Dokumentasjon – KI-støttet arbeid med figurgrunnlag og figurer til kapittel 1.2: kontroller og analysearbeid

I arbeidet med kapittel 1.2 er det brukt et KI-verktøy (Claude pro, med KI-modellen Opus 4.8 innstilt på et relativt høyt innsatsnivå). KI-verktøyet er brukt til å etablere figurgrunnlag og som støtte for å analysere tallene. Dette verktøyet arbeider dels generativt, dvs probabilistisk, dels deterministisk. Alle figurgrunnlag er laget med pythonkode og API-kall, som er rent deterministiske prosesser fra kode til resultat og der det ikke kan skje «hallusinasjon» ved at tall framkommer som resultat av KI-baserte sannsynligheter. Om KI-modellen er «trent på FoU-tall» eller på tekstmateriale om FoU etc eller ikke er trent på FoU, er med andre ord helt irrelevant. Derimot er modellen trent på å utføre oppgaver som å hente data med API, ekstrahere tall fra for eksempel regneark, hente tekst og diagrammer som er lagret i egen, avsondret prosjektmappe eller hente data fra web. Modellen er også trent på å skrive kode. Koding og databehandling skjer i egen «sandkasse», isolert fra alt annet enn de dedikerte dataene, og med en linux-container der koden eksekveres.

Kodebiblioteket og eksekvering av kode er isolert og upåvirkelig av støy. Selve kodingen skjer derimot generativt, og kan inneholde feil. At modellen er trent på denne typen oppgaver (koding og databehandling, uansett tema) betyr også at den på egen hånd vil kunne beslutte å gjøre kontroller av sitt eget arbeid; nettopp fordi den erfaringsmessig noen ganger gjør feil. Slike feil kan for eksempel være kolonneforskyvninger eller å ta med delsummer i totaler (dobbeltelling). Slik egenkontroll er ikke en del av systemarkitekturen til Claude, men er i seg selv generativt og dermed ikke noe som skjer automatisk og ikke nødvendigvis hver gang.

I det følgende gjør vi rede for kontrollene som er gjennomført i etablering av figurgrunnlagene. Grovt sett består det av kontroll før prosessering, kontroll under prosessering og kontroll etter prosessering. Deretter gis det et eksempel på hvordan KI-verktøyet er brukt analytisk (på kontrollerte data). Beskrivelsen av kontroller utført under prosessering er basert på loggfiler fra oppdragene utført med KI.

Før prosessering: Kildevalg. Forfatter valgte kildefiler og la inn i en dedikert prosjektmappe (regnearket med tabellvedlegget A6, Indikatorrapporten for 2025, kapittel 1.2 Næringsliv på pdf-format), og bestemte slik hvilke kilder som skulle være autoritative. Dette for å unngå at modellen henter tall fra hukommelsen eller fra et tilfeldig nettsøk. Modellen ble instruert om å bruke API fra SSB der hvor tabellnumrene var oppgitt i pdf-dokumentet, og lete i regnearket A6 der hvor tabellnumre ikke var oppgitt (tabellnumrene står i kapittelets kildelinjer, kun figur 1.2a manglet presis henvisning («FoU-statistikk, SSB») og der ble data dermed hentet fra tabellvedlegget A6. Ikke alle tabeller fra 2025 er med i årets utgave, dels fordi siste FoU-undersøkelse ikke dekker de samme temaene, dels utfra forfatters seleksjon.

Under prosessering: Kontroller. Det ble til sammen gjort 114 kontroller av type «K» (se nedenfor), de fleste KI-genererte (altså Claudes selvinitierte egenkontroller) og noen på instruks. 71 kontroller var på dataene og 43 på figurene.

I tillegg ble det gjort kryssjekker mot uavhengige kilder: Tall fra tabellvedlegget A6 (fig.1.2a) ble (på forfatters instruks) sjekket mot statistikkbankens tabell 11010, deflatoren ble kontrollert uavhengig mot tre kilder (dette var en kontroll initiert av KI i løpet av dialogen), og samme totaltall ble verifisert i fem tabeller (også initiert av KI). Noen «negative tester» ble også gjort ved at KI la inn kunstige feil for å bekrefte at kontrollene faktisk slår ut[1]. I utgangspunktet var også disse kontrollene initiert av KI, men antakelig utløst eller framprovosert av kontrollspørsmål i dialogen.

Alle avledede tall (faste priser) ble lagt inn som formler i levende regneark, med peker til (de trippelkontrollerte) deflatorene og til kolonnene som inneholdt løpende priser. Foruten å være mer robust enn rene tallkolonner med lite transparent opphav, var de dermed enkle å kontrollere (er deflator riktig? Er

[1] Konkret besto negative kontroller av å fjerne en kategori og sjekke om kontroll av summer mot publiserte totalsummer slo ut.

formelen riktig?).

Etter prosessering: Manuell kontroll. Tall i teksten og figurgrunnlagene ble kontrollert av to medarbeidere og i to omganger.

Kildevalg

Figur 1.2a er hentet fra regnearket i tabellvedlegg A6, siden det her manglet presis kildeangivelse i 2025-årgangen. Den samme serien er som nevnt også hentet fra statistikkbanktabell 11010 via API og bygd som en egen kontrollversjon, slik at versjonen basert på A6-arket kunne etterprøves mot statistikkbanken år for år (kontrollversjonen inngår ikke i kapittelet). Dette tjener også som test både på reliabiliteten til datainnhenting med API og fra regneark

Figurene 1.2b–1.2k er hentet direkte fra SSBs statistikkbank-API, fra de tabellene som er oppgitt i kildehenvisningen under hver figur i næringslivskapittelet i 2025.

Deflator er konstruert med tre uavhengige metoder og denne trianguleringen inngår blant de KI-genererte egenkontrollene. Tabell 1 angir oversikt over brukte datakilder, slik det framgår av loggfiler. Se også boks 1.

Tabell 1. Datakilder brukt av KI (fra loggfil).

Figur	Kilde	Innhold
1.2a	A6-arket	FoU-utgifter i næringslivet, løpende og faste 2015-priser (default)
1.2a-K	11010	Samme serie fra API – kontrollversjon på bestilling, inngår ikke i kapittelet
1.2b	07963	FoU-kostnader, tjenesteytende enkeltnæringer
1.2c	07963	FoU-kostnader, industrinæringer
1.2d	07963	FoU-kostnader, andre næringer
1.2e	07967	FoU-kostnader etter hovednæring og sysselsettingsgruppe
1.2f	07967	FoU-kostnader etter utgiftsart og hovednæring
1.2g	07969	Finansiering etter kilde
1.2h	07969	Finansiering etter kilde og sysselsettingsgruppe
1.2i	07965	Finansiering etter kilde og næring
1.2j	07970	Innkjøpte FoU-tjenester etter aktør
1.2k	07970	Innkjøpte FoU-tjenester etter hovednæring og sysselsettingsgruppe
(deflator)	09170	Prisindeks for næring 72, kjedet til 2015 = 1

Boks 1. Presis kilde for figur 1.2.1 brukt av KI.

Presis kilde i A6-arket for figur 1.2a (fra loggfil og verifisert av forfatter)

- **Fil:** A6tabeller_FASTE_PRISER_20142024.xlsx
- **Fane:** «A.6.3a,kostn»
- **Årshoder:** rad 9, kolonne 3–13 (2014–2024)
- **Løpende priser:** blokk «LØPENDE PRISER» → rad 10, celler **C10:M10**
- **Faste 2015-priser:** blokk «FASTE PRISER / Absolutte tall (mill.kr)» → rad 159, celler **C159:M159**
- **Kontrollserie:** blokk «FASTE PRISER / Prosent endring» → rad 60 (A6s egen publiserte realendring, brukt som uavhengig radkontroll)
- **Identifisert ved:** blokkoverskrift (eksakt treff, kolonne A) + næringskode «A-N» i kolonne A; årskolonner fra årshoderaden

Kontroller under prosessering

Kontrollene utført under prosessering, de fleste KI-genererte, kan grupperes i 5 typer:

- K1 API-respons: KI sjekket statuskode, innholdstype, at kategoriene er de som ble bedt om og at verdiene er numeriske. «Statuskode» forteller om kilden er oppe og går (for å stoppe ev forsøk på å gjette tall dersom API'et er nede eller ikke gyldig). «Innholdstype» verifiserer tabellnummer. Verifiseringen skjer med Python-kode og gir feilmelding og terminerer med feilmelding dersom kontrollen feiler.
- K2 kolonnevis kontroll: at kolonneetiketten stemmer overens med tallet som leses, at radetiketten

er den korrekte.

- K3 radvis kontroll: komponentsummer ble sjekket mot publiserte totaler, kontroll av aggregat vs komponent-klassifisering, beregning av restposter og kontroll mot forventede verdier
- K4 figurvalidering: kategoriantall, verditreff mot arbeidsboken, prosentsummer
- K5 løpende dialog mellom forfatter og KI-modell (prompt og oppfølgingsprompt)

Hvordan jobber språkmodellen sammen med andre funksjonaliteter i Claude Pro?

Det er viktig å ha klart for seg at KI-modellen Claude ikke utfører en visuell "titt" på data i regneark slik vi ser det i et regneark åpnet i Excel, men søker seg fram til strukturen med kode (her: Python) i et «sandkasse-miljø» (dvs. en isolert kode-plattform, typisk med openpyxl for å lese celler direkte eller pandas/openpyxl kombinert for å navigere strukturen). Tilsvarende er innlesing av data med API heller ikke «KI» men deterministisk. Nedenfor beskriver vi prosedyrene steg for steg:

Regneark:

1. Åpne arbeidsboken programmatisk, ikke visuelt
KI bruker openpyxl (evt. pandas for enklere tabeller) inne i sandkasse-miljøet. KI «ser» aldri et regneark men får celleverdier tilbake som Python-objekter, radvis og kolonnevis.

2. Kartlegg strukturen før noe innhold leses inn
Første steg er alltid *wb.sheetnames* for å finne riktig fane, deretter et generelt søk i kolonner og rader for å finne blokkoverskrifter. Her kreves *eksakt* match mot malen. I tabellvedlegget A6, arkfane A.6.3a,kostn inneholder for eksempel rad 6 «% endring fra forrige år (løpende priser)» mens rad 8 inneholder «LØPENDE PRISER». Et delstrengsøk etter «LØPENDE PRISER» ville truffet rad 6 hvis KI ikke hadde krevd eksakt celletekst-match i Python-koden.

3. Søk opp posisjoner, ikke hardkod dem
KI finner blokkens startkolonne fra blokkhodet, verifiserer årstall i hver kolonne det leses fra og verifiserer radetiketten («A-N i kolonne A, NÆRINGSLIVET TOTALT i kolonne B») før verdien godtas. (Dette er K2- og K3-kontrollene).

4. Eksekvering av formler
Når beregnede verdier behøves (altså ikke formelen), åpnes filen og tallene beregnes av Excel/LibreOffice. KI skriver skriver formler, men må kjøre LibreOffice for å få tallverdier tilbake. Det framgår av loggfilen at det faktisk er dette som skjer, altså at KI faktisk kjører den genererte koden og bruker denne.

5. Selve figurgrunnlagsarbeidet
Når KI bygger et regneark *.xlsx med avledede tall, brukes openpyxl til å skrive levende formler, *ikke* innskrevne tall. Deretter kjøres LibreOffice å kjøre rekalkulering og lese tilbake verdiene som kontroll. Selve regnearket inneholder levende formler med referanse til input-celler (for eksempel til kolonner med løpende priser og celler med deflator), som både gir transparens og gjør manuell kontroll enkelt.

API:

KI får ikke et grensesnitt å bla i, men henter strukturen gjennom programmering og bygger opp en forståelse av dataene i svaret fra kilden. Slik foregår prosessen etter at kilden er verifisert og bekreftet aktiv:

1. Hent metadata først, alltid. GET mot samme URL gir dimensjonskoder og kategorikoder. KI gjetter *aldri* kildens kodenavn; feil dimensjonsnavn (f.eks. SyssGrp vs. SysselsettGr) gir feilmelding (HTTP 400), mens feil kategorikode vil gi feil tall uten varsel. Derfor er denne innledende kontrollen viktig.

2. Bygg API-spørringen med de bekreftede kodene. JSON-body med query-liste per dimensjon og response.format "json-stat2".

3. Pakk ut svaret etter strukturen i kilden (SSB-tabellen), ikke spørringens struktur. KI bygger et rekursivt oppslag over indeksene uavhengig av rekkefølgen i API-kallet. Et eksempel: KI spør om [2024, 2023], men svaret i API'et kommer tilbake ordnet 2023, 2024 fordi SSB ignorerer spørringens rekkefølge og normaliserer til tabellens egen.

4. Kontroller gjennomføres som mengder (sett), ikke sekvenser. KI sjekker set(fikk) == set(bedt) for kategoriene (for eksempel år). En sekvenssjekk (bedt == fikk) i eksempelet foran ville slått ut hver eneste gang på grunn av avvikende rekkefølge og blitt en falsk alarm i stedet for en reell kontroll.

5. Håndter null. Prikkede celler kommer som null i value-listen. De registreres eksplisitt (hvilken kategori, hvilket år).

6. Logg alt. Statuskode, antall celler, antall null og tabellens tidsstempel («sist oppdatert») fra metadata inngår i API-loggen.

Kontroll K1 – API-responssjekk

Uttrekket avbryter hvis statuskoden ikke er «200,» hvis Content-Type ikke er JSON, hvis kategoriene i svaret ikke er nøyaktig de det ble spurt om eller hvis noen verdier ikke er numeriske. Tabell 2 oppsummerer resultater fra API-loggen.

Tabell 2. API-responser.

Tabell	HTTP	Celler	Uten publisert verdi	Sist oppdatert	Gjelder
09170	200	10	0	2026-05-28T06:00:00Z	deflator: prisindeks FoU (næring 72), årlig prisendring
11010	200	22	0	2026-04-16T06:00:00Z	figur 1.2a-K: kontrollversjon fra statistikkbanken
07963	200	94	6	2026-04-16T06:00:00Z	figur 1.2b/c/d: FoU-kostnader etter næring
07967	200	28	0	2026-04-16T06:00:00Z	figur 1.2e: FoU-kostnader etter hovednæring og sysselsetting
07967	200	24	0	2026-04-16T06:00:00Z	figur 1.2f: FoU-kostnader etter utgiftsart og hovednæring
07969	200	22	0	2026-04-16T06:00:00Z	figur 1.2g: finansiering etter kilde
07969	200	42	0	2026-04-16T06:00:00Z	figur 1.2h: finansiering etter sysselsettingsgruppe
07965	200	282	19	2026-04-16T06:00:00Z	figur 1.2i: finansiering etter næring
07970	200	16	0	2026-04-16T06:00:00Z	figur 1.2j: innkjøpt FoU etter aktør
07970	200	28	0	2026-04-16T06:00:00Z	figur 1.2k: innkjøpt FoU etter hovednæring og sysselsetting

Kontroll K2 – kolonnevise kontroller (samsvar mellom etikett og tall)

A6 (fig 1.2a): Kolonneplasseringen søkes opp, den antas ikke. Blokkoverskriften finnes med eksakt tekstsammenlikning i kolonne A (et delstrengsøk kunne gitt falskt treff). Årskolonnene leses fra årshoderaden innenfor blokken, og det verifiseres at årene er 2014–2024 og ligger sammenhengende. Ved hver enkelt avlesning kontrolleres dessuten at årshodet i kolonnen faktisk er det året verdien tilskrives, og at raden har næringskoden «A-N» og navnet «NÆRINGSLIVET TOTALT».

Dette er nødvendig fordi årshodene gjentas i hver kostnadsart-blokk: kolonnen rett etter «Totalt»-blokken er 2014 igjen, starten på «Driftskostnader». Uten blokkforankring kunne KI lese riktig år fra feil blokk.

- K2 A6: årskolonner funnet ved oppslag i årshoderad (ikke hardkodet): 2014 = kolonne 3, 2024 = kolonne 13
- K2 A6: radetiketter verifisert (kode «A-N» og navn «NÆRINGSLIVET TOTALT»): rad 10/159/60. Kode og navn stemmer i alle tre blokker

API: Kolonneforskyvning finnes ikke som feiltype ved API, men en beslektet feilkilde gjør det: SSB returnerer kategoriene i tabellens egen rekkefølge, ikke nødvendigvis i den rekkefølgen det spørres om. Verdioppslaget bygges derfor på svarets egen dimensjonsrekkefølge, og responssjekken sammenlikner kategoriene strukturelt. Uten dette kunne verdier blitt tilordnet feil kategori, uten feilmelding. Også dette er en KI-generert kontroll.

Kontroll K3 – radvise kontroller (delsummer, aggregater, underposter)

Tre feiltyper kontrolleres (KI-genererte kontroller):

1. **Komponentsummer mot publiserte totaler.** For hver figur summeres komponentene og sammenliknes med det publiserte totaltallet. Avviker de, er en post utelatt eller dobbelttelt. 2. **Aggregat/komponent-klassifisering.** Tabellene inneholder både hovednæringsaggregater (A-N, C, G-N, B05-B09, B06_B09.1) og delkomponenter (C26.3, C26.5, C30.1, C32.5, J58.2). Aggregater og delkomponenter holdes utenfor enkeltnæringsfigurene, med ett unntak (figur 1.2i). Tilsvarende for finansiering og innkjøp, der underposter er merket og holdt utenfor hovedkildesummene. 3. **Restposter beregnes, ikke antas bort.** «Øvrig finansiering fra utlandet» publiseres ikke separat og framkommer som differansen mellom «Utlandet» og de tre underpostene. Restposten beregnes eksplisitt i stedet for å ignoreres.

Totalt 71 kontroller er kjørt: 50 nullkontroller (komponentsummer mot publiserte totaler), 8 forventede restposter, og resten strukturkontroller.

Resultat: Alle summerer som de skal, ingen av 50 nullkontroller avviker mer enn avrundingstoleransen på 0,3 mill. kr. Restposter og strukturkontroller med forventet verdi ulik null er gjengitt i tabell 3.

Tabell 3. Loggførte kontroller.

Kontroll	Resultat	Enhet	Forklaring
A6: implisitt deflator (løpende/faste) mot kjedet 09170	0.0004	indekspunkt	To uavhengige veier til samme deflator.
A6: «indeks»-faktor mot kjedet 09170	0.0004	indekspunkt	A6 oppgir tre desimaler; avvik på dette nivået er avrunding.
A6: realendring beregnet av fastprisserien mot A6s egen publiserte realendring	0.0	prosentpoeng	A6 har egen blokk «FASTE PRISER / Prosent endring»; brukt som uavhengig radkontroll.
A6 mot tabell 11010, løpende priser (største avvik)	0.4	mill. kr	≤ 0,5 (11010 oppgir hele mill.)
A6 mot tabell 11010, faste priser (største avvik)	8.0	mill. kr	A6-fanen «indeks» opplyser at faktoren for 2014 er avrundet til tre desimaler, mens statistikkbanken bruker flere desimaler for dette året.
A6 mot tabell 11010, faste priser, 2015–2024	0.6	mill. kr	Utenom 2014 er seriene sammenfallende.
K3 Sum publiserte enkeltnæringer industri – aggregat (C), 2024	-593.9	mill. kr	Restposten svarer til næringer uten publisert verdi: Møbelindustri (C31), Annen industri (C32). Disse inngår i aggregatet, men kan ikke vises enkeltvis.
K3 Sum med komponenten B06_B09.1 i stedet for aggregatet – hovednæringen, 2024	-49.1	mill. kr	Viser hvorfor figur 1.2d må bruke aggregatet: med komponenten Utvinning av råolje og naturgass og utvinningstjenester alene mangler øvrig bergverksdrift, og postene summerer ikke til hovednæringen.
K3 Aggregat B05-B09 – komponent B06_B09.1 (øvrig bergverksdrift), 2024	49.0	mill. kr	Bekrefter at B05-B09 er et aggregat OVER B06_B09.1. Figur 1.2i viser komponenten «Utvinning av råolje og naturgass og utvinningstjenester»; aggregatet holdes utenfor.
K3 figur 1.2d bruker aggregatet «Bergverksdrift og utvinning» (B05-B09)	1		Figuren bryter hovednæringen «andre næringer» ned i bestanddeler; aggregatet skal stå som én post slik at delene summerer til hovednæringen.
K3 figur 1.2d har ikke også komponenten B06_B09.1	1		Aggregat og komponent skal aldri stå sammen i samme figur.
K3 Delkomponent J58.2 holdt utenfor (J58 – J58.2), 2024	142.0	mill. kr	Bekrefter at J58.2 «Utgivelse av programvare» er en delkomponent av J58.
K3 «Utlandet» – (C1+C2+C3) = øvrig utland, 2023 (1.2g)	48.6	mill. kr	«Øvrig finansiering fra utlandet» publiseres ikke separat; beregnes som restpost.
K3 «Utlandet» – (C1+C2+C3) = øvrig utland, 2024 (1.2g)	52.2	mill. kr	«Øvrig finansiering fra utlandet» publiseres ikke separat; beregnes som restpost.
K3 figur 1.2i bruker KOMPONENTEN B06_B09.1, ikke aggregatet B05-B09	1		Bare én av dem skal være med, ellers dobbelttelles utvinning.
«Fra utlandet» – (D1+D2+D3), 2023 (1.2j)	0.0	mill. kr	≈ 0 eller liten restpost
«Fra utlandet» – (D1+D2+D3), 2024 (1.2j)	0.1	mill. kr	≈ 0 eller liten restpost

K3 – spesielt om oljeutvinning i figur 1.2d og 1.2i

B05-B09 «Bergverksdrift og utvinning» er et aggregat som inneholder B06_B09.1 «Utvinning av råolje og naturgass og utvinningstjenester». De to må aldri stå sammen i samme figur, men hvilken av dem som skal brukes, avhenger av hva figuren viser (tabell 4):

Tabell 4. Aggregat vs komponenter hovednæring B.

Figur	Bruker	Begrunnelse
1.2d	**aggregatet** B05-B09	Figuren bryter hovednæringen «andre næringer» ned i sine bestanddeler. Med aggregatet summerer postene til hovednæringen; med komponenten alene mangler øvrig bergverksdrift.
1.2i	**komponenten** B06_B09.1	Figuren viser enkeltnæringer over en terskel, og det er utvinningsnæringen kapitteltekstene omtaler.

Summekontrollen bekrefter skillet: med aggregatet summerer postene i 1.2d til hovednæringen (avvik −0,1

mill. kr, som er avrunding), mens de med komponenten i stedet ville manglet 49,1 mill. kr., altså nettopp differansen mellom aggregatet og komponenten.

Kontroll K4 – validering av figurene mot arbeidsboken

Figurene generert her ble brukt som støtte i arbeidet med kapittelet. De publiserte figurene er laget i redaksjonen, basert på figurgrunnlag tilsendt fra SSB.

Figurbyggingen valideres i et eget steg. Kontrollene i K1–K3 gjelder dataene; K4 gjelder det som faktisk blir tegnet[2]. Se detaljer i tabell5.

Tabell 5. Kontroll av figurer mot figurgrunnlag.

Kontroll	Hva den sjekker
K4.1	Kategoriantall: hver figur har forventet antall kategorier
K4.2	Obligatoriske kategorier er til stede
K4.3	Figurens datapunkter finnes igjen i arbeidsboken
K4.3b	Figur 1.2j: de tegnede postene summerer til publisert total
K4.4	Prosentstablede figurer summerer til 100
K4.5	PNG-filene er lesbare og har reelt innhold

De forventede kategoriantallene er utledet av 2025-kapittelets tekst, ikke av dataene selv. Ellers ville kontrollen vært sirkulær og aldri kunne slå ut. Resultat: 0 feil av 43 kontroller. Dette tjener også som en ekstra kontroll av figurgrunnlagene.

K4 Kontroll av deflatoren

Faste priser er beregnet som løpende / deflator, med deflator fra tabell 09170 kjedet til 2015 = 1. Deflatoren er kontrollert mot tre uavhengige kilder, se tabell 6:

Tabell 6. Kontroll av implisitt deflator mot beregnet deflator mot statistikkbanktabell 09170.

År	Kjedet fra tabell 09170	Implisitt i A6 (løpende/faste)	A6-arkets «indeks»-faktor
2014	0.9737	0.9740	0.9740
2015	1.0000	1.0000	1.0000
2016	1.0220	1.0220	1.0220
2017	1.0414	1.0410	1.0410
2018	1.0727	1.0730	1.0730
2019	1.1038	1.1040	1.1040
2020	1.1214	1.1210	1.1210
2021	1.1427	1.1430	1.1430
2022	1.1919	1.1920	1.1920
2023	1.2789	1.2790	1.2790
2024	1.3428	1.3430	1.3430

A6-arket inneholder FoU-utgiftene både i løpende og faste priser; forholdet mellom dem er den implisitte deflatoren som faktisk er brukt. Den oppgitte «indeks»-faktor i A6 er en tredje kilde. Alle tre faller sammen innenfor avrunding (A6 oppgir tre desimaler).

A6-fanen «indeks» opplyser: «Oppdatert april 2026».

Arket har dessuten denne merknaden (påpekt av KI):

NB: 02.juni 2026 ble faktor for 2014 avrundet til tre desimaler. I statistikkbank ligger fastpristall for 2014 beregnet med faktor som hadde flere desimaler. Dette skal rettes opp ved nærmeste anledning, men får ingen merkbare konsekvenser for analyse.

Det forklarer avviket mot statistikkbanken for 2014, som det også framgår av K2 (tabell 3).

[2] De endelige figurene bygges i redaksjonen, basert på et manuelt kontrollert figurgrunnlag

Tabell 7. FoU-utgifter i faste priser lest fra regnearket (tabellvedlegg A6) vs statistikkbanktabell 11010.

År	Faste priser, A6	Faste priser, tabell 11010	Avvik
2014	25463.96	25472.0	-8.04 ←
2015	27782.40	27782.0	+0.40
2016	28854.40	28854.0	+0.40
2017	30729.88	30730.0	-0.12
2018	30520.22	30520.0	+0.22
2019	32072.55	32072.0	+0.55
2020	32895.90	32896.0	-0.10
2021	33512.77	33513.0	-0.23
2022	35900.00	35900.0	+0.00
2023	36782.17	36782.0	+0.17
2024	34724.05	34724.0	+0.05

Seriene er sammenfallende for 2015–2024. Avviket i 2014 er en ren avrundingseffekt og har ingen betydning for analysen av 2024.

K5. Løpende dialog

Før kjøringer ga KI tilbakemelding om hvordan oppdraget som spesifisert av forfatter var oppfattet. Forfatter verifiserte eller korrigerte/presiserte. Tilbakemeldinger ble også produsert løpende under kjøring, og det ble generert loggfiler.

Kontroll etter prosessen

Hovedforfatters kontroll av resultatene begrenset seg til rent strukturelle sjekkpunkter som at figurgrunnlagene i tabellformat hadde riktige dimensjoner og kolonneoverskrifter og omfattet alle variabler, samt vurdering av de kontrollene som var utført av (med) KI selv; sjekk av innlesingssteget i Python-koden og sjekk av loggfiler med resultatet, visuell kontroll av diagrammer og faglig kontroll av at diagrammer samsvarte kvalitativt med kjente resultater allerede publisert av SSB.

Deretter ble figurgrunnlagene og tall som er brukt i kapitteltekstene vurdert av to medarbeidere som i to omganger gikk i gjennom tallene etter beste evne.

Kort om KI-støttet analyse

KI ble brukt til å beskrive tendenser og sammenligne tall på tvers av figurene, som hjelp til å identifisere både felles trender og unntak. Videre ble KI brukt som støtte for å vurdere usikkerheten i FoU-tallene, i lys av den sterke og brede nedgangen i egenutført FoU til faste priser fra 2023 til 2024 i kontrast til de senere års realvekst. At «tallene er usikre» påpekes rutinemessig i Indikatorrapportens FoU-omtaler, men usikkerheten blir som regel ikke tallfestet. Som en del av vurderingene i årets næringslivskapittel ble KI brukt til å kode og estimere 2 hovedtyper statistiske modeller for utviklingen i næringslivets FoU-utgifter som en ikke-stasjonær prosess med trend, etter forfatterens spesifikasjoner: Én for FoU uten andre forklaringsvariabler enn en lagget verdi av seg selv, altså en autokorrelert prosess, og én med BNP (faste 2023-priser omregnet til 2015-priser) som driver av FoU-trendvekst (også i en modell med autokorrelasjon). Begge varianter bekrefter at FoU utvikler seg trendmessig, med svingninger rundt en bane med positiv realvekst. Usikkerhet ble vurdert ved residualene sammenlignet med et konfidensintervall på 2 x standardfeilen. Begge modeller peker i retning av at man skal være tilbakeholden med å legge for mye vekt på endringer fra år til år, de ligger stort sett innenfor det man kan forvente gitt historikken. For noen kategorier av foretak (gruppen 10-49 sysselsatte) er nedgangen fra 2023 til 2024 likevel påfallende stor. Neste årgang(er) av Indikatorrapporten vil gi bedre grunnlag for å trekke konklusjoner om dette. Resultatene ga generelt støtte til tydelige forbehold om observerte endringer fra år til år, og som er tatt inn i kapitteltekstene, men selve analysen er ikke skrevet inn i kapittelet. Modellspesifikasjonen eksekvert som

Python-kode er gjengitt i boks 2, fra loggfilen (den økonometriske modellen er bestemt av forfatter, koden er generert av KI). Fullstendig kode (datainnhenting, datavalidering og koding av empirisk modell) kan fås på forespørsel.

Boks 2. Metadata for KI-støttet analyse. Fra loggfil.

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
FoU-utgifter mot BNP-trend. KUN BNP som driver.
BEST-PRACTICE-VERSJON med validering av API-respons og inndata.
Fire avhengige variabler:
  1) Hele næringslivet
  2) Foretak 10-49 sysselsatte
  3) Foretak 50+ sysselsatte
  4) IT-tjenester (NACE J62)
Modell (alt i faste 2015-priser):
  Trend:    ln FoU_t = a + b*ln BNP_t + u_t     (b = langtidselastisitet)
  Dynamikk:  u_t     = c + rho*u_{t-1} + eps_t    (ren AR(1) paa avviket)
  Halveringstid for avviket = ln(0.5)/ln(rho)
Datakilder:
  - fou_lang_11010_*.csv      : FoU næringsliv totalt, løpende OG faste 2015-priser
  - bnp_09189_*.csv           : BNP Fastlands-Norge, faste 2023-priser (-> faste 2015 x 0.78971)
  - FoU_detaljert_07967_*.csv  : FoU etter næringsgruppe x sysselsettingsgruppe (løpende)
  - SSB PxWebApi 07965        : FINANSIERINGS-tabell; "Finansiering" (i alt) = samlet FoU.
                                Brukes til IT-tjenester (J62).
Hva "best practice" betyr her:
  * HTTP-statuskode sjekkes eksplisitt (ikke bare .read()).
  * JSON-strukturen valideres før den tolkes.
  * Hentede serier valideres: ikke tom, numeriske positive verdier, forventet
    årsspenn, ingen hull. Feil stopper kjøringen HØYT (raise) i stedet for aa gli
    videre som en tom/halv serie inn i estimeringen.
  * Alle inndata-serier logges (år-spenn, antall punkter).
```