nobs1=.., ratio=..)`
- fx:
`smp.TTestIndPower().solve_power(effect_size=delta/sd, alpha=0.05, power=0.80, nobs1=n1, ratio=..)`

Husk at man regner på størrelsen af den **éne stikprøve** ($nobs1 = n_1$).

Der gælder at $nobs2 = nobs1 \cdot ratio$.

(og hvis de to stikprøver er lige store: $nobs2 = nobs1$, dvs $ratio = 1$).

Python kommandoer til styrke-beregninger

To stikprøver (udklip fra dokumentation):

```
TTestIndPower.solve_power(  
    effect_size=None,  
    nobs1=None,  
    alpha=None,  
    power=None,  
    ratio=1.0,  
    alternative='two-sided'  
)
```

[\[source \]](#)

solve for any one parameter of the power of a two sample t-test

for t-test the keywords are:

effect_size, nobs1, alpha, power, ratio

exactly one needs to be `None`, all others need numeric values

Python kommandoer til styrke-beregninger, opsummering

Én stikprøve:

- `smp.TTestPower().solve_power(...)`

$$n = \left(\sigma \frac{z_{1-\beta} + z_{1-\alpha/2}}{\mu_0 - \mu_1} \right)^2$$

To stikprøver:

- `smp.TTestIndPower().solve_power(...)`
- angiv desuden:
`nobs1 = n1`
 og
`ratio = n2/n1 (= 1/k)`

$$n = (k + 1) \left(\sigma \frac{z_{1-\beta} + z_{1-\alpha/2}}{\mu_1 - \mu_2} \right)^2$$

$$k = n_1/n_2$$

(denne formel er ikke i bogen)

OBS: Python kommandoerne beregner ikke præcis det samme som formlerne!

Kahoot!

(x3)

Introduktion til simulering

DTU Compute
Danmarks Tekniske Universitet
2800 Kgs. Lyngby

Motivation

Vi har allerede brugt simulering:

- Til at simulere terningekast
- Til at motivere CLT (Central Limit Theorem)
- Til at afprøve diverse scenarier relateret til stikprøver

Vi har de sidste par uger beskæftiget os med hypotesetest for **middelværdi(er)**.

Med simulering kan vi undersøge mange andre størrelser - også tilfældige størrelser, som følger mere komplicerede fordelinger.

Hvad er simulering egentlig?

- **Stokastisk simulering** - "sandsynligheds calculus værktøj"
 - (Pseudo)-tilfældige tal genereret af en computer.
 - En *tilfældighedsgenerator* er en algoritme, der kan generere pseudo-tilfældige tal.
 - Algoritmen kræver et såkaldt *seed* (ofte bruges klokkeslet, hvis andet seed ikke er angivet).
-
- Gå til dagens Python notebook i VS Code
 - "Simulating random numbers"



Simulering fra forskellige fordelinger

Hvordan fungerer det?

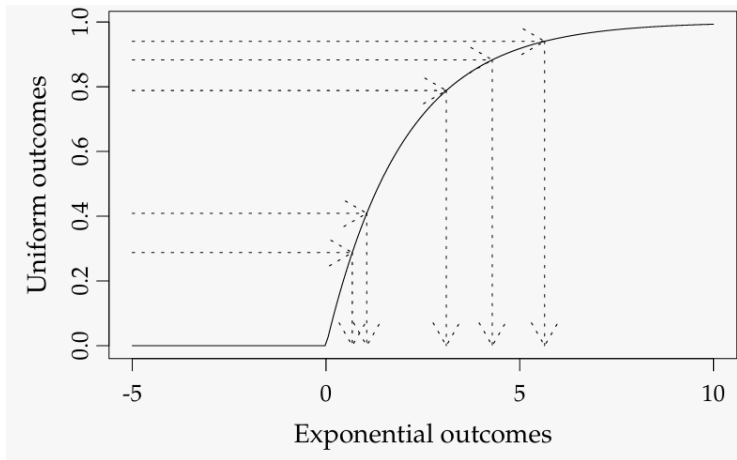
- Man kan simulere fra (næsten) alle fordelinger ved at invertere fordelingsfunktionen F (cdf), og indsætte uniformt fordelt tal:

||| Theorem 2.51

If $U \sim \text{Uniform}(0,1)$ and F is a distribution function for any probability distribution, then $F^{-1}(U)$ follow the distribution given by F

Eksempel: Eksponentialfordelingen med $\lambda = 0.5$

$$F(x) = \int_0^x f(t)dt = 1 - e^{-0.5x}$$



I praksis i Python

Mange fordelinger er tilgængelige via `scipy.stats` pakken og man kan simulere tilfældige tal med `.rvs()`-funktionen (står for *random variates*):

<code>scipy.stats.binom.rvs(...)</code>	Binomialfordelingen
<code>scipy.stats.poisson.rvs(...)</code>	Poissonfordelingen
<code>scipy.stats.hypergeom.rvs(...)</code>	Den hypergeometriske fordeling
<code>scipy.stats.norm.rvs(...)</code>	Normalfordelingen
<code>scipy.stats.lognorm.rvs(...)</code>	Lognormalfordelingen
<code>scipy.stats.expon.rvs(...)</code>	Eksponentialfordelingen
<code>scipy.stats.uniform.rvs(...)</code>	Den uniforme fordeling
<code>scipy.stats.t.rvs(...)</code>	t-fordelingen
<code>scipy.stats.chi2.rvs(...)</code>	χ^2 -fordelingen
<code>scipy.stats.f.rvs(...)</code>	F-fordelingen

Husk også:

<code>np.random.choice(...)</code>	en diskret fordeling (fx terning)
------------------------------------	-----------------------------------

Eksempel: Areal af plader

En virksomhed producerer rektangulære plader.

Længden af pladerne (i meter), X , antages at kunne beskrives ved normalfordelingen $N(2, 0.01^2)$, medens bredden af pladerne (i meter), Y , antages at kunne beskrives ved normalfordelingen $N(3, 0.02^2)$. Man kan antage, at pladernes længder og bredder er uafhængige.

Man er interesseret i arealet, A , som jo så givet ved $A = XY$.

- Hvad er middelarealet?
- Hvad er spredningen i arealet fra plade til plade?
- Hvor ofte har sådanne plader et areal, der afviger mere end 0.1 m^2 fra de angivne 6 m^2 ? (en sandsynlighed)
- Hvad er fordelingen af arealer?

Eksempel: Areal af plader (simulation)

Hvordan simulerer vi længden af pladerne?

$$X \sim N(2, 0.01^2)$$

Hvordan simulerer vi bredden af pladerne?

$$Y \sim N(3, 0.02^2)$$

Hvordan simulerer vi arealet af pladerne?

$$A = XY$$

Kahoot!

(x1)

- Gå til dagens Python notebook i VS Code
 - "Area of plates Example"



Visual Studio Code

Eksempel: Areal af plader (simulation)

Resultat fra simulering:

Stikprøve-standardafvigelsen af arealerne var: $s_A = 0.05$

Risikoen (sandsynligheden) for at pladerne afviger mere end 0.1 m^2 fra de angivne 6 m^2 var ca: 0.0476, dvs 4-5 %.

Eksamensopgave III fra Juni 2021

Exercise III

The card game Casino is played with an ordinary deck of cards, i.e. 52 cards, 13 of each colour (clubs, hearts, diamonds and spade), and 4 of each value (e.g. 4 aces). The game starts by placing 4 cards face up on the table.

Question III.1 (7)

What is the probability that at least 3 of the 4 opening cards are aces?

- 1 ☐ 0.00057
- 2 ☐ $2.7 \cdot 10^{-6}$
- 3 ☐ 0.025
- 4 ☐ 0.00071
- 5 ☐ $3.7 \cdot 10^{-6}$

Eksamensopgave III fra Juni 2021 - alternative svarmuligheder til papir-og-blyant eksamen

```
print(1 - stats.hypergeom.cdf(2, M=56, n=4, N=4))  
0.00057  
  
print(1 - stats.hypergeom.cdf(3, M=56, n=4, N=4))  
2.7e-06  
  
print(stats.hypergeom.pmf(2, M=52, n=4, N=4))  
0.025  
  
print(1 - stats.hypergeom.cdf(2, M=52, n=4, N=4))  
0.0007  
  
print(1 - stats.hypergeom.cdf(3, M=52, n=4, N=4))  
3.7e-06
```

Eksamensopgave III.1 fra Juni 2021 - Simulering

Exercise III

The card game Casino is played with an ordinary deck of cards, i.e. 52 cards, 13 of each colour (clubs, hearts, diamonds and spade), and 4 of each value (e.g. 4 aces). The game starts by placing 4 cards face up on the table.

Question III.1 (7)

What is the probability that at least 3 of the 4 opening cards are aces?

- 1 ☐ 0.00057
- 2 ☐ $2.7 \cdot 10^{-6}$
- 3 ☐ 0.025
- 4 ☐ 0.00071
- 5 ☐ $3.7 \cdot 10^{-6}$



Visual Studio Code

Python: "Exam question III June 2021 (1)"

Eksamensopgave III fra Juni 2021

Question III.2 (8)

What is the probability that all the 4 opening cards are clubs in exactly 1 out of 4 games?

1 ☐ 0.010

2 ☐ 0.39

3 ☐ 0.42

4 ☐ 0.44

5 ☐ 0.015

Eksamensopgave III fra Juni 2021 - simulering

Question III.2 (8)

What is the probability that all the 4 opening cards are clubs in exactly 1 out of 4 games?

1 ☐ 0.010

2 ☐ 0.39

3 ☐ 0.42

4 ☐ 0.44

5 ☐ 0.015



Visual Studio Code

Python: "Exam question III June 2021 (2)"

Fejlophobning

DTU Compute
Danmarks Tekniske Universitet
2800 Kgs. Lyngby

Fejlophobning (eng: "Error propagation")

Man ønsker at finde:

$$\sigma_{f(X_1, \dots, X_n)}^2 = \text{Var}(f(X_1, \dots, X_n))$$

Eksempel: $U = f(X, Y) = 2X + Y$

Lineærkombination af uafhængige variable:

$$\sigma_{f(X_1, \dots, X_n)}^2 = \sum_{i=1}^n a_i^2 \sigma_i^2 \text{ når } f(X_1, \dots, X_n) = \sum_{i=1}^n a_i X_i \text{ (med uafhængighed)}$$

Eksempel: $\sigma_U^2 = 2^2 \sigma_X^2 + \sigma_Y^2$

Fejlophobning (eng: "Error propagation")

Man ønsker at finde:

$$\sigma_{f(X_1, \dots, X_n)}^2 = \text{Var}(f(X_1, \dots, X_n))$$

Eksempel: $W = f(X, Y) = 3X \cdot Y$

IKKE en lineær kombination!

Så har vi ikke nogen formel til at bestemme σ_W^2

Vi må finde σ_W^2 på en anden måde!

Vi har allerede set at man kan simulere - men der findes også en anden metode:
"Error propagation"

Fejlophobning (eng: "Error propagation")

|||| Method 4.3 The non-linear approximative error propagation rule

If $X_1, \dots, X_n$ are independent random variables with variances $\sigma_1^2, \dots, \sigma_n^2$ and f is a (potentially non-linear) function of n variables, then the variance of the f -transformed variables can be approximated linearly by

$$\sigma_{f(X_1, \dots, X_n)}^2 = \sum_{i=1}^n \left(\frac{\partial f}{\partial x_i} \right)^2 \sigma_i^2, \quad (4-2)$$

where $\frac{\partial f}{\partial x_i}$ is the partial derivative of f with respect to the i 'th variable

Eksempel: Areal af plader (fejlophobning)

I første del af forelæsningen brugte vi simulation til at undersøge variansen af arealet af plader.

Vi kan også beregne variansen af $A = f(X, Y) = X \cdot Y$ med fejlophobningsloven:

$$\sigma_A^2 \approx \left(\frac{\partial f}{\partial x} \right)^2 \sigma_X^2 + \left(\frac{\partial f}{\partial y} \right)^2 \sigma_Y^2$$

Eksempel: Areal af plader (fejlophobning)

$$\sigma_A^2 \approx \left(\frac{\partial f}{\partial x} \right)^2 \sigma_X^2 + \left(\frac{\partial f}{\partial y} \right)^2 \sigma_Y^2$$

Varianserne er:

$$\sigma_X^2 = 0.01^2 \text{ og } \sigma_Y^2 = 0.02^2$$

Eksempel: Areal af plader (fejlophobning)

$$\sigma_A^2 \approx \left(\frac{\partial f}{\partial x} \right)^2 \sigma_X^2 + \left(\frac{\partial f}{\partial y} \right)^2 \sigma_Y^2$$

Varianserne er:

$$\sigma_X^2 = 0.01^2 \text{ og } \sigma_Y^2 = 0.02^2$$

Funktionen og dens partielt afledte er:

$$f(x, y) = xy, \quad \frac{\partial f}{\partial x} = y, \quad \frac{\partial f}{\partial y} = x$$

Eksempel: Areal af plader (fejlophobning)

$$\sigma_A^2 \approx \left(\frac{\partial f}{\partial x}\right)^2 \sigma_X^2 + \left(\frac{\partial f}{\partial y}\right)^2 \sigma_Y^2$$

Varianserne er:

$$\sigma_X^2 = 0.01^2 \text{ og } \sigma_Y^2 = 0.02^2$$

Funktionen og dens partielt afledte er:

$$f(x,y) = xy, \quad \frac{\partial f}{\partial x} = y, \quad \frac{\partial f}{\partial y} = x$$

Vi skal bruge en værdi for $\frac{\partial f}{\partial x} = y$ og $\frac{\partial f}{\partial y} = x$:

Det er normalt at man bruger middelværdierne som estimat for værdierne y og x :

$$y \approx \mu_Y = 3 \text{ og } x \approx \mu_X = 2$$

Eksempel: Areal af plader (fejlophobning)

$$\sigma_A^2 \approx \left(\frac{\partial f}{\partial x}\right)^2 \sigma_X^2 + \left(\frac{\partial f}{\partial y}\right)^2 \sigma_Y^2$$

Så resultatet bliver:

$$\begin{aligned}\sigma_A^2 &\approx y^2 \sigma_X^2 + x^2 \sigma_Y^2 \\ &\approx 3^2 \cdot 0.01^2 + 2^2 \cdot 0.02^2 = 0.0025\end{aligned}$$

og derfor:

$$\sigma_A \approx 0.05 \quad (\text{passer med } s_A \text{ fra simuleringen!})$$

Fejlophobning – ved simulation

|||| Method 4.4 Non-linear error propagation by simulation

Assume we have actual measurements $x_1, \dots, x_n$ with known/assumed error variances $\sigma_1^2, \dots, \sigma_n^2$:

1. Simulate k outcomes of all n measurements from assumed error distributions, e.g. $N(x_i, \sigma_i^2)$: $X_i^{(j)}, j = 1 \dots, k$.
2. Calculate the standard deviation directly as the observed standard deviation of the k simulated values of f :

$$s_{f(X_1, \dots, X_n)}^{\text{sim}} = \sqrt{\frac{1}{k-1} \sum_{j=1}^k (f_j - \bar{f})^2}, \quad (4-3)$$

where

$$f_j = f(X_1^{(j)}, \dots, X_n^{(j)}). \quad (4-4)$$

Eksempel: Areal af plader (teoretisk udledning)

Faktisk kan vi i dette eksempel udlede variansen for $A = XY$ teoretisk (fra ex. 4.5):

following fundamental relationship

$$V(X) = E(X - E(X))^2 = E(X^2) - E(X)^2. \quad (4-5)$$

So, one can actually deduce the variance of A theoretically, it is only necessary to know in addition that for independent random variables: $E(XY) = E(X)E(Y)$ (which by the way then also tells us that $E(A) = E(X)E(Y) = 6$)

$$\begin{aligned} V(XY) &= E[(XY)^2] - E(XY)^2 \\ &= E(X^2)E(Y^2) - E(X)^2E(Y)^2 \\ &= [V(X) + E(X)^2][V(Y) + E(Y)^2] - E(X)^2E(Y)^2 \\ &= V(X)V(Y) + V(X)E(Y)^2 + V(Y)E(X)^2 \\ &= 0.01^2 \cdot 0.02^2 + 0.01^2 \cdot 3^2 + 0.02^2 \cdot 2^2 \\ &= 0.00000004 + 0.0009 + 0.0016 \\ &= 0.00250004. \end{aligned}$$

Note, how the approximate error propagation rule actually corresponds to the two latter terms in the correct variance, while the first term – the product of the two variances is ignored. Fortunately, this term is the smallest of the three in this case. It does not always have to be like that. If you want to learn how to make a theoretical derivation of the density function for $A = XY$ then take a course in probability calculation.

Forskellige tilgange

Tre forskellige tilgange:

- 1 Simulation
- 2 Den approksimative *fejlophobningslov*
- 3 Teoretisk udledning

Forskellige tilgange

Tre forskellige tilgange:

- 1 Simulation
- 2 Den approksimative *fejlophobningslov*
- 3 Teoretisk udledning

Simulation har nogle vigtige fordele:

- 1 Nem måde at beregne andre størrelser end σ .
- 2 Nem måde at bruge andre fordelinger end normalfordelingen.
- 3 Afhænger ikke af en lineær tilnærmelse af den underliggende ikke-lineære funktion (i modsætning til fejlophobningsloven).

Eksamensopgave XI fra Maj 2024

Exercise XI

In a laboratory, one wants to dilute a stock solution A of concentration C_A to obtain a solution B of concentration C_B . The following rule is applied:

$$C_B = \frac{C_A V_A}{V_B}$$

Here, V_A and V_B represent the volumes of the initial stock solution A and the resulting solution B, respectively. C_A , C_B , V_A and V_B are all random variables.

Question XI.1 (20)

Which of the following formulas can be utilized to approximate the standard deviation of solution B's concentration (σ_{C_B}) via the non-linear error propagation rule?

Eksamensopgave XI fra Maj 2024

$$C_B = \frac{C_A V_A}{V_B}$$

$$1 \quad \square \quad \sigma_{C_B} = \sqrt{\frac{\left(\frac{\partial C_B}{\partial C_A} \sigma_{C_A}\right)^2 + \left(\frac{\partial C_B}{\partial V_A} \sigma_{V_A}\right)^2}{\left(\frac{\partial C_B}{\partial V_B} \sigma_{V_B}\right)^2}}$$

$$2 \quad \square \quad \sigma_{C_B} = \left(\frac{\partial C_B}{\partial C_A} \sigma_{C_A}\right)^2 + \left(\frac{\partial C_B}{\partial V_A} \sigma_{V_A}\right)^2 + \left(\frac{\partial C_B}{\partial V_B} \sigma_{V_B}\right)^2$$

$$3 \quad \square \quad \sigma_{C_B} = \sqrt{\left(\frac{\partial C_B}{\partial C_A} \sigma_{C_A}\right)^2 + \left(\frac{\partial C_B}{\partial V_A} \sigma_{V_A}\right)^2 + \left(\frac{\partial C_B}{\partial V_B} \sigma_{V_B}\right)^2}$$

$$4 \quad \square \quad \sigma_{C_B} = \frac{\left(\frac{\partial C_B}{\partial C_A} \sigma_{C_A}\right)^2 + \left(\frac{\partial C_B}{\partial V_A} \sigma_{V_A}\right)^2}{\left(\frac{\partial C_B}{\partial V_B} \sigma_{V_B}\right)^2}$$

$$5 \quad \square \quad \sigma_{C_B} = \left(\frac{\partial C_B}{\partial C_A} \sigma_{C_A}\right) + \left(\frac{\partial C_B}{\partial V_A} \sigma_{V_A}\right) + \left(\frac{\partial C_B}{\partial V_B} \sigma_{V_B}\right)$$

Kahoot!
(x1)

Eksamensopgave XI fra Maj 2024

$$C_B = \frac{C_A V_A}{V_B}$$

Hvad ville man faktisk sætte ind i formelen?

$$3 \quad \sigma_{C_B} = \sqrt{\left(\frac{\partial C_B}{\partial C_A} \sigma_{C_A}\right)^2 + \left(\frac{\partial C_B}{\partial V_A} \sigma_{V_A}\right)^2 + \left(\frac{\partial C_B}{\partial V_B} \sigma_{V_B}\right)^2}$$

Kræver konkrete værdier:

$$C_A = 1M$$

$$(\sigma_{C_A} = 0.01M)$$

$$V_A = 5mL$$

$$(\sigma_{V_A} = 0.03mL)$$

$$V_B = 1000mL$$

$$(\sigma_{V_B} = 1mL)$$

Eksamensopgave XI fra Maj 2024

Hvordan kunne man løse samme opgave med simulering?



Visual Studio Code

Python: "Exam question XI May 2024"

Tjekliste

Efter i dag skal du kunne:

- Simulere tilfældige tal fra diverse sandsynlighedfordelinger
- Opsætte en simulering der kan bruges til at undersøge en kontekst med tilfældige udfald
- Simulere afledte størrelser fra stokastiske variable og udføre diverse beregninger på disse afledte størrelser
- Benytte fejlophobningsloven til at estimere usikkerheder på afledte størrelser af mere simple stokastiske variable