UHASSELT

KNOWLEDGE IN ACTION

Faculteit Bedrijfseconomische Wetenschappen

master handelsingenieur in de beleidsinformatica

Masterthesis

Evaluation of modern tools for data scientists

Ruben Hermans

Scriptie ingediend tot het behalen van de graad van master handelsingenieur in de beleidsinformatica

PROMOTOR :

Prof. dr. Koenraad VANHOOF

2024
2025

UHASSELT

KNOWLEDGE IN ACTION

www.uhasselt.be
Universiteit Hasselt
Campus Hasselt:
Martelarenlaan 42 | 3500 Hasselt
Campus Diepenbeek:
Agoralaan Gebouw D | 3590 Diepenbeek

Faculteit Bedrijfseconomische Wetenschappen

master handelsingenieur in de beleidsinformatica

Masterthesis

Evaluation of modern tools for data scientists

Ruben Hermans

Scriptie ingediend tot het behalen van de graad van master handelsingenieur in de beleidsinformatica

PROMOTOR :

Prof. dr. Koenraad VANHOOF

Evaluatie van AI-tools voor Data Science Workflows: Een Vergelijking van ChatGPT, Claude, DeepSeek en Grok

Ruben Hermans

Universiteit Hasselt, Faculteit Bedrijfseconomische wetenschappen, Agoralaan Gebouw D, Diepenbeek, 3590, Limburg, België

Abstract

Met de opkomst van grote taalmodellen (LLMs) worden steeds meer onderdelen van data-analyse geautomatiseerd, al is het onduidelijk hoe nauwkeurig deze modellen zijn bij inhoudelijk complexe taken. Dit onderzoek evalueert in welke mate generatieve artificiële-intelligentietools (AI-tools) kunnen fungeren als betrouwbare assistenten binnen de voorbereidende fases van data-analyse, met name data cleaning en exploratieve data-analyse (EDA). Vier actuele AI-modellen — ChatGPT (GPT-4o), Claude (3.7 Sonnet), DeepSeek (V3) en Grok (3) — werden geëvalueerd op hun vermogen om code te genereren, visuele patronen te interpreteren en statistische tabellen te analyseren. De evaluatie gebeurde aan de hand van zorgvuldig opgebouwde testcases in R, gebaseerd op de Titanic- en Palmer Penguins-datasets, en werd gescoord op nauwkeurigheid, aangevuld met een kwalitatieve beoordeling van bruikbaarheid, volledigheid of redeneringsdiepte, afhankelijk van de aard van de taak. De resultaten tonen aan dat generatieve AI-tools fundamentele data cleaning- en EDA-taken in veel gevallen accuraat kunnen uitvoeren, hoewel de kwaliteit sterk varieert per tool en per taak. ChatGPT en Claude toonden sterke prestaties op alle vlakken, met Claude als uitschieter in redenering en volledigheid. DeepSeek leverde technisch correcte en gestructureerde antwoorden, maar had moeite met visuele interpretatie. Grok presteerde goed bij numerieke taken, maar was minder consistent bij grafische analyses en complexe cleaningtaken. De verschillen tussen de tools kwamen vooral tot uiting in de bruikbaarheid, volledigheid en redeneringsdiepte van de output. Hoewel het onderzoek werd uitgevoerd met slechts twee datasets en zonder intersubjectieve triangulatie, bieden de bevindingen waardevolle inzichten in de praktische bruikbaarheid van AI-assistentie bij data-analyse. Verdere studies kunnen onder meer focussen op iteratieve interactie, een bredere datasetvariatie en vergelijking met menselijke output.

1 Inleiding

De opkomst van generatieve artificiële intelligentie (AI) heeft de manier waarop data scientists hun werk organiseren en uitvoeren fundamenteel veranderd [33]. Grote taalmodellen (LLMs) zoals ChatGPT (OpenAI), Claude (Anthropic), DeepSeek (open-source) en Grok (xAI) openen nieuwe mogelijkheden om repetitieve of cognitief intensieve taken binnen de data-analyseworkflow te automatiseren [13]. Deze modellen kunnen natuurlijke taal begrijpen en genereren, waardoor ze inzetbaar zijn voor uiteenlopende toepassingen zoals het schrijven van code [34], het interpreteren van grafieken of het analyseren van tabellen [23]. Binnen data science nemen voorbereidende stappen zoals data cleaning en exploratieve data-analyse (EDA) traditioneel een aanzienlijk deel van de tijd en expertise in beslag [33]. Het correct opschonen van datasets, het detecteren van patronen en het trekken van voorlopige conclusies vormen immers de basis waarop verdere modellering wordt gebouwd [33]. Door AI-tools in te schakelen bij deze taken, kan het analyseproces niet alleen versneld worden, maar ook toegankelijker worden gemaakt voor gebruikers met beperkte programmeer- of statistische achtergrond [13].

Ondanks de toenemende aanwezigheid van generatieve AI binnen data science, blijven diepgaande vergelijkende evaluaties van recente modellen schaars [9]. De markt evolueert snel: taalmodellen

worden frequent geüpdatet of vervangen, terwijl de literatuur zich vaak nog baseert op oudere of minder gangbare systemen [24]. Bovendien zijn bestaande studies vaak beperkt in reikwijdte [22] of niet expliciet afgestemd op de voorbereidende fases van data-analyse [33]. Als gevolg ontbreekt een actueel en onderbouwd inzicht in de inhoudelijke capaciteiten, beperkingen en toepassingsdomeinen van de tools die vandaag effectief in gebruik zijn. Een systematische vergelijking van deze modellen, gericht op hun prestaties binnen realistische workflows, is dan ook wenselijk.

Deze masterproef beoogt een inhoudelijke evaluatie van vier veelgebruikte generatieve AI-tools — ChatGPT, Claude, DeepSeek en Grok — binnen twee fundamentele stappen van de data science workflow: data cleaning en EDA. De vergelijking wordt uitgevoerd aan de hand van realistische testcases in R, toegepast op de Titanic- en Palmer Penguins-datasets. Binnen drie taaktypes — codegeneratie, visuele interpretatie en numerieke analyse — wordt nagegaan in welke mate de gegenereerde output van de tools overeenkomt met vooropgestelde modeloplossingen. De evaluatie is primair kwantitatief van aard, met nadruk op nauwkeurigheidsscores, en wordt aangevuld met een kwalitatieve bespreking van bruikbaarheid, volledigheid of redeneringsdiepte. Op die manier biedt dit onderzoek een onderbouwd beeld van hoe deze tools presteren in realistische analysetaken en in hoeverre ze inhoudelijk waardevolle ondersteuning kunnen bieden. De centrale onderzoeksvraag luidt als volgt: In welke mate kunnen generatieve AI-tools functioneren als inhoudelijk betrouwbare assistenten tijdens de voorbereidende fases van data science, met name data cleaning en EDA?

Deze studie beperkt zich tot de voorbereidende fases van de data science workflow, met name data cleaning en EDA. Latere stappen zoals modelselectie, modeltraining of implementatie vallen buiten het onderzoeksdomein, gezien hun sterk contextafhankelijke invulling en beperkte standaardisering. De focus ligt uitsluitend op de inhoudelijke kwaliteit van de gegenereerde output: hoe accuraat, volledig en inhoudelijk onderbouwd zijn de gegenereerde oplossingen in verhouding tot vooraf gedefinieerde benchmarks. Niet-inhoudelijke aspecten zoals gebruiksvriendelijkheid, snelheid of interactiemogelijkheden worden buiten beschouwing gelaten. Deze afbakening laat toe om de tools gericht te beoordelen op hun potentieel als inhoudelijke ondersteuning bij analytische kerntaken.

Deze thesis bestaat uit zes hoofdstukken. In hoofdstuk 1 wordt de algemene context van het onderzoek geschetst, gevolgd door de probleemstelling, doelstelling en afbakening. Hoofdstuk 2 presenteert een literatuurstudie over de inzet van generatieve AI in data science, met bijzondere aandacht voor toepassingen in data cleaning en EDA. Hoofdstuk 3 beschrijft de methodologie, inclusief het onderzoeksopzet, onderzoeksontwerp en de evaluatiemethode. De resultaten worden gepresenteerd in hoofdstuk 4, gestructureerd per taaktype en AI-tool. In hoofdstuk 5 volgt een reflectie op de belangrijkste bevindingen, samen met de beperkingen van het onderzoek en suggesties voor vervolgstudies. Hoofdstuk 6 sluit af met de algemene conclusie en plaatst de resultaten binnen het bredere kader van AI-integratie in data-analysepraktijken.

2 Literatuurstudie

In dit hoofdstuk wordt bestaande literatuur besproken over de rol van generatieve AI in data science, met focus op de voorbereidende stappen data cleaning en EDA. Sectie 2.1 bespreekt het bredere gebruik van LLMs binnen data science, gevolgd door een overzicht van de vier onderzochte tools in sectie 2.2. De secties 2.3 en 2.4 behandelen respectievelijk eerdere studies rond AI-ondersteuning bij data cleaning en bij EDA. Samen vormen deze onderdelen de inhoudelijke basis voor het onderzoek in hoofdstuk 3.

2.1 Generatieve AI binnen data science

Generatieve AI, en in het bijzonder LLMs, heeft de afgelopen jaren snel aan belang gewonnen binnen uiteenlopende domeinen, waaronder softwareontwikkeling [32], communicatie [27], onderwijs [28] en data-analyse [33]. LLMs zoals ChatGPT, Claude, DeepSeek en Grok zijn in staat om syntactisch correcte en semantisch plausibele tekst te genereren op basis van natuurlijke taalprompts [19].

Deze capaciteit maakt hen inzetbaar voor tal van taken waarin natuurlijke taal, programmeertaal en analytische redenering samenkomen [19].

Binnen data science opent dit mogelijkheden om onderdelen van de analyseworkflow te automatiseren of te versnellen [13]. In principe kunnen LLMs ondersteuning bieden tijdens alle fases van de cyclus — van data cleaning en exploratie tot modellering en rapportering — maar in de praktijk worden ze het vaakst ingezet bij afgebakende, voorbereidende taken [33]. Voorbeelden zijn het genereren van R- of Python-code [34], het interpreteren van grafieken of het analyseren van tabellen [23]. Net die voorbereidende stappen, zoals data cleaning en EDA, nemen traditioneel veel tijd in beslag en vormen voor minder ervaren gebruikers een inhoudelijke of technische drempel [33]. Door deze taken (deels) te automatiseren, kunnen AI-tools de drempel tot data-analyse verlagen [13].

Tegelijkertijd zijn er fundamentele vragen over de betrouwbaarheid van dergelijke AI-systemen in inhoudelijk gevoelige contexten. De studie van Khatun et al. [15] wijst op de gevoeligheid van gegenereerde output voor de formulering van prompts, de beschikbare context en het onderliggende model. Zeker in analytische toepassingen — waar interpretatiefouten of foutieve veralgemeningen snel tot verkeerde conclusies kunnen leiden — blijft een kritische blik op de output noodzakelijk volgens Sclar et al. [31]. Deze bezorgdheid vormt een belangrijk uitgangspunt voor de verdere analyse in deze studie.

2.2 Beschrijving van de geselecteerde AI-tools

In deze studie worden vier recente en veelgebruikte generatieve AI-tools geëvalueerd: ChatGPT, Claude, DeepSeek en Grok. Deze selectie is gebaseerd op hun actualiteit, brede beschikbaarheid, en het feit dat ze representatief zijn voor verschillende ontwikkelmodellen (commercieel, open-source, geïntegreerd) [17]. In wat volgt wordt per tool kort het ontwikkelkader, de technische basis en de beoogde functionaliteit geschetst.

ChatGPT is een generatief taalmodel ontwikkeld door OpenAI, gebaseerd op de GPT-4-architectuur. Het model is toegankelijk via een webinterface en API, en ontworpen voor een brede waaier aan toepassingen in natuurlijke taalverwerking en programmeerondersteuning. ChatGPT ondersteunt meerdere programmeertalen, waaronder R en Python, en wordt vaak ingezet in contexten waar geautomatiseerde tekstgeneratie, codeproductie of analyseondersteuning vereist is [25].

Claude, ontwikkeld door Anthropic, bouwt voort op een eigen modelreeks (Claude 1 t.e.m. 3). De tool is ontworpen rond het principe van ”constitutional AI”, waarbij veiligheid, transparantie en redeneringskwaliteit centraal staan. Claude wordt beschikbaar gesteld via een chatinterface, en sinds Claude 3 ook via API’s. Het model is ontworpen om natuurlijke taal op een betrouwbare en contextbewuste manier te verwerken, met toepassingen in tekstgeneratie, analyse, codering en complexe instructievolging [1].

DeepSeek is een open-source taalmodel ontwikkeld door het gelijknamige onderzoeksproject DeepSeek, dat zich richt op toepassingen in natuurlijke taalverwerking en programmeren. De huidige versie, DeepSeek-V3, wordt vrij beschikbaar gesteld via platforms zoals GitHub en Hugging Face, en kan zowel via API als lokaal worden ingezet. Het model onderscheidt zich door een expliciete focus op efficiëntie, prestaties en toegankelijkheid, en positioneert zich als een kosteneffectief alternatief voor commerciële LLMs zoals die van OpenAI. Dankzij het open-source karakter en de lage gebruiksdrempel is DeepSeek bijzonder geschikt voor onderzoekers en ontwikkelaars die flexibiliteit en transparantie belangrijk vinden bij het inzetten van taalmodellen in technische contexten [20].

Grok is een taalmodel ontwikkeld door xAI, het AI-onderzoeksbedrijf van Elon Musk, en is rechtstreeks geïntegreerd in het X-platform (voorheen Twitter). De huidige versie, Grok-3, maakt deel uit van een gesloten modelarchitectuur met focus op actuele informatieverwerking via koppeling aan het X-platform. Daarnaast biedt het model standaardondersteuning voor natuurlijke taalverwerking, programmeertalen en vraaggestuurde data-analyse. Door zijn directe toegang tot real-time context vormt Grok een uniek modeltype, al is de inzetbaarheid voor data science-toepassingen minder uitgebreid gedocumenteerd dan bij andere AI-tools. Desondanks blijft het een relevant model vanwege zijn actieve ontwikkeling en uitgesproken positionering binnen het generatieve AI-landschap [35].

Deze vier tools vertegenwoordigen samen een mix van commerciële en open-source benaderingen, variërend in toegankelijkheid, doelpubliek en focus. Hun beoordeling in deze studie gebeurt op basis van outputkwaliteit binnen concrete data science-taken, onafhankelijk van interfacevoorkeur of gebruikerservaring.

2.3 Generatieve AI voor data cleaning

Data cleaning is een essentiële voorbereidende stap binnen elke data-analyseworkflow [6]. Ze omvat onder meer het detecteren en behandelen van ontbrekende waarden, het corrigeren van foutieve of inconsistente invoer, het herstructureren van variabelen, en het standaardiseren van formaten of typen [6]. Een grondige opschoning van data is cruciaal om latere analyses betrouwbaar en reproduceerbaar te maken [6]. Tegelijk is het een proces dat vaak repetitief, technisch en tijdsintensief is, zeker voor datasets met ongestructureerde of onvolledige informatie [18].

Generatieve AI-tools bieden potentieel om dit proces te versnellen door automatisch codevoorstellen te doen op basis van natuurlijke taalbeschrijvingen van het probleem. Zo tonen zowel Bien en Mukherjee [3] als Bray [4] aan dat gebruikers zonder programmeerkennis met behulp van modellen zoals ChatGPT functionele R- of Python-code kunnen laten genereren voor uiteenlopende cleaning-taken, wat de instapdrempel voor data-analyse aanzienlijk verlaagt. Naast basisbewerkingen tonen systemen zoals RetClean aan dat LLMs ook complexere cleaning-taken aankunnen, waaronder het aanvullen van ontbrekende waarden [22]. Door relevante informatie op te halen uit externe databronnen, zoals data lakes, kunnen deze modellen nauwkeurige en contextafhankelijke correcties uitvoeren, wat ze bijzonder geschikt maakt voor gebruik in situaties waar dataprivacy of domeinspecifieke kennis vereist is [22].

Hoewel er in toenemende mate onderzoek wordt verricht naar de inzet van generatieve AI voor cleaning-doeleinden, blijft het aantal diepgaande, systematische evaluaties beperkt. Zo illustreren Bien en Mukherjee [3] en Bray [4] dat taalmodellen eenvoudige cleaning-taken kunnen uitvoeren, maar hun studies richten zich voornamelijk op educatieve toepassingen en geven weinig inzicht in de robuustheid of inhoudelijke correctheid van de gegenereerde code. Daarnaast toont Naeem et al. [22] aan dat LLMs ook complexere taken kunnen aanpakken via retrieval-gebaseerde technieken, maar biedt voornamelijk een systeembeschrijving zonder brede, modelvergelijkende evaluatie. Ten slotte maken die studies gebruik van oudere modelversies of minder gangbare systemen, waardoor de prestaties van recente tools onderbelicht blijven.

Deze studie beoogt die drie tekortkomingen aan te pakken door generatieve AI-tools systematisch te vergelijken zoals Oni et al. [24], toegepast op actuele en populaire tools zoals ChatGPT, Claude, DeepSeek en Grok. De evaluatie gebeurt aan de hand van concrete, representatieve R-opdrachten, met expliciete aandacht voor de inhoudelijke juistheid van de gegenereerde code en de mate waarin deze bruikbaar is in realistische data cleaning-scenario's.

2.4 Generatieve AI voor EDA

EDA is een kernonderdeel van elke data science workflow en vormt de brug tussen ruwe data en statistische modellering [16]. Tijdens EDA worden beschrijvende statistieken berekend, worden verdelingen en relaties onderzocht, en wordt gezocht naar patronen, uitschieters of dataproblemen [16]. Zowel numerieke als visuele technieken spelen hierin een rol: van samenvattingstabellen en frequentieverdelingen tot boxplots, scatterplots of correlatiematrices [16]. Het doel is om inzicht te verwerven in de structuur en kenmerken van de dataset [16].

Generatieve AI-tools kunnen het proces van EDA toegankelijker maken, zo blijkt uit de studie van DeJeu [9]. Zij toont aan dat dergelijke modellen op basis van natuurlijke taal effectieve visualisaties kunnen genereren, wat het EDA-proces zowel vereenvoudigt als versnelt [9]. Hierdoor daalt de instapdrempel voor gebruikers met beperkte technische of statistische voorkennis aanzienlijk [9]. Daarnaast bevorderen deze tools de ontwikkeling van data-geletterdheid, doordat ze gebruikers ondersteunen bij het herkennen van patronen, structuren en inzichten in gegevens [9]. Ook uit de

studie van Zhecheva [37] blijkt dat generatieve AI binnen de bedrijfswereld aanzienlijke meerwaarde biedt. Zij stelt dat het geen kwestie meer is óf organisaties deze technologie moeten inzetten, maar hoe ze dat het best doen [37]. Door EDA-processen te versnellen, dragen LLMs bij aan snellere besluitvorming, efficiënter databeheer en betere schaalbaarheid [37]. Voor bedrijven die hun data-analyse willen professionaliseren, wordt de inzet van generatieve AI dan ook steeds vaker gezien als een strategische noodzaak [37].

Hoewel bovenstaande studies aantonen dat generatieve AI-tools in staat zijn om grafieken te genereren op basis van natuurlijke taal, blijft de kwaliteit van de bijbehorende tekstuele interpretaties vaak beperkt. Zo stelt DeJeu [9] vast dat de verklaringen die ChatGPT Plus geeft bij de grafieken en tabellen vaak oppervlakkig zijn en niet altijd inhoudelijk correct, waardoor verdere analyse of bijsturing door de gebruiker noodzakelijk blijft. Bovendien werd deze studie uitgevoerd met de betalende versie van het model, waardoor de prestaties van gratis toegankelijke alternatieven onbelicht blijven [9]. Zowel DeJeu [9] als Zhecheva [37] richten zich uitsluitend op ChatGPT, waardoor er geen inzicht is in hoe andere generatieve AI-tools zich verhouden op vlak van nauwkeurigheid, volledigheid of interpretatievermogen. Een systematische vergelijking van meerdere modellen ontbreekt voorlopig, terwijl dat juist essentieel is om de betrouwbaarheid en meerwaarde van deze technologieën in uiteenlopende EDA-contexten te kunnen beoordelen [37].

Deze studie vult de bestaande lacunes aan door een systematische analyse uit te voeren van de tekstuele beschrijvingen die generatieve AI-tools genereren tijdens EDA. De vergelijking omvat meerdere actuele en gratis toegankelijke tools, om na te gaan in hoeverre deze modellen bruikbare ondersteuning bieden in realistische en breed toegankelijke analysetoepassingen. De focus van de analyse ligt op de nauwkeurigheid, volledigheid en diepgang van de gegenereerde beschrijvingen. Om een volledig beeld te krijgen van het interpretatievermogen van deze modellen, worden zowel visuele (grafieken) als numerieke (tabellen) EDA-output onderzocht. Deze opsplitsing wordt ook aangehouden in het resultatenhoofdstuk.

3 Methodologie

In dit hoofdstuk wordt de methodologische opzet van het onderzoek toegelicht. De focus ligt op de manier waarop de vier geselecteerde AI-tools — ChatGPT, Claude, DeepSeek en Grok — geëvalueerd werden op hun inhoudelijke prestaties binnen data cleaning en EDA. Na een toelichting van het algemene onderzoeksontwerp wordt per luik beschreven welke cases gebruikt werden, hoe de input werd opgebouwd en hoe de output van de AI-tools werd beoordeeld. Het hoofdstuk sluit af met een overzicht van de gehanteerde evaluatiemethode, inclusief de scoresystematiek en maatregelen om de reproduceerbaarheid en betrouwbaarheid van de beoordeling te waarborgen.

3.1 Onderzoeksopzet

Het doel van dit onderzoek is om te evalueren in welke mate vier generatieve AI-tools — ChatGPT, Claude, DeepSeek en Grok — kunnen fungeren als inhoudelijk betrouwbare assistenten voor data scientists. De evaluatie richt zich op twee fundamentele stappen in de data science workflow: data cleaning en EDA. Hierbij staat centraal hoe accuraat de output is die deze tools genereren of interpreteren in realistische cases die een data scientist in de praktijk zou kunnen tegenkomen.

Het onderzoek is bewust afgebakend tot twee specifieke stappen in de workflow: data cleaning en EDA. Latere stappen, zoals modeltraining, hyperparameteroptimalisatie of deployment, worden buiten beschouwing gelaten. De keuze hiervoor is tweeledig: enerzijds ligt de grootste behoefte aan betrouwbare AI-assistentie net in deze voorbereidende fases, anderzijds zijn deze fases inhoudelijk goed afgebakend en relatief onafhankelijk van modelkeuze of domeinspecifieke doelstellingen [33]. Binnen data cleaning worden typische opschoontaken onderzocht zoals het behandelen van ontbrekende waarden en duplicaten. Binnen EDA worden zowel visuele als numerieke interpretaties geëvalueerd.

Naast de inhoudelijke focus op cleaning en EDA, is ook de evaluatievorm bewust afgebakend tot het criterium nauwkeurigheid. Performantie-aspecten zoals snelheid of consistentie van output werden niet geëvalueerd, aangezien die moeilijk objectief vergelijkbaar zijn door factoren zoals netwerkvertragingen, serverbelasting of interfaceverschillen (bv. API vs. chatomgeving). Ook gebruiksvriendelijkheid werd niet als officieel evaluatiecriterium gehanteerd, omdat dit sterk afhankelijk is van subjectieve factoren zoals voorkennis en persoonlijke voorkeur. Wel werd dit aspect — waar relevant — kwalitatief meegenomen in de beschrijvende bespreking, om stilistische of communicatieve verschillen tussen de tools aan te duiden. De kern van dit onderzoek blijft echter de inhoudelijke correctheid van de gegenereerde of geïnterpreteerde output.

De vier onderzochte AI-tools zijn gekozen omwille van hun actualiteit, functionaliteit en complementariteit. ChatGPT (OpenAI) is de meest gevestigde tool, gekend om zijn brede toepasbaarheid en sterke codegeneratie [25]. Claude (Anthropic) focust op ethisch redeneren en lange contextverwerking [1]. DeepSeek is een technisch geavanceerd model met een open-source achtergrond en een specifieke focus op programmeertaken [20]. Grok (xAI) integreert actuele informatie via X (het voormalige Twitter) en profileert zich als een assistent met directe toegang tot recente context [35]. Voor elk van deze tools werd de gratis beschikbare versie gebruikt: GPT-4o [26], Claude 3.7 Sonnet [2], DeepSeek-V3 [8] en Grok 3 [36]. Door deze selectie wordt een representatief staal gevormd van de huidige generatie LLMs met toepassingen in data science.

Voor de evaluatie zijn twee datasets gebruikt: de Titanic dataset[1] en de Palmer Penguins dataset[2]. De Titanic dataset werd uitsluitend gebruikt in de cleaningcases, gezien de aanwezigheid van typische onvolkomenheden zoals duplicaten, ontbrekende waarden en inconsistente types. De Penguins dataset werd in alle visuele en numerieke EDA-cases toegepast, omwille van de aanwezigheid van meerdere continue en categorische variabelen over drie pinguïnsoorten. Die structuur maakt het mogelijk om variatie, groepsverschillen, verdelingen en correlaties visueel én statistisch te analyseren.

De gekozen opzet — vier tools, twee datasets, en drie luiken (data cleaning, visuele EDA, numerieke EDA) — laat toe om AI-tools te evalueren op drie complementaire vaardigheden: syntactisch correcte en logische codegeneratie (cleaning), visuele patroonherkenning (EDA visueel), en statistisch redeneren (EDA numeriek). Zo biedt het onderzoek een afgebakend maar diepgaand kader voor het beoordelen van de toegevoegde waarde van generatieve AI binnen de voorbereidende fases van data-analyse.

3.2 Onderzoeksontwerp

Om de centrale onderzoeksvraag te beantwoorden, werd een onderzoeksontwerp opgesteld dat de AI-tools uitdaagt op drie uiteenlopende cognitieve vaardigheden: het genereren van correcte code (data cleaning), het interpreteren van visuele patronen (visuele EDA) en het analyseren van statistische tabellen (numerieke EDA). Voor elk van deze luiken werden representatieve testcases geformuleerd, telkens met een heldere prompt in natuurlijke taal, een vooraf bepaalde modeloplossing en een consistente evaluatiemethode. Op die manier kon de accuraatheid van de gegenereerde of geïnterpreteerde output gecontroleerd en reproduceerbaar beoordeeld worden.

3.2.1 Data cleaning

Voor de data cleaning-cases werd telkens gewerkt met een vaste steekproef van de originele Titanic-dataset. Er werd een representatieve subset van 200 observaties genomen via set.seed(), zodat elke case op identieke en reproduceerbare data kon worden getest. De testcases bevatten telkens een specifieke manipulatie, afhankelijk van de beoogde vaardigheid:

- Case 1 – Verwijderen van duplicaten: De eerste 20 rijen van de steekproef werden gedupliceerd en toegevoegd onderaan de dataset, wat resulteerde in exact 20 volledige dubbele rijen. *Doel:* testen of AI-tools volledige rijen konden identificeren en verwijderen.

[1] Titanic-dataset beschikbaar via Kaggle: https://www.kaggle.com/competitions/titanic/data
[2] Penguins-dataset beschikbaar via Kaggle: https://www.kaggle.com/datasets/larsen0966/penguins

- Case 2 – Aanvullen van ontbrekende waarden: De oorspronkelijke steekproef bevatte reeds NA's in de kolom Age (totaal 42 NA's). In de kolom Embarked kwam slechts één ontbrekende waarde voor; daarom werden er met set.seed() nog 9 bijkomende NA's willekeurig geïntroduceerd. De AI moest NA's in Age imputeren met de mediaan en die in Embarked met de modus.

- Case 3 – Foutieve invoer detecteren en verwijderen: Uit de steekproef werden eerst alle rijen met NA's verwijderd, wat leidde tot een subset van 158 observaties. Daarna werden 10 ongeldige waarden (zoals "???", "missing", "/") willekeurig ingevoerd in de kolom Age, opnieuw via set.seed(). De bedoeling was dat de AI deze foute strings zou detecteren, verwijderen en vervolgens de kolom correct zou converteren naar integer.

- Case 4 – Informatie extraheren uit tekstvelden: In deze case werd geen enkele aanpassing aan de data gedaan. De originele Name-kolom werd behouden, met als opdracht aan de AI om een nieuwe kolom Title te creëren op basis van titels zoals "Mr.", "Mrs.", "Miss.", enz.

De gemanipuleerde datasets werden per case afzonderlijk opgeslagen als CSV-bestand, en samen met de opdracht meegegeven aan de AI-tools. Die opdracht werd telkens geformuleerd in natuurlijke taal, met een korte contextbeschrijving en een duidelijke taakomschrijving, zoals: "Verwijder de volledig dubbele rijen uit de dataset." of "Vervang NA's in kolom Age door de mediaan." Voor elke case werd daarnaast ook een modeloplossing geformuleerd in R, waarin de verwachte code en output werden vastgelegd. De gegenereerde output van de AI-tools werd geëvalueerd op correctheid (gescoord van 0 tot 2), en aangevuld met een kwalitatieve bespreking van de volledigheid en bruikbaarheid van de gegenereerde code.

3.2.2 Visuele exploratieve data-analyse

Het tweede luik onderzoekt in hoeverre AI-tools in staat zijn om visuele informatie te interpreteren zoals die wordt weergegeven in grafieken. Hiervoor werd de Palmer Penguins-dataset gebruikt, die morfologische kenmerken bevat van drie pinguïnsoorten in Antarctica, aangevuld met informatie over geslacht en leefgebied. Voorafgaand aan de visualisaties werd de dataset gepreprocessed: enkel observaties zonder ontbrekende waarden werden behouden (n = 333). De grafieken werden opgesteld in R met behulp van ggplot2, waarbij consistente kleurcodering per soort en vormcodering per geslacht werd toegepast om multivariabele interpretatie mogelijk te maken. Elke grafiek werd opgeslagen als PNG-bestand met behulp van de functie ggsave(), zodat deze eenduidig en reproduceerbaar kon worden meegegeven aan de AI-tools.

Deze PNG-bestanden werden vervolgens samen met een bijhorende prompt in natuurlijke taal aan de AI-tools meegegeven. Die prompts introduceerden de grafiek en formuleerden één of meerdere gerichte interpretatievragen, zoals: "Welke soort is het zwaarst?", "Wat zegt het verschil tussen mediaan en gemiddelde over de verdeling?", of "Welke groepen vertonen overlap of scheiding?". Bij elke case werd ook een modelantwoord geformuleerd met de belangrijkste inzichten die minimaal aanwezig moesten zijn in een correcte interpretatie.

De output van de AI-tools werd beoordeeld op nauwkeurigheid (gescoord van 0 tot 2), gebaseerd op de mate waarin de centrale observaties uit de grafiek correct werden benoemd en verklaard. Daarnaast werd per tool een kwalitatieve bespreking opgesteld op basis van twee aanvullende dimensies: volledigheid (hoeveel relevante elementen werden benoemd) en redeneringsdiepte (in hoeverre werden observaties verklaard of in context geplaatst). Zo kon per case zowel de feitelijke juistheid als het interpretatieve vermogen van elke tool worden vergeleken.

Onderstaande cases illustreren de vijf interpretatievaardigheden die binnen visuele EDA worden bevraagd, gaande van eenvoudige categorische analyses tot multivariabele patronen en correlaties:

- Case 1 - Categorische verdelingen herkennen: een barplot van het aantal observaties per soort. *Doel:* test of de AI het dominante aantal observaties per pinguïnsoort correct identificeert en relatieve verschillen duidt.

- Case 2 - Centrum en spreiding interpreteren: een boxplot van flipperlengte per soort. *Doel:* evalueert of de AI verschillen in mediaan, interkwartielafstand en outliers kan herkennen en verklaren.
- Case 3 - Verdeling en scheefheid herkennen: een density plot van lichaamsgewicht per soort. *Doel:* meet het vermogen van de AI om verdelingsvormen, overlappingen en relatieve verhoudingen te interpreteren.
- Case 4 - Multivariabele patronen analyseren: een scatterplot van snavellengte versus flipperlengte, gecodeerd per soort (kleur) en geslacht (vorm). *Doel:* beoordeelt of AI simultaan met kleur- en vormcodering kan omgaan en patronen correct interpreteert.
- Case 5 - Correlaties duiden: een correlatiematrix van geslacht en fysieke kenmerken. *Doel:* test of de AI richting en sterkte van correlaties correct inschat en of deze biologisch worden geïnterpreteerd.

Deze indeling wordt consistent aangehouden in de resultatenhoofdstuk, waar elke case afzonderlijk wordt besproken.

3.2.3 Numerieke exploratieve data-analyse

Het derde luik test de capaciteit van AI-tools om statistische samenvattingen correct te interpreteren. Hiervoor werd opnieuw de Palmer Penguins-dataset gebruikt, zij het in tabelvorm. Voor elke case werd een specifieke samenvatting gegenereerd in R, waarbij telkens per groep of over meerdere variabelen beschrijvende statistieken werden berekend. In totaal werden acht cases voorbereid, verspreid over vier categorieën: centrale tendens en spreiding, multivariabele groepsvergelijking, correlaties, en toetsende statistiek (t-test).

De tabellen omvatten onder meer: gemiddelden en mediaanwaarden per soort of geslacht, interkwartielafstanden en standaarddeviaties, minimum- en maximumwaarden, en correlatiematrices van continue variabelen. Ook werden frequentietabellen en de output van een t-test tussen geslachten bij één soort meegegeven als input. Sommige cases gebruikten dezelfde tabel, maar met een andere interpretatieve focus. De tabellen werden zorgvuldig opgebouwd met enkel de relevante variabelen, zodat de focus lag op interpretatie en redenering.

De tabellen werden telkens gegenereerd in R, waarbij beschrijvende statistieken werden berekend met behulp van tidyverse-functies zoals group_by() en summarise(). Om de resultaten leesbaar en visueel overzichtelijk te maken, werden ze opgemaakt met behulp van het pander-pakket. Dankzij deze opmaak konden de tabellen in een nette en uniforme tekstvorm worden gekopieerd en geplakt in de prompts van de AI-tools, zonder verlies aan structuur of interpretatiegemak. Hierdoor konden de tools zich volledig richten op de inhoudelijke analyse, zonder dat ze zelf nog structuur of indeling moesten afleiden uit ongestructureerde tekst.

Elke case werd begeleid door een heldere prompt in natuurlijke taal, waarin expliciet werd gevraagd naar één of meerdere conclusies op basis van de gegeven cijfers uit de meegegeven tabellen. Voorbeelden zijn: "Welke soort heeft gemiddeld het hoogste lichaamsgewicht, en komt dit overeen met de mediaan?", "In welke soort is het verschil tussen mannetjes en vrouwtjes het grootst in relatieve termen?" of "Waarom zegt de negatieve correlatie tussen snavellengte en snaveldiepte niet dat een langere snavel automatisch minder diep is?"

Voor elke case werd ook een modelantwoord geformuleerd, waarin werd vastgelegd welke inzichten minimaal correct, relevant en cijfermatig onderbouwd moesten zijn. De gegenereerde output werd geëvalueerd op nauwkeurigheid (gescoord van 0 tot 2), en daarnaast kwalitatief beoordeeld op volledigheid en redeneringsdiepte, net zoals in het visuele EDA luik.

Onderstaande cases zijn thematisch gegroepeerd volgens de interpretatievaardigheden die ze beogen te testen binnen numerieke EDA:

- Case 1 – Centrale tendens en spreiding: een tabel met het gemiddelde en de mediaan van het lichaamsgewicht per pinguïnsoort. *Doel:* test of de AI verschillen tussen soorten correct identificeert en interpreteert op basis van centrum- en spreidingsmaten.
- Case 2 – Centrale tendens en spreiding: een tabel met gemiddelde, standaarddeviatie en interkwartielafstand van de flipperlengte per geslacht. *Doel:* evalueert of AI verschillen in gemiddelde waarden en spreiding tussen mannetjes en vrouwtjes correct kan duiden.
- Case 3 – Multivariabele groepsvergelijking: een tabel met gemiddelde lichaamsmassa per soort en geslacht. *Doel:* beoordeelt of AI absolute en relatieve verschillen tussen subgroepen correct kan berekenen en verklaren.
- Case 4 – Multivariabele groepsvergelijking: een tabel met de minimum- en maximumwaarden van snavellengte per soort en geslacht. *Doel:* test of AI het grootste absolute en relatieve verschil correct detecteert op basis van de ranges.
- Case 5 – Multivariabele groepsvergelijking: een tabel met Q1- en Q3-waarden van snaveldiepte per soort en geslacht. *Doel:* evalueert of AI kwartielwaarden correct rangschikt en interpreteert in context.
- Case 6 – Correlatie-inzicht en associatie: een correlatiematrix van lichaamsmassa en andere kenmerken. *Doel:* test of AI de sterkste positieve correlatie correct aanwijst en plausibel biologisch verklaart.
- Case 7 – Correlatie-inzicht en associatie: een correlatiematrix met een negatieve correlatie tussen snavellengte en snaveldiepte. *Doel:* test of AI betekenisvol kan redeneren over negatieve correlaties en foutieve interpretaties vermijdt.
- Case 8 – Statistische significantie: output van een t-test op lichaamsmassa bij mannelijke en vrouwelijke Adelie-pinguïns. *Doel:* evalueert of AI de p-waarde correct interpreteert en statistische significantie nauwkeurig weet te duiden.

Deze indeling wordt ook in het resultatenhoofdstuk gehanteerd, waarbij per categorie de prestaties van de AI-tools systematisch geanalyseerd worden.

3.3 Evaluatiemethode

De evaluatie van de gegenereerde output van de AI-tools gebeurde op systematische wijze, volgens een vooraf vastgelegde beoordelingsprocedure en aan de hand van transparante scoringsmodellen. De beoordeling werd uitgevoerd door de onderzoeker zelf, met als doel een consistente en inhoudelijk onderbouwde evaluatie te garanderen. Alle cases werden in één batch geëvalueerd per luik, zodat het beoordelingskader coherent toegepast kon worden binnen eenzelfde denkkader en zonder beïnvloeding van eerdere scores. Hoewel de evaluatie door één persoon gebeurde en dus geen intersubjectieve triangulatie werd toegepast, werd dit risico op subjectiviteit opgevangen door het gebruik van vaste evaluatieschema's, modeloplossingen en dubbele controle bij twijfelgevallen.

Elke testcase was gekoppeld aan een modeloplossing, opgesteld op basis van gangbare praktijken binnen data-analyse en data cleaning. Deze modeloplossing fungeerde als inhoudelijke referentie voor de beoordeling. Om subjectiviteit te beperken, werd de beoordeling uitsluitend gebaseerd op de aanwezigheid of afwezigheid van elementen uit deze modeloplossing. Er werd niet geëvalueerd op schrijfstijl of uitvoerige formulering, zolang de inhoud correct, relevant en interpreteerbaar was. Voor elk luik werd een specifiek scoremodel gehanteerd:

- Data cleaning: scores van 0 (foutief), 1 (gedeeltelijk correct), of 2 (volledig correct). De beoordeling werd aangevuld met een kwalitatieve bespreking van de volledigheid en bruikbaarheid van de gegenereerde code.

- Visuele EDA: beoordeling nauwkeurigheid (gescoord op een schaal van 0 tot 2), en daarnaast kwalitatieve bespreking op volledigheid (hoeveel relevante elementen werden benoemd) en redeneringsdiepte (in hoeverre observaties werden verklaard of in context geplaatst).
- Numerieke EDA: beoordeling op nauwkeurigheid (eveneens op een 0–2-schaal), aangevuld met een kwalitatieve bespreking van volledigheid en redeneringsdiepte, conform het kader van de andere luiken.

Deze schaal werd bewust eenvoudig gehouden om het beoordelingsproces reproduceerbaar te maken en ruis door nuanceverschillen te vermijden. In een vroege fase van het onderzoek werd ook geprobeerd om aanvullende scores toe te kennen aan volledigheid en bruikbaarheid (bij data cleaning), en aan volledigheid en diepgang (bij EDA). Al snel bleek echter dat deze aspecten moeilijk objectief te scoren waren, waardoor ze verder kwalitatief besproken werden. Door de scores per criterium afzonderlijk toe te kennen, ontstond een gedifferentieerd maar transparant beeld van de prestaties van elke tool.

Om de betrouwbaarheid van de evaluatie te versterken, werd er gewerkt met vaste scoreformats per case. Tijdens het scoren werd gebruikgemaakt van een evaluatiematrix in Excel, waarin per tool en per case de nauwkeurigheid werd gescoord en bijkomende observaties werden genoteerd over volledigheid, bruikbaarheid of redeneringsdiepte. Na het toekennen van de scores werd een controle uitgevoerd op consistentie tussen vergelijkbare cases. Wanneer twijfel bestond over een score, werd de modeloplossing opnieuw geraadpleegd, en werd de prompt en tooloutput herlezen in de oorspronkelijke context.

Ten slotte werd ervoor gezorgd dat de prompts, modeloplossingen en outputdocumenten systematisch gearchiveerd werden per tool en per luik. Deze documentatie maakt het mogelijk om het volledige evaluatieproces te reconstrueren en vormt een belangrijke waarborg voor de transparantie en reproduceerbaarheid van het onderzoek.

Hoewel deze methode systematisch en onderbouwd werd opgezet, zijn er ook enkele beperkingen. Zo werd de evaluatie uitgevoerd door slechts één onderzoeker, wat onvermijdelijk een zekere graad van interpretatie met zich meebrengt. Daarnaast werd per case slechts één prompt geformuleerd, waardoor variatie in promptformulering buiten beschouwing bleef. Tot slot werd de evaluatie beperkt tot de inhoud van de eerste gegenereerde output, zonder opvolgingsinteractie met de tool. Deze keuzes verhogen de controleerbaarheid, maar beperken ook de volledigheid van de evaluatiecontext. Deze beperkingen worden in het discussiehoofdstuk verder besproken.

Door te werken met realistische testcases, gecontroleerde input, vooraf gedefinieerde modeloplossingen en een transparant scoringssysteem, biedt dit onderzoeksontwerp een gestructureerd en herhaalbaar kader voor de vergelijking van vier generatieve AI-tools binnen twee fundamentele stappen van data-analyse. In het volgende hoofdstuk worden de resultaten van deze evaluatie gepresenteerd, gestructureerd per taaktype en aangevuld met een bespreking van nauwkeurigheid, volledigheid en redeneringsdiepte.

4 Resultaten

In dit hoofdstuk worden de resultaten gepresenteerd van het vergelijkend onderzoek naar de nauwkeurigheid van de vier generatieve AI-tools — ChatGPT, Claude, DeepSeek en Grok — binnen twee cruciale stappen van de data science workflow: data cleaning en EDA. De evaluatie gebeurde aan de hand van zorgvuldig ontworpen testcases, telkens vergezeld van een modeloplossing, duidelijke evaluatiecriteria en een gestructureerde beoordelingsprocedure (zie Methodologie). De resultaten worden opgedeeld in drie hoofdsecties, die elk een specifiek luik van de workflow behandelen:

- Data cleaning richt zich op de nauwkeurigheid van gegenereerde R-code voor het opschonen van ruwe data;
- Visuele EDA onderzoekt hoe nauwkeurig de AI-tools visuele patronen uit grafieken interpreteren;

- Numerieke EDA evalueert het vermogen van de tools om statistische tabellen nauwkeurig te analyseren.

Per luik worden de prestaties van de tools besproken aan de hand van twee evaluatieniveaus. In de eerste plaats is er de nauwkeurigheid, waarbij objectief wordt nagegaan ten opzichte van de model-oplossing of de gegenereerde output van de tools correct is. Daarnaast volgt er ook een kwalitatieve bespreking van aspecten zoals volledigheid, bruikbaarheid en diepgang van de output. Binnen elke sectie worden de cases geordend volgens de onderliggende interpretatievaardigheid, zodat patronen in sterktes en zwaktes van de tools overzichtelijk kunnen worden blootgelegd. Elke sectie wordt afgesloten met een tussentijdse conclusie, waarin de belangrijkste bevindingen worden samengevat en vergeleken over de cases heen. Deze gelaagde benadering biedt een gedetailleerd inzicht in hoe generatieve AI zich gedraagt binnen verschillende cognitieve taken die typerend zijn voor data-analyse in de praktijk.

4.1 Data Cleaning

Data cleaning is het eerste luik dat in dit hoofdstuk aan bod komt, en vormt een essentiële voorbereidende stap binnen elke data-analyseworkflow [6]. Ruwe datasets bevatten vaak inconsistenties, ontbrekende waarden, foutieve types of ongestructureerde tekstvelden die een correcte analyse in de weg kunnen staan [6]. Het adequaat voorbereiden van de data is dan ook een noodzakelijke voorwaarde om betrouwbare en reproduceerbare inzichten te verkrijgen [6]. In deze sectie wordt onderzocht in welke mate LLMs in staat zijn om R code te genereren die correct, volledig en bruikbaar is op basis van een concrete opdracht in natuurlijke taal.

De cleaning-taken in deze evaluatie beslaan vier representatieve cases die verschillende aspecten van data cleaning belichten: het verwijderen van duplicaten, het aanvullen van ontbrekende waarden met statistische kengetallen, het verwerken van niet-converteerbare tekstwaarden in een numerieke kolom, en het extraheren van betekenisvolle informatie uit tekstvelden. De moeilijkheidsgraad varieert van eenvoudig tot meer complex.

Hoewel de evaluatie vooral focust op nauwkeurigheid als hoofdcriterium met scores op 2, wordt in de beschrijving per case ook aandacht besteed aan de mate van volledigheid en bruikbaarheid van de code. Deze factoren hebben namelijk een invloed op de interpretatie en toepasbaarheid van de gegenereerde oplossing in de praktijk.

In de volgende paragrafen worden de vier cases individueel besproken. Elke case bevat een analyse van de gegenereerde oplossingen door vier AI-tools (ChatGPT, Claude, DeepSeek en Grok).

4.1.1 Verwijderen van duplicaten

Het verwijderen van volledig identieke rijen is een van de meest voorkomende eerste stappen in data cleaning [21]. Bij het combineren van databronnen of het verwerken van invoer van gebruikers of externe systemen, komen dubbels vaak voor [21]. Zulke duplicaten kunnen de resultaten van statistische analyses verstoren, leiden tot oververtegenwoordiging van bepaalde observaties of fouten veroorzaken bij het trainen van modellen [21]. In deze case werd geëvalueerd of de AI-tools in staat zijn om in R correct werkende code te genereren die alle volledig identieke rijen verwijdert, op basis van alle kolommen in de dataset. De dataset bevatte opzettelijk enkele duplicaten, die exact overeenkwamen over elke variabele, zoals besproken in de methodologie. De volgende prompt werd aan elk van de AI-tools voorgelegd, samen met de gemanipuleerde CSV-file als input: *"Schrijf R-code die alle rijen verwijdert die volledig identiek zijn aan een andere rij, zodat elke overblijvende rij uniek is op basis van alle kolommen."*

Het doel van deze test was om na te gaan of de tools begrijpen wat bedoeld wordt met een "volledig identieke rij" en of ze daarvoor de juiste functies gebruiken. Er werd niet enkel gekeken naar de correctheid van de gegenereerde code, maar ook naar de volledigheid van de oplossing: beperken de tools zich tot enkel de transformatiestap, of voorzien ze ook in het inlezen van het bestand, het controleren van het resultaat en eventueel het opslaan van de output? Daarnaast werd bruikbaarheid

geëvalueerd: kon de gegenereerde code onmiddellijk gebruikt worden op de meegegeven dataset, zonder bijkomende aanpassingen, en was de output begrijpelijk voor een gebruiker? Hierbij werd ook de aanwezigheid van begeleidende uitleg of samenvattende opmerkingen mee in rekening gebracht.

De verwachte oplossing in deze context bestaat uit het gebruik van de functie unique() of een combinatie van duplicated() met een negatie (!duplicated(data)) om enkel de eerste unieke rijen te behouden. Beide methodes zijn geldig in R. Een correcte oplossing bevat minimaal het inlezen van de dataset, de toepassing van de duplicaatverwijdering, en optioneel het bewaren of inspecteren van het resultaat. Volledige oplossingen geven ook inzicht in de impact van de bewerking, bijvoorbeeld door het aantal verwijderde rijen te tonen of de eerste rijen van de opgeschoonde dataset weer te geven. Tot slot werd een oplossing als optimaal bruikbaar beschouwd indien de gegenereerde code zonder enige aanpassing kon worden gekopieerd en toegepast in een R-script, inclusief het gebruik van de correcte bestandsnaam zoals vermeld in de prompt.

Alle vier de tools – ChatGPT, Claude, DeepSeek en Grok – genereerden correcte R-code die de opdracht inhoudelijk juist uitvoerde. ChatGPT, Claude en DeepSeek gebruikten de functie unique(data) (zie respectievelijk Code 2, 1 en 3 in Bijlage A.1.1), terwijl Grok koos voor data[!duplicated(data),] (zie Code 4 in Bijlage A.1.1). Opmerkelijk is dat DeepSeek ook een alternatieve methode voorstelde met dplyr::distinct(), waarmee eveneens unieke rijen behouden blijven (zie Code 5 in Bijlage A.1.1). Alle benaderingen leiden tot hetzelfde resultaat en werden aanvaard als correcte oplossing. In termen van correctheid scoorden alle tools 2 op 2.

Op het vlak van volledigheid waren er enkele verschillen merkbaar. Claude, DeepSeek en Grok voegden controle-uitvoer toe, bijvoorbeeld door het aantal rijen voor en na de verwijdering van duplicaten te tonen. ChatGPT deed dit niet en beperkte zich tot het uitvoeren van de transformatie en het printen van het resultaat. Opvallend is dat alle vier de tools gebruikmaakten van write.csv() om de opgeschoonde dataset weg te schrijven, wat positief bijdroeg aan de volledigheid van de oplossing. Alle tools lazen het bestand correct in met read.csv(), met uitzondering van Grok, die de bestandsnaam in het inleescommando generiek hield en niet aanpaste aan de concrete bestandsnaam zoals in de prompt was opgegeven (zie Code 4 in Bijlage A.1.1).

Deze tekortkoming in de bestandsnaam bij Grok had ook gevolgen voor de bruikbaarheid van de gegenereerde oplossing. Waar de andere tools code genereerden die zonder aanpassingen kon worden gekopieerd en uitgevoerd, vereiste Grok dat de gebruiker de bestandsnaam nog manueel corrigeerde. ChatGPT en Claude leverden relatief bondige output, met een minimale hoeveelheid commentaar. Grok en DeepSeek daarentegen voegden onder het codeblok expliciet een toelichting toe, met het label "Uitleg:" waarin kort werd uitgelegd wat de code deed (zie toelichtingen in Bijlage A.1.1). Dit verhoogt de bruikbaarheid voor gebruikers met minder ervaring in R, en draagt bij aan de begrijpelijkheid van de gegenereerde oplossing.

Deze case bevestigt dat de AI-tools bijzonder betrouwbaar zijn in het herkennen en oplossen van eenvoudige, goed gedefinieerde cleaning-taken in R. Zowel de syntactische uitvoering als de praktische toepasbaarheid waren in vrijwel alle gevallen foutloos. Bovendien werd duidelijk dat er meerdere geldige manieren zijn om duplicaten te verwijderen, en dat de tools elk zelfstandig tot een correcte methode kwamen. Dit wijst op een voldoende diep ingebed begrip van de functie van deze bewerking in de R-taal en binnen de bredere context van data cleaning. Het is een eerste indicatie dat voor standaard cleaning-taken, nauwkeurigheid nauwelijks een onderscheidende factor is tussen tools.

4.1.2 Aanvullen van ontbrekende waarden

Het aanvullen van ontbrekende waarden is een fundamentele stap binnen data cleaning, aangezien NA's een belemmering vormen voor veel standaardanalyses en machine learning-algoritmes [29]. Een vaak toegepaste techniek is het imputeren van missende waarden met behulp van samenvattende statistieken zoals de mediaan of de modus [11]. In deze case werd geëvalueerd of de AI-tools in staat zijn om in R code te genereren die ontbrekende waarden in een numerieke kolom (Age) correct aanvult met de mediaan, en in een categorische kolom (Embarked) met de modus. Deze

combinatie van numerieke en niet-numerieke variabelen maakt de opdracht iets complexer dan een standaardvervanging met één type waarde. De volgende prompt werd gebruikt voor deze case en meegegeven aan elke tool: *"Schrijf R-code om ontbrekende waarden enkel in de kolommen 'Age' en 'Embarked' aan te vullen: gebruik de mediaan voor 'Age' en de meest voorkomende waarde (modus) voor 'Embarked'. Laat de andere kolommen ongewijzigd."*

Het doel van deze test was om na te gaan of de tools de juiste strategieën toepassen voor het imputeren van ontbrekende waarden, afhankelijk van het datatype van de kolom. Daarbij werd gekeken naar het correct gebruik van median() met na.rm = TRUE voor de numerieke kolom, en naar een geldige implementatie van een modusberekening voor de categorische kolom. Verder werd gelet op volledigheid (zoals het controleren van de hoeveelheid missende data voor en na de bewerking, en het wegschrijven van het resultaat), en op bruikbaarheid: kon de gegenereerde code meteen worden gebruikt zonder aanpassingen, en werd er voldoende uitleg of context voorzien?

De verwachte oplossing omvat het inlezen van de dataset, het berekenen van de mediaan van Age met uitsluiting van NA's (na.rm = TRUE), en het berekenen van de modus van Embarked, bijvoorbeeld via een zelfgeschreven get_mode()-functie of via table() in combinatie met which.max(). Daarna moeten beide kolommen worden aangevuld, en idealiter volgt een controle op het resultaat, samen met een eventuele write.csv() om het eindresultaat op te slaan. Een oplossing werd als bruikbaar beschouwd indien ze zonder aanpassingen kon worden uitgevoerd op de meegegeven dataset, inclusief correcte bestandsnaam.

Alle vier de tools gebruikten de juiste methode om ontbrekende waarden in de kolom Age aan te vullen, namelijk via de functie median() in combinatie met na.rm = TRUE. Voor het aanvullen van Embarked met de modus varieerde de aanpak. ChatGPT en Grok gebruikten een aangepaste get_mode()-functie waarin de modus werd berekend op basis van unique(), tabulate() en which.max() (zie respectievelijk Code 7 en Code 9 in Bijlage A.1.2). Claude gebruikte een eenvoudiger alternatief door table() en which.max() rechtstreeks toe te passen, zonder deze logica in een aparte functie te gieten (zie Code 6 in Bijlage A.1.2). DeepSeek gebruikte evenmin een aparte functie, maar sorteerde de frequentietabel (table()) in aflopende volgorde via decreasing = TRUE en selecteerde vervolgens de eerste waarde met "[1]" (zie Code 8 in Bijlage A.1.2). Deze aanpak vermijdt which.max() en levert eveneens het correcte resultaat op. In termen van correctheid leverden alle tools een geldige en foutloze oplossing met score 2 op 2.

Wat betreft volledigheid viel op dat enkel Claude expliciete tussenresultaten toonde, waaronder het aantal ontbrekende waarden in Age en Embarked vóór en na de imputatie. De andere tools – ChatGPT, Grok en DeepSeek – beperkten zich tot een eenvoudige summary() of head() van het eindresultaat. Wat het in- en wegschrijven betreft: Claude gebruikte zowel read.csv() als write.csv() en bewaarde dus de opgeschoonde dataset, terwijl ChatGPT, Grok en DeepSeek enkel read.csv() toepasten. Opnieuw viel bij Grok op dat de bestandsnaam in het read.csv()-commando niet overeenkwam met de prompt en handmatig aangepast moest worden, wat de volledigheid licht negatief beïnvloedde (zie Code 9 in Bijlage A.1.2).

De bruikbaarheid van de gegenereerde code was over het algemeen hoog: alle oplossingen werkten correct en konden grotendeels zonder aanpassingen worden uitgevoerd. Grok vereiste wel dat de gebruiker eerst de bestandsnaam aanpaste in het inleescommando. Claude leverde de meest toegankelijke output, met zowel duidelijke uitleg onder het codefragment als extra verduidelijking bij de tussenresultaten (zie toelichtingen in Bijlage A.1.2). Grok en DeepSeek gaven eveneens toelichting onder de gegenereerde code, telkens in de vorm van een beknopte samenvatting van de werkwijze (zie toelichtingen in Bijlage A.1.2). ChatGPT was het meest beknopt en leverde enkel het noodzakelijke, zonder extra uitleg of interpretatie.

Deze case toont aan dat de onderzochte AI-tools betrouwbaar kunnen omgaan met eenvoudige imputatietaken, zelfs wanneer verschillende datatypes en bijhorende strategieën gecombineerd worden. De oplossingen zijn niet alleen correct, maar ook – op één uitzondering na – volledig toepasbaar zonder verdere tussenkomst. Er was geen enkele tool die systematisch tekortschiet op inhoudelijk vlak. De grootste verschillen situeren zich in de stijl van uitleg, de mate van begeleiding, en kleine

details in het formatteren van de invoer. Ook hier blijkt dat nauwkeurigheid als differentiërende factor beperkt is binnen deze categorie van cleaning-taken.

4.1.3 Foutieve invoer detecteren en verwijderen

In realistische datasets komt het vaak voor dat numerieke waarden zoals leeftijden foutief als tekst worden opgeslagen [7]. In zulke kolommen kunnen zich ook onconventionele waarden bevinden, zoals ”?”, `"unknown"` of ”missing”, die niet rechtstreeks naar een getal kunnen worden geconverteerd [7]. In deze case werd getest of de AI-tools in staat zijn om een kolom Age, die als tekst werd ingelezen, correct om te zetten naar een geheel getal, en om rijen waarin dit niet mogelijk is, te verwijderen. Hiervoor werd de volgende prompt gebruikt binnen deze case: *”Schrijf R-code om ‘Age’ (ingevoerd als tekst) om te zetten naar een geheel getal. Rijen met ongeldige waarden die niet geconverteerd kunnen worden, moeten worden verwijderd.”*

Het doel van deze test was na te gaan of de tools de juiste typeconversie kunnen toepassen – van character naar integer – en foutieve of niet-converteerbare waarden correct kunnen identificeren en uitsluiten. Er werd niet geëist dat de tools specifieke ongeldige waarden vooraf detecteerden of uitsloten. Het volstond dat zij as.integer() gebruikten en rijen met NA verwijderden als gevolg van een mislukte conversie. Tools die toch expliciet foutieve tekstwaarden vervingen of afronding toepasten, werden als grondiger of robuuster beschouwd, maar dit werd niet vereist om als correct beoordeeld te worden.

De verwachte oplossing bestond uit het inlezen van de dataset, het converteren van de Age-kolom naar integer en het verwijderen van rijen waarvoor de waarde niet kon worden geïnterpreteerd als geheel getal (via is.na()). Het tonen van het resultaat via summary(), str() of head() was een pluspunt, net als het wegschrijven van de opgeschoonde dataset. De correcte bestandsnaam werd eveneens mee in overweging genomen voor de bruikbaarheid van de oplossing.

Alle tools pasten een correcte typeconversie toe, zij het op verschillende manieren. ChatGPT gebruikte rechtstreeks as.integer() op de kolom Age, vertrouwde op de automatische omzetting van niet-converteerbare invoer naar NA, en verwijderde die rijen met is.na() (zie Code 11 in Bijlage A.1.3). Deze aanpak is functioneel, maar minder robuust: tekstuele waarden en decimalen worden zonder waarschuwing omgezet naar NA of afgekapt, zonder zicht op de aard van de foutieve invoer. Grok volgde een iets explicietere strategie door eerst as.numeric() toe te passen en pas daarna te converteren met as.integer(), wat de detectie van niet-numerieke waarden transparanter maakt (zie Code 13 in Bijlage A.1.3). In beide gevallen werd de invoerproblematiek impliciet opgevangen, zonder specifieke controle op het type vervuiling. Claude en DeepSeek kozen voor een iets uitgebreidere aanpak. Claude verving met gsub() verschillende gekende foutwaarden zoals `"--"`, ”?”, ”NaN”, ”###” en ”missing” vooraf door NA, en verwijderde daarna de NA-rijen (zie Code 10 in Bijlage A.1.3). DeepSeek definieerde een aparte conversiefunctie (convert_age()) waarin foutieve waarden werden afgehandeld met ifelse(), en waarin ook negatieve leeftijden werden uitgesloten (zie Code 12 in Bijlage A.1.3). Deze aanpakken bieden meer controle over de aard van de invoerfouten en maken de onderliggende vervuiling expliciet zichtbaar, wat de transparantie en robuustheid van de data cleaning verhoogt. In termen van correctheid waren alle oplossingen geldig, aangezien ze resulteerden in een kolom met correcte gehele waarden en zonder foutieve rijen. Bijgevolg behaalden ze allen een score van 2 op 2.

Wat betreft volledigheid waren Claude en DeepSeek het meest uitgebreid. Claude toonde het aantal verwijderde rijen en de resterende observaties na filtering. DeepSeek leverde een robuuste oplossing via een aparte functie, voerde afronding uit naar hele getallen en filterde negatieve waarden. ChatGPT en Grok beperkten zich tot de kern: zij voerden de conversie en filtering uit zonder aanvullende controle of rapportering. Enkel Claude gebruikte write.csv() om de aangepaste dataset op te slaan; de andere tools deden dit niet. Alle tools gebruikten read.csv() om de data in te lezen, maar Grok specifieerde de bestandsnaam niet correct, wat een kleine tekortkoming vormt in termen van reproduceerbaarheid.

De bruikbaarheid van de gegenereerde code was over het algemeen hoog: alle oplossingen werkten zonder fouten en konden grotendeels zonder aanpassingen worden uitgevoerd. Claude bood de

meest uitgebreide output, waarbij het boven het codefragment eerst een verkennende analyse uitvoerde om onzuivere waarden in de Age-kolom te identificeren. Onder het codefragment volgde een uitgebreid stappenplan, met commentaar bij elke stap en concrete feedback over het aantal verwijderde rijen (zie toelichtingen in Bijlage A.1.3). DeepSeek gaf eveneens verduidelijking bij de aanpak onder de code waarbij de stappen en de herbruikbare functie werden toegelicht (zie toelichtingen in Bijlage A.1.3). Grok gaf een iets beknoptere uitleg onder de code maar deze was wel correct en overzichtelijk. Wel moest de bestandsnaam opnieuw handmatig worden aangepast in de read.csv()-functie, wat de praktische bruikbaarheid licht verminderde. ChatGPT leverde tot slot de meest bondige output, met een beperkte en minder uitgebreide toelichting.

Deze case toont aan dat alle tools in staat zijn om eenvoudige conversies correct uit te voeren, zelfs wanneer de brondata onregelmatigheden bevat. De kernfunctionaliteit werd bij alle modellen correct toegepast, maar de mate van robuustheid en begeleiding verschilde duidelijk. Tools die extra controlemechanismen inbouwden of feedback gaven over de impact van de filtering, leverden meerwaarde voor de eindgebruiker. Claude en DeepSeek waren in die zin het meest compleet, terwijl ChatGPT en Grok uitgingen van een standaardworkflow zonder veel context of validatie.

4.1.4 Informatie extraheren uit tekstvelden

In veel datasets bevindt zich relevante informatie binnen vrije tekstvelden, bijvoorbeeld in naamvelden waar titels (zoals ”Mr”, ”Miss”, ”Mrs”) vervat zitten [14]. Het extraheren van deze informatie is een typisch voorbeeld van text preprocessing binnen data cleaning, en vereist vaak het gebruik van reguliere expressies (regex) [14]. In deze case werd geëvalueerd of de AI-tools in staat zijn om de titel van een persoon correct te extraheren uit een Name-kolom met gestructureerde tekst in het formaat ”Achternaam, Titel. Voornaam”. De volgende prompt werd aan elk van de AI-tools voorgelegd, samen met de CSV-file als input: *”Schrijf R-code die uit de kolom ‘Name’ de titel (zoals Mr, Miss, Mrs...) extraheert en toevoegt als een nieuwe kolom ‘Title’, terwijl de originele ‘Name’-kolom behouden blijft.”*

Het doel van deze test was om na te gaan of de tools met behulp van een gepaste regex-string de titel konden isoleren en opslaan in een nieuwe kolom Title, zonder de bestaande Name-kolom te wijzigen. Er werd beoordeeld of de gebruikte regex correct was opgebouwd om de substring tussen de komma en het punt te capteren, en of de oplossing toepasbaar was op de meegegeven dataset. Daarnaast werd gekeken naar de volledigheid van de gegenereerde code (zoals het inlezen en controleren van de data), en naar de mate van bruikbaarheid: kon de gegenereerde code zonder aanpassingen worden toegepast, en werd deze voorzien van voldoende uitleg?

De verwachte oplossing bestond uit het inlezen van de dataset, het gebruik van een reguliere expressie met sub() of gsub() om de titel uit de Name-kolom te extraheren, en het toekennen van het resultaat aan een nieuwe kolom Title. Een correcte regex bevat minstens één capture group die de titel selecteert, typisch via een expressie zoals `", (.*?)\\."` of een gelijkwaardige variant. Een optionele controle via head() of summary() was aanbevolen, net als het bewaren van de nieuwe dataset via write.csv().

Alle tools pasten een regex toe die erin slaagde om eenvoudige titels zoals `"Mr"` of `"Mrs"` correct te extraheren. Alle vier de tools maakten hierbij gebruik van `sub()` of `gsub()`, maar verschilden in de manier waarop ze de titel binnen de string isoleerden (zie Code 14, Code 15, Code 16 en Code 17 in Bijlage A.1.4). Claude en Grok gebruikten een patroon dat, volgens hun uitleg, de titel samen met de rest van de string na de komma zou afvangen, zodat ook langere of samengestelde titels correct konden worden geëxtraheerd. In de praktijk faalde Grok echter bij complexere titels bestaande uit meerdere woorden en een spatie, zoals `"the Countess"`, waarbij enkel `"the"` werd geëxtraheerd. Claude voerde deze taak wel correct uit. ChatGPT gebruikte een eenvoudiger patroon dat de substring tussen de komma en de punt correct isoleerde en slaagde erin om ook deze complexe gevallen juist te extraheren. DeepSeek maakte dezelfde fout als Grok, maar vermeldde in zijn uitleg niet expliciet dat het patroon bedoeld was om meerdere woorden of de volledige titel te vangen. In termen van

correctheid scoorden Claude en ChatGPT 2 op 2; Grok en DeepSeek bleven achter met 1 op 2 door de fout bij `"the Countess"`.

Wat betreft volledigheid gebruikten alle tools een `read.csv()`-commando om de dataset in te lezen. Grok specifieerde opnieuw niet de correcte bestandsnaam uit de prompt, net zoals in eerdere cases, en gebruikte een generieke naam (zie Code 17 in Bijlage A.1.4). De andere tools namen de correcte bestandsnaam wel over. Verder toonden alle modellen het resultaat van de extractie via head() of summary(). Enkel DeepSeek voorzag in de mogelijkheid om de aangepaste dataset op te slaan via `write.csv()`, en vermeldde dit als optionele stap. Claude, ChatGPT en Grok boden deze mogelijkheid niet aan.

De bruikbaarheid van de gegenereerde code was over het algemeen hoog: de oplossingen waren grotendeels inhoudelijk correct en konden vrijwel direct worden uitgevoerd. Claude, DeepSeek en Grok voorzagen allemaal in een duidelijke uitleg onder of rond het codefragment, wat de interpretatie en toepasbaarheid ondersteunde (zie toelichtingen in Bijlage A.1.4). ChatGPT leverde een correcte, maar beknoptere oplossing, met minimale begeleidende uitleg. Grok vroeg, net zoals in eerdere cases, nog manuele aanpassing van de bestandsnaam in het `read.csv()`-commando, wat de directe uitvoerbaarheid licht beperkte. Enkel DeepSeek gaf ook de mogelijkheid om de aangepaste dataset op te slaan via `write.csv()`, hoewel dit als optioneel werd vermeld.

Deze case toont aan dat AI-tools in staat zijn om reguliere expressies correct toe te passen, maar dat kleine verschillen in patroondefinitie kunnen leiden tot fouten bij uitzonderlijke inputs. Claude en ChatGPT toonden aan dat zij niet alleen syntactisch juiste regex konden formuleren, maar ook foutgevoelige gevallen correct behandelden. Grok en DeepSeek faalden op dit vlak in minstens één complex voorbeeld, wat het belang onderstreept van grondige controle bij het genereren van text processing-logica via AI.

4.1.5 Tussentijdse conclusie

De vier uitgewerkte cases tonen aan dat AI-tools in staat zijn om typische data cleaning-taken in R op een correcte en bruikbare manier uit te voeren. In eenvoudige opdrachten, zoals het verwijderen van duplicaten of het aanvullen van NA's met standaardstatistieken, slaagden alle tools erin om foutloze code te genereren. Naarmate de cases complexer werden – bijvoorbeeld bij het verwerken van vervuilde tekstdata of het extraheren van informatie via reguliere expressies – kwamen er stilaan verschillen naar voren in de aanpak en foutgevoeligheid van de tools.

Claude leverde in alle cases correcte en goed onderbouwde oplossingen, met veel aandacht voor toelichting, controles en outputopslag. DeepSeek gaf doorgaans nauwkeurige en gestructureerde code, maar maakte in case vier een inhoudelijke fout bij het extraheren van informatie uit tekst. ChatGPT beantwoordde alle opdrachten correct en efficiënt, al bleven zijn oplossingen eerder beknopt en zonder veel extra uitleg of validatie. Grok scoorde correct op drie van de vier cases, maar maakte in case vier dezelfde fout als DeepSeek en bleef in elke case vasthouden aan een generiek read.csv()-commando, wat de directe bruikbaarheid beperkte. Hieronder volgt een samenvattende tabel van de nauwkeurigheidsscores van de tools binnen het luik van data cleaning (Tabel 1).

Case	**Claude**	**ChatGPT**	**Grok**	**DeepSeek**
Case 1	2	2	2	2
Case 2	2	2	2	2
Case 3	2	2	2	2
Case 4	2	2	1	1
Totaal	**8**	**8**	**7**	**7**

Tabel 1: Overzicht scores data cleaning

Hoewel alle tools hoge nauwkeurigheidsscores behaalden, bleek uit de kwalitatieve analyse dat vooral bij complexere cleaning-taken de verschillen in robuustheid, begeleiding en praktische toepasbaarheid meer naar voren kwamen. Deze nuances zijn van belang voor gebruikers die AI-tools inzetten als ondersteuning bij realistische data cleaning-workflows.

4.2 Exploratieve Data Analyse - Visuele Inzichten

Naast data cleaning vormt visuele exploratie van data een tweede essentieel luik binnen het bredere data science-proces [30]. In dit deel van het onderzoek wordt onderzocht in welke mate LLMs in staat zijn om inzichten af te leiden uit grafische voorstellingen. Visuele interpretatievaardigheid is cruciaal om patronen, trends, groepsverschillen en verbanden te herkennen nog vóór er formele statistische modellen worden toegepast — en vormt zo een sleutelfase binnen exploratieve data-analyse [30].

De vijf cases in dit luik bestrijken een breed scala aan grafiektypes, waaronder barplots, boxplots, density plots, scatterplots met meerdere visuele coderingen, en correlatiematrices. In elk van deze grafieken werd aan de tools gevraagd om relevante inzichten te formuleren over verschillen tussen pinguïnsoorten, geslachten of fysieke kenmerken. Dit vereist niet alleen het correct lezen van de grafiekstructuur, maar ook het semantisch duiden van de onderliggende gegevens.

De evaluatie gebeurde uitsluitend op basis van nauwkeurigheid: per case werd beoordeeld of de tool de juiste interpretaties gaf op basis van de visuele informatie, met een score van 0 (foutief), 1 (gedeeltelijk correct) of 2 (volledig correct). Daarnaast werd per case ook de mate van volledigheid, nuance en interpretatieve diepgang besproken, om inhoudelijke verschillen tussen de tools te kunnen duiden. Deze aspecten werden echter niet afzonderlijk gescoord, aangezien ze moeilijk objectief te kwantificeren zijn en niet bepalend zijn voor de correctheid van het antwoord.

De cases worden in chronologische volgorde besproken, van eenvoudig naar complex, en per case wordt de output van de vier onderzochte AI-tools geanalyseerd: ChatGPT, Claude, Grok en DeepSeek.

4.2.1 Categorische verdelingen herkennen

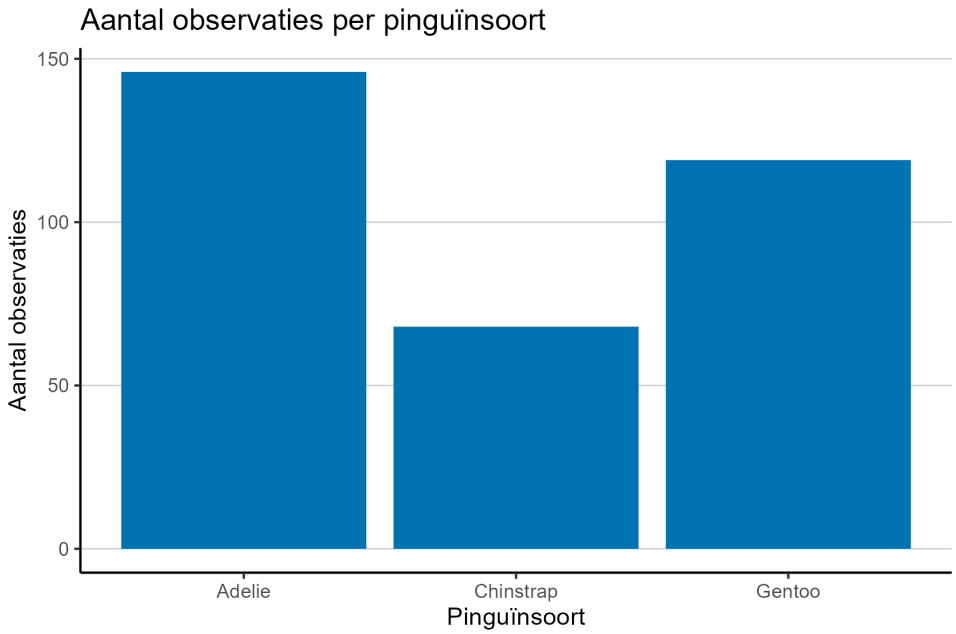
In de eerste case kregen de AI-tools een eenvoudige barplot te zien die het aantal observaties per pinguïnsoort in de Palmer Penguins dataset weergaf (Figuur 1). De drie soorten — Adelie, Chinstrap en Gentoo — werden elk voorgesteld met een verticale balk waarvan de hoogte overeenkomt met het aantal waargenomen individuen. De tools kregen de opdracht om op basis van deze grafiek te beschrijven welke soort het meest en minst voorkwam, en om eventuele opvallende verschillen in verdeling te benoemen. De prompt die met de PNG-afbeelding werd aangeleverd, luidde als volgt: *"Bekijk deze barplot die het aantal observaties per pinguïnsoort toont. Wat valt je op aan de verdeling? Welke pinguïnsoorten zijn het meest of minst vertegenwoordigd?"*

Het herkennen van categorieën en hun relatieve frequentie vormt een basisvaardigheid binnen visuele exploratieve data-analyse [11]. Een correcte lezing van een barplot is essentieel om de steekproefsamenstelling te begrijpen, biases te detecteren, en te kunnen beoordelen of vergelijkingen tussen groepen statistisch zinvol zijn [11]. Als een AI-tool hier al de mist in gaat, is dat een waarschuwingssignaal voor latere interpretatiefouten in meer complexe grafieken.

Een correcte interpretatie van deze grafiek vereist het benoemen van alle drie de soorten, het aanduiden van de soort met het hoogste en laagste aantal observaties, en het correct inschatten van de relatieve verhouding tussen de groepen. Verwarringen met andere grafiektypes (zoals histogrammen of boxplots), of foutieve conclusies duiden op een gebrekkig begrip van de visualisatie.

Claude gaf een volledige en inhoudelijk correcte interpretatie. De tool benoemde alle drie de soorten en schatte het aantal observaties per soort correct in. Bovendien wees Claude erop dat de steekproefverdeling onevenwichtig is en dat dit relevant kan zijn bij latere analyses — een voorbeeld van diepgang en contextbewustzijn. ChatGPT maakte eveneens een correcte inschatting van de verhoudingen en vermeldde expliciet dat Adelie het meest en Chinstrap het minst voorkomt. De drie soorten werden benoemd, al was de samenvatting iets beknopter dan bij Claude. Grok beschreef de algemene trend correct, maar rapporteerde foutieve absolute aantallen: Chinstrap werd onderschat (50 in plaats van ±70), wat de interpretatie inhoudelijk minder accuraat maakt. DeepSeek maakte fundamentele fouten door volledig foutieve aantallen te noemen (bv. 50 voor Adelie, 10 voor Chinstrap), wat duidt op een mislukte extractie of verwarring met een andere visualisatie.

Bij vergelijking tussen de tools behaalden Claude en ChatGPT de maximale score van 2 op 2, met correcte en duidelijke interpretaties van de grafiek. Claude deed dit bovendien met meer context en



Figuur 1: Barplot aantal observaties per pinguïnsoort

analytische reflectie. Grok leverde een gedeeltelijk correcte interpretatie, maar de onnauwkeurigheid van de cijfers maakt dat deze slechts deels betrouwbaar is en resulteert in een score van 1 op 2. DeepSeek slaagde er niet in om de visualisatie correct te interpreteren en scoorde dus niet goed op nauwkeurigheid, namelijk 0 op 2. Wat betreft volledigheid en diepgang sprong Claude eruit met de meest genuanceerde en contextuele beschrijving, gevolgd door ChatGPT. Grok bleef op hoofdlijnen correct, maar mist detail. DeepSeek bleef zowel inhoudelijk als contextueel zwak.

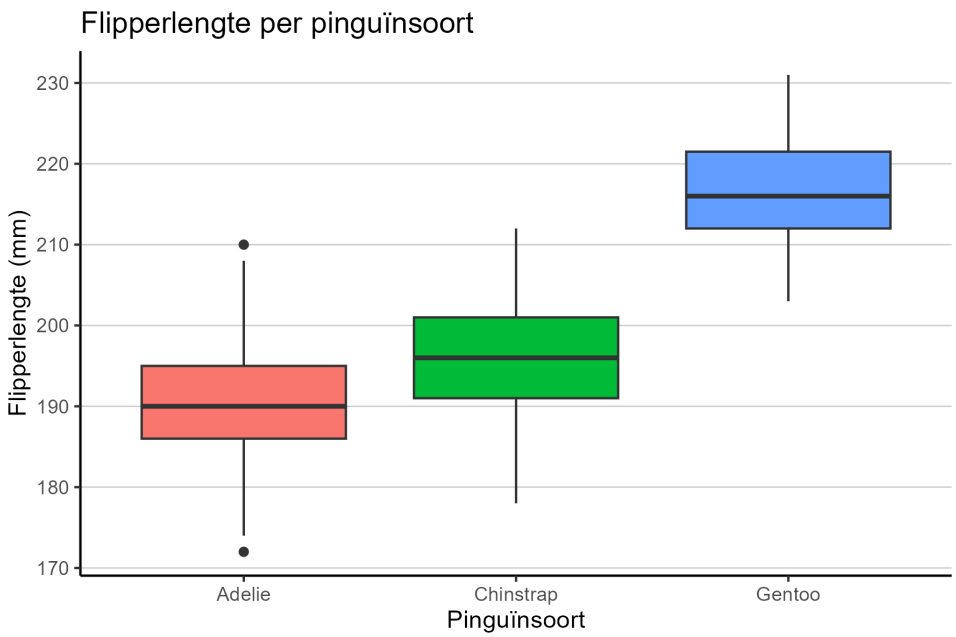
Deze case toont aan dat een eenvoudige barplot al een duidelijke spreiding in interpretatiekwaliteit tussen de tools oplevert. Claude en ChatGPT tonen zich in staat om deze basisgrafiek correct en nuttig te analyseren, terwijl Grok het bij benadering correct doet en DeepSeek fundamenteel tekortschiet. De verschillen in diepgang maken duidelijk dat niet alleen nauwkeurigheid, maar ook contextueel inzicht een rol speelt bij visuele EDA.

4.2.2 Centrum en spreiding interpreteren

In de tweede case werd aan de AI-tools een boxplot voorgelegd waarin de flipperlengte van drie pinguïnsoorten (Adelie, Chinstrap en Gentoo) werd weergegeven (Figuur 2). Elke soort had een eigen box, waarbij de centrale lijn de mediaan voorstelt, de box zelf de interkwartielafstand, en de whiskers en eventuele stippen de spreiding en uitschieters. De opdracht aan de tools was om opvallende verschillen tussen de soorten te beschrijven op vlak van centrale tendens, spreiding en eventuele outliers. De tools kregen hiervoor de volgende prompt mee: *"Deze boxplot toont de flipperlengte van pinguïns per soort. Wat valt je op aan de verschillen in gemiddelde, spreiding of outliers tussen de pinguïnsoorten?"*

Het correct interpreteren van boxplots is essentieel binnen exploratieve data-analyse, omdat deze visualisatie meerdere kenmerken van een verdeling tegelijk toont [11]. Naast verschillen in mediaan kan ook de overlap tussen groepen, de mate van variatie en de aanwezigheid van uitschieters belangrijke inzichten opleveren [11]. In het kader van groepsvergelijking en classificatie zijn zulke inzichten vaak de basis voor verdere statistische analyses.

Een correcte interpretatie van deze grafiek houdt in dat de tool minstens de volgorde van medianen juist benoemt, verschillen in spreiding opmerkt, en eventuele outliers correct identificeert. Misvattingen zoals verwarring tussen mediaan en gemiddelde, of onterechte uitspraken over statistische significantie, worden als foutief beschouwd.



Figuur 2: Boxplot flipperlengte per pinguïnsoort

Claude gaf een bijzonder volledige en correcte interpretatie. Hij benoemde de medianen per soort, beschreef de spreidingsbreedtes in concrete waarden, en gaf aan dat er geen overlap was tussen de box van Gentoo en de andere twee soorten. Opmerkelijk was de toevoeging dat flipperlengte hierdoor een mogelijk onderscheidend kenmerk vormt tussen de soorten — een waardevolle analytische reflectie. ChatGPT gaf eveneens een nauwkeurige beschrijving van de medianen, de spreiding en de outliers. De interpretatie was correct, goed gestructureerd en duidelijk, al bleef de toelichting iets oppervlakkiger dan bij Claude. Grok leverde eveneens een volledig correcte interpretatie: de volgorde van de medianen werd juist benoemd, spreidingsbreedtes werden besproken en outliers correct herkend. DeepSeek gaf een algemene beschrijving van de verdelingen en hun onderlinge verschillen, maar vermeldde geen concrete waarden of details over de outliers. Daardoor bleef de interpretatie eerder beschrijvend dan feitelijk onderbouwd.

Bij vergelijking van de tools behaalden Claude, ChatGPT en Grok de maximale score van 2 op 2, aangezien ze elk een correcte en volledige interpretatie van de grafiek gaven. Claude viel hierbij op door extra analytische meerwaarde te bieden. DeepSeek bleef vaag in zijn beschrijvingen en gaf geen exacte waarden of duidelijke observaties over uitschieters, waardoor de interpretatie slechts gedeeltelijk correct was en een score van 1 op 2 opleverde. Op het vlak van volledigheid en diepgang scoorde Claude het sterkst, gevolgd door ChatGPT en Grok. DeepSeek bleef het meest oppervlakkig en miste concrete onderbouwing van de waarnemingen.

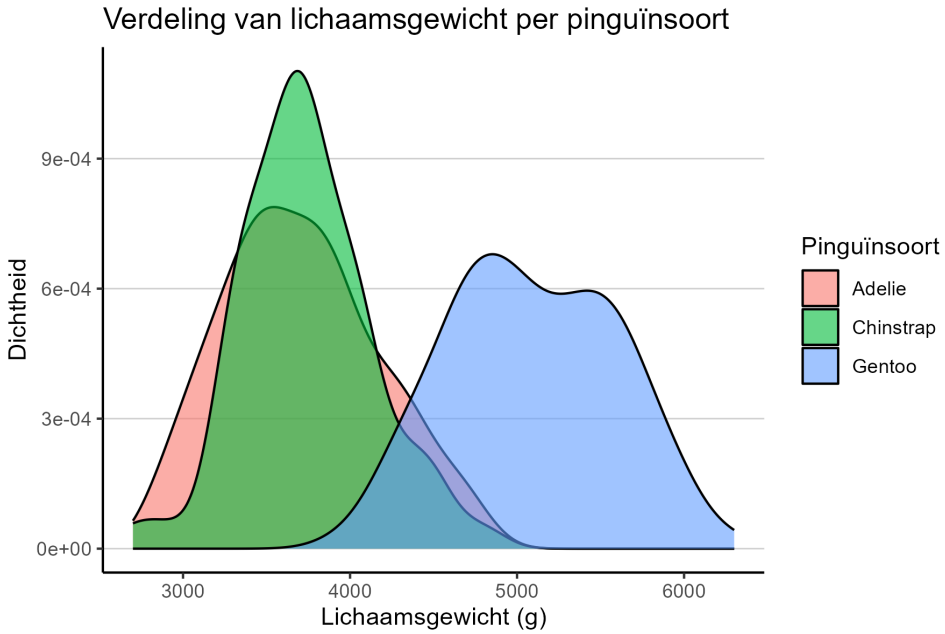
Deze case toont aan dat meerdere AI-tools in staat zijn om boxplots inhoudelijk juist te interpreteren. Toch verschilt de mate waarin zij hun antwoord onderbouwen met feiten en context. Claude blinkt uit in volledigheid en interpretatieve diepgang, terwijl DeepSeek aantoont dat een gebrek aan precisie snel leidt tot een zwakkere beoordeling. Deze oefening benadrukt dat correcte grafiekinterpretatie niet alleen afhangt van herkenning, maar ook van het correct benoemen en toepassen van wat gezien wordt.

4.2.3 Verdeling en scheefheid herkennen

In de derde case kregen de AI-tools een density plot te zien die de verdeling van het lichaamsgewicht van drie pinguïnsoorten (Adelie, Chinstrap en Gentoo) visualiseert (Figuur 3). De verdelingen werden per soort weergegeven in verschillende kleuren, met de x-as als lichaamsgewicht in gram. De tools kregen als opdracht te beschrijven welke verschillen of overlappingen er waren tussen de soorten,

en welke soort gemiddeld het zwaarst lijkt. De bijgevoegde prompt waarmee de AI's aan de slag gingen, was: *"Deze density plot toont de verdeling van het lichaamsgewicht per pinguïnsoort. Zijn er overlappingen of duidelijke verschillen tussen de verdelingen? Welke pinguïnsoort lijkt het zwaarst?"*

Het interpreteren van density plots vereist inzicht in de vorm van verdelingen (zoals piekpositie, breedte, symmetrie), maar ook de vaardigheid om overlappingen of scheidingen tussen groepen correct te benoemen [10]. Deze grafiekvorm speelt een belangrijke rol bij het verkennen van groepsverschillen, vooral wanneer het gaat om continue variabelen [10]. Een correcte lezing in deze case draagt bij aan het begrijpen van biologische verschillen, classificatiepotentieel en mogelijke scheiding tussen groepen.



Figuur 3: Density plot verdeling van lichaamsgewicht per pinguïnsoort

Een correcte interpretatie van deze grafiek houdt in dat de tool herkent dat Gentoo-pinguïns gemiddeld zwaarder zijn dan Adelie en Chinstrap, en dat er een duidelijke overlap is tussen de laatste twee. Daarbij is het belangrijk dat de AI correct weergeeft waar de pieken liggen, of er scheiding is tussen de verdelingen, en welke soort zich onderscheidt. Verkeerd ingeschatte assen of onbestaande variabelen duiden op een misinterpretatie van de visualisatie.

Claude, ChatGPT en Grok gaven een correcte en duidelijke interpretatie van de verdelingen. Alle drie benoemden ze dat Adelie en Chinstrap sterk overlappen, met pieken rond 3500–3800 g, terwijl Gentoo-pinguïns duidelijk zwaarder zijn, met een piek rond 5000 g en weinig overlap met de andere soorten. Claude ging het verst in zijn beschrijving door ook concrete gewichtsintervallen en de mate van overlapping tussen de groepen te benoemen. Hij wees er bovendien op dat Gentoo's verdeling volledig rechts ligt, wat hun onderscheidbaarheid in gewicht bevestigt. DeepSeek daarentegen gaf aan dat hij de grafiek mogelijk niet correct kon interpreteren en baseerde zijn antwoord op vermoedens. In een tweede poging introduceerde de tool zelfs een niet-bestaande temperatuurvariabele, wat duidt op een foutieve interpretatie van de plotstructuur. Daardoor werd de kernvraag van deze case niet correct beantwoord.

Bij vergelijking van de tools behaalden Claude, ChatGPT en Grok de maximale score van 2 op 2. Hun interpretaties waren correct, volledig en consistent met de visuele informatie. Claude onderscheidde zich bovendien door zijn uitgebreide beschrijving van verdelingsvormen en overlapping. DeepSeek slaagde er niet in om een bruikbare interpretatie te geven: de observaties bleven vaag, foutief onderbouwd en uiteindelijk inhoudelijk onjuist en resulteerde in 0 op 2. Op vlak van volledigheid en diepgang stond Claude opnieuw het sterkst, gevolgd door Grok en ChatGPT, die beide

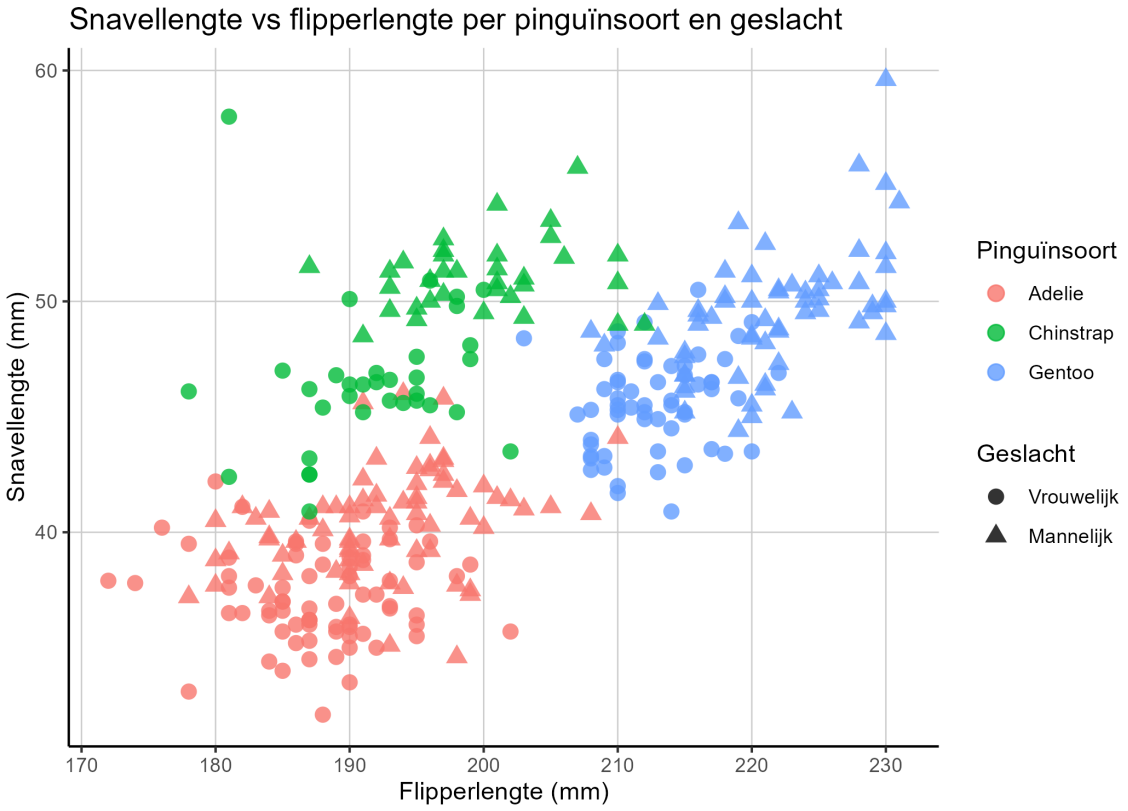
correct werkten maar minder uitgebreid waren. DeepSeek leverde de minst informatieve en de minst betrouwbare interpretatie.

Deze case toont dat het correct lezen van density plots meer vereist dan enkel het herkennen van de "meest rechtse" curve. Tools die erin slagen om de verdelingsvorm te verbinden aan een statistisch of biologisch relevant onderscheid, zoals Claude, leveren een grotere meerwaarde. DeepSeek toont daarentegen dat interpretaties die niet geworteld zijn in concrete gegevens of die verwarring creëren over de grafiekstructuur, weinig bruikbaar zijn in een EDA-context.

4.2.4 Multivariabele patronen analyseren

In case vier kregen de AI-tools een scatterplot te zien waarin twee continue variabelen — snavellengte (y-as) en flipperlengte (x-as) — per pinguïnsoort en per geslacht werden weergegeven (Figuur 4). De kleur gaf de soort aan (Adelie, Chinstrap, Gentoo), terwijl de vorm van de punten (cirkel of driehoek) het geslacht weerspiegelde. De tools kregen de opdracht om opvallende verschillen tussen soorten en geslachten te benoemen, met aandacht voor de verdeling, groottes, clustering en overlapping. De bijhorende prompt bij deze grafiek was: *"Deze scatterplot toont de relatie tussen snavellengte en flipperlengte. De kleur geeft de pinguïnsoort aan, de vorm het geslacht. Wat valt je op aan de verschillen tussen pinguïnsoorten en geslachten?"*

Scatterplots met meerdere visuele coderingen testen of AI in staat is om meerdere variabelen tegelijk correct te interpreteren [11]. Het onderscheiden van trends, groepsclustering, spreiding en morfologische verschillen vraagt om een combinatie van statistisch inzicht en visueel patroonherkenningsvermogen [11]. Bovendien vereist het herkennen van geslachtsverschillen dat de tool begrijpt welke symboolvorm bij welk geslacht hoort, en hoe de posities in de plot die verschillen reflecteren [11].



Figuur 4: Scatterplot snavellengte vs flipperlengte per pinguïnsoort en geslacht

Een correcte interpretatie van deze plot houdt in dat de tool erkent dat Gentoo-pinguïns gemiddeld het langst zijn qua snavel en flipper, dat Adelie het kleinst is, en dat Chinstrap er tussenin zit. Daar-

naast moet een correcte interpretatie vermelden dat er geslachtsverschillen zichtbaar zijn binnen de clusters, met mannetjes (driehoeken) die meestal rechtsboven zitten t.o.v. de vrouwtjes (cirkels). Ook de herkenning van soortclustering en mogelijke correlatie is van belang.

Claude leverde een bijzonder gedetailleerde en correcte analyse. Hij beschreef niet alleen de verschillen in snavel- en flipperlengte per soort, maar benoemde ook duidelijke clustering op soortniveau, de geslachtsverschillen binnen elk cluster, en de positieve correlatie tussen de twee variabelen. Claude wees bovendien op variatie binnen soorten en identificeerde enkele opvallende uitschieters — een interpretatie met veel diepgang. ChatGPT leverde eveneens een correcte en goed gestructureerde beschrijving. De tool onderscheidde de drie soorten correct qua plaatsing en gemiddelde grootte, benoemde subtiele geslachtsverschillen op basis van symboolvorm, en verwees naar clustering binnen elke soort. Grok maakte vergelijkbare observaties en leverde een inhoudelijk correcte interpretatie van de verdeling, met juiste waarden en duidelijke verwijzing naar verschillen tussen soorten en geslachten. Ook DeepSeek leverde een juiste interpretatie van de grafiek: hij benoemde Gentoo als de grootste soort, beschreef correct de relatieve positie van Adelie en Chinstrap, en merkte geslachtsverschillen op. Hoewel de toelichting iets minder uitgebreid was dan bij Claude, was de inhoud volledig correct.

Alle vier de tools behaalden een correcte interpretatie en kregen hiervoor de maximale score van 2 op 2. Claude viel op door zijn breedte en detailniveau, met extra duiding rond correlaties, interne spreiding en outliers. ChatGPT, Grok en DeepSeek leverden eveneens correcte observaties en noemden alle belangrijke elementen, zij het in iets bondigere bewoordingen. Op het vlak van volledigheid en diepgang was Claude het meest uitgebreid, terwijl de andere drie modellen functionele, correcte maar minder diepgaande beschrijvingen gaven.

Deze case toont aan dat alle tools in staat zijn om een scatterplot met meerdere visuele coderingen correct te interpreteren. Het gelijktijdig herkennen van soort-, geslachts- en variatieverschillen is essentieel in multidimensionale exploratieve analyse, en de vier modellen slaagden erin deze informatie correct te koppelen aan de grafiek. Claude onderscheidde zich door verder te gaan in interpretatie en context, maar inhoudelijk waren alle modellen accuraat.

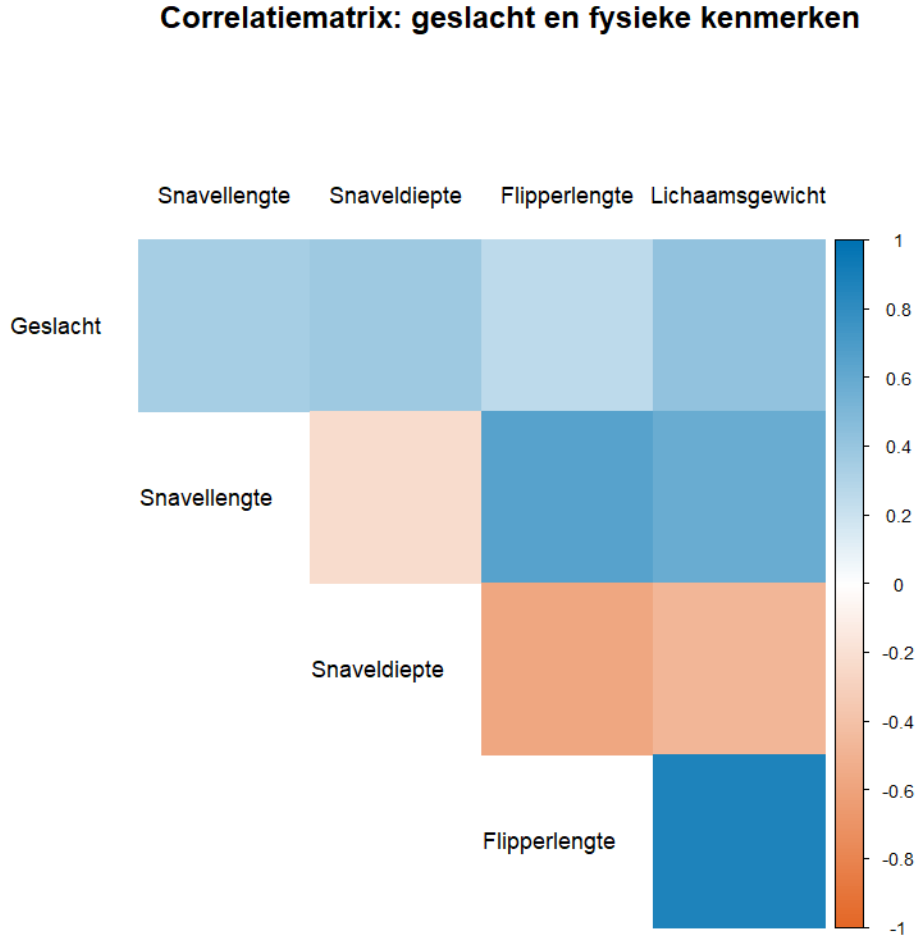
4.2.5 Correlaties duiden

In de laatste case kregen de AI-tools een correlatiematrix te zien die de verbanden toont tussen het geslacht van de pinguïns en vier fysieke kenmerken: snavellengte, snaveldiepte, flipperlengte en lichaamsgewicht (Figuur 5). De matrix was visueel gecodeerd met kleuren die de richting en sterkte van het verband aanduiden, gaande van negatief (bv. oranje of rood) tot positief (blauw), met intensiteit als maat voor correlatiecoëfficiënt. De opdracht aan de tools was om te beschrijven welke variabelen sterk of zwak correleren met geslacht, en welke fysieke kenmerken onderling samenhangen. De tools kregen hiervoor de volgende prompt mee: *"Deze correlatiematrix toont de verbanden tussen geslacht en vier fysieke kenmerken van pinguïns. Welke kenmerken vertonen een sterke of zwakke correlatie met geslacht? Welke variabelen hangen onderling samen?"*

Correlatiematrices zijn een essentieel onderdeel van exploratieve data-analyse, omdat ze in één oogopslag een overzicht geven van alle lineaire verbanden tussen variabelen [10]. Een correcte interpretatie vereist inzicht in de betekenis van positieve en negatieve correlaties, het onderscheiden van sterktegraden, en het kunnen verbinden van deze relaties aan biologische of gedragsmatige kenmerken. In deze case was het bovendien belangrijk dat de tools de context van geslacht correct verwerkten, en niet zomaar elk getal benoemden zonder interpretatie.

Een correcte interpretatie vereist het herkennen van zowel de richting als de orde van grootte van de correlaties. Wat betreft de relatie met geslacht, geldt dat:

- Lichaamsgewicht, snavellengte en snaveldiepte elk een matige positieve correlatie vertonen (0.4)
- Flipperlengte vertoont een zwakke positieve correlatie (0.2)



Figuur 5: Correlatiematrix geslacht en fysieke kenmerken

Wat betreft de onderlinge verbanden tussen fysieke kenmerken zijn de volgende correlaties relevant:

- Flipperlengte en lichaamsgewicht: zeer sterke positieve correlatie (0.9)
- Snavellengte en flipperlengte: sterke positieve correlatie (0.7)
- Snavellengte en lichaamsgewicht: sterke positieve correlatie (0.6)
- Snaveldiepte met lichaamsgewicht: matige negatieve correlatie (−0.4)
- Snaveldiepte met flipperlengte: sterke negatieve correlatie (−0.6)
- Snavellengte en snaveldiepte: zwakke negatieve correlatie (−0.2)

Claude gaf een correcte en volledige interpretatie. Hij benoemde terecht dat geslacht slechts matig positief correleert met de fysieke kenmerken, en gaf correcte inschattingen van de sterkte per variabele. Ook de onderlinge relaties tussen de kenmerken werden volledig en juist besproken, met vermelding van zowel richting als relatieve sterkte. Claude koppelde zijn analyse bovendien aan biologische plausibiliteit, wat de interpretatie versterkte. ChatGPT gaf eveneens een juiste interpretatie van de correlaties met geslacht, en benoemde een aantal belangrijke verbanden tussen fysieke kenmerken. Hij vergat echter enkele onderlinge relaties te vermelden, zoals de correlaties tussen snavellengte met zowel flipperlengte als lichaamsgewicht. De analyse was deels correct, maar minder volledig dan die van Claude.

Grok benoemde de juiste variabelen en vermeldde ook vrijwel alle relevante onderlinge verbanden. Hij gaf concrete inschattingen van correlaties, maar maakte af en toe fouten bij het afleiden van de

sterkte op basis van de kleurschaal. Zo werd bijvoorbeeld een zwakke correlatie omschreven als sterk, en soms als geen correlatie (0). Hoewel de structuur van zijn antwoord degelijk was, leidde deze onnauwkeurigheden tot een gedeeltelijk foutieve interpretatie. DeepSeek kon de grafiek niet analyseren, leverde geen inhoudelijke output, en scoorde dus niet.

In vergelijking met elkaar leverden Claude en ChatGPT een correcte interpretatie, met Claude als de meest volledige met 2 op 2. ChatGPT maakte geen inhoudelijke fouten, maar liet enkele verbanden onvermeld dus verdiende 1 op 2. Grok benoemde weliswaar de juiste relaties, maar inschattingen over sterkte klopten niet altijd, wat resulteert in een gedeeltelijke score van 1 op 2. DeepSeek leverde geen analyse en verkreeg een score van 0 op 2. Wat betreft volledigheid en duiding onderscheidde Claude zich, met een brede en biologisch geïnformeerde benadering.

Deze case toont aan dat het correct interpreteren van correlatiematrices niet alleen draait om herkenning van variabelen, maar ook om het juist inschatten van sterkte, richting en relevantie. Tools zoals Claude tonen aan dat AI in staat is om deze informatie in een breder interpretatiekader te plaatsen. Kleine onnauwkeurigheden in kleur- of schaalinterpretatie, zoals bij ChatGPT en Grok, kunnen echter leiden tot misleidende conclusies.

4.2.6 Tussentijdse conclusie

De vijf visuele interpretatiecases tonen aan dat grote taalmodellen in staat zijn om een breed scala aan grafiektypes correct te interpreteren. Van eenvoudige categorische barplots tot complexe scatterplots en correlatiematrices, de tools toonden in veel gevallen inzicht in de visuele structuur, de gebruikte codering en de betekenis van de weergegeven variabelen. In deze sectie werd opnieuw uitsluitend beoordeeld op nauwkeurigheid, met een score van 0 tot 2 per case. Volledigheid, nuance en interpretatieve diepgang werden kwalitatief besproken.

Claude leverde in alle vijf de cases correcte en inhoudelijk diepgaande interpretaties. De tool viel op door zijn consistente volledigheid, de correcte inschatting van visuele patronen, en de relevante biologische duiding van de waargenomen verschillen. ChatGPT scoorde ook bijna in alle cases correct, met goed gestructureerde en accurate beschrijvingen. Zijn analyses waren doorgaans beknopter dan die van Claude, en miste belangrijke aspecten in case vijf. Grok leverde in de meeste gevallen correcte interpretaties, maar maakte hier en daar fouten in het inschatten van correlatiesterktes of in de nauwkeurigheid van de getallen, wat in twee cases resulteerde in een gedeeltelijke score. DeepSeek scoorde wisselend: bij enkele cases werd een correcte interpretatie gegeven, maar in andere gevallen bleef de output vaag, onvolledig of zelfs afwezig. Hieronder volgt een samenvattende tabel van de nauwkeurigheidsscores van de tools binnen het luik van visuele EDA (Tabel 2).

Case	**Claude**	**ChatGPT**	**Grok**	**DeepSeek**
Case 1	2	2	1	0
Case 2	2	2	2	2
Case 3	2	2	2	0
Case 4	2	2	2	2
Case 5	2	1	1	0
Totaal	**10**	**9**	**8**	**4**

Tabel 2: Overzicht scores visuele EDA

Uit de kwalitatieve bespreking blijkt dat vooral volledigheid, precisie en contextuele duiding de onderscheidende factoren zijn tussen de tools. Claude blinkt hieruit, terwijl ChatGPT eerder kort en krachtig blijft. DeepSeek ondervindt duidelijk beperkingen wanneer het gaat om grafieken waarbij meerdere visuele elementen of kleurgradaties gelijktijdig geïnterpreteerd moeten worden. Deze sectie toont aan dat AI-modellen visuele EDA-taken goed aankunnen, maar dat kleine onnauwkeurigheden — bijvoorbeeld in het interpreteren van kleurgradaties of overlap — nog steeds voorkomen en niet zonder impact zijn op de uiteindelijke conclusie.

4.3 Exploratieve Data Analyse - Numerieke inzichten

Tot slot wordt binnen dit hoofdstuk, naast visuele verkenning, ook stilgestaan bij een derde essentieel onderdeel binnen het data-science proces: het interpreteren van numerieke output. In dit luik wordt onderzocht in welke mate generatieve AI-modellen in staat zijn om numerieke samenvattingen van gegevens correct te interpreteren. Dit vereist niet alleen cijfermatige nauwkeurigheid, maar ook begrip van onderliggende statistische concepten zoals variabiliteit, groepsverschillen, correlatiecoëfficiënten en significantieniveaus. De evaluatie in deze sectie is opgebouwd rond vier soorten numerieke vaardigheden:

1. Het begrijpen van centrale tendens en spreiding;
2. Het interpreteren van meerlagige groepsvergelijkingen;
3. Het lezen en duiden van correlaties;
4. Het beoordelen van statistische significantie.

Net zoals in de vorige hoofdstukken worden de AI-tools per case beoordeeld op nauwkeurigheid (gescoord op 2 punten), en worden daarnaast volledigheid en diepgang kwalitatief besproken. De gebruikte prompts zijn telkens gebaseerd op tabellen of toetsuitvoer in R, en representeren realistische taken uit een data-analyseproces waarin numeriek inzicht vereist is.

De acht geselecteerde cases vormen samen een evenwichtige doorsnede van numerieke interpretatie binnen een biologisch datasetkader, meer bepaald de Palmer Penguins dataset. Ze evalueren niet alleen het rekenkundig vermogen van de modellen, maar ook of ze hun interpretaties correct kunnen kaderen binnen een biologische realiteit — een vaardigheid die essentieel is voor de bruikbaarheid van AI binnen data-science.

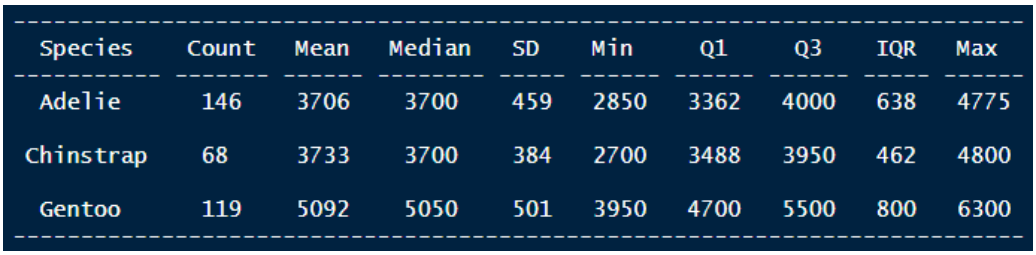
4.3.1 Centrale tendens en spreiding

Een eerste belangrijke vaardigheid in numerieke exploratieve data-analyse is het interpreteren van maatstaven voor centrale tendens en spreiding [12]. Gemiddelde, mediaan, standaarddeviatie (SD) en interkwartielafstand (IQR) zijn klassieke statistieken die gebruikt worden om groepen met elkaar te vergelijken, om verdelingskenmerken zoals symmetrie of scheefheid af te leiden, of om de consistentie binnen een groep in te schatten [12]. In deze subsectie worden twee basisgevallen besproken waarin AI-tools dergelijke beschrijvende statistieken moesten interpreteren. De prompts met bijhorende tabellen zijn opgenomen in Bijlage A.2.1.

Een correcte interpretatie vereist dat de AI correct herkent welke groep (soort of geslacht) het hoogste gemiddelde en/of de hoogste mediaan heeft, en of die twee waarden al dan niet ver uit elkaar liggen. Een klein verschil tussen gemiddelde en mediaan wijst doorgaans op een symmetrische verdeling, terwijl een groter verschil kan wijzen op scheefheid. Daarnaast wordt verwacht dat de tools spreidingsmaten zoals SD en IQR correct gebruiken om uitspraken te doen over variabiliteit of consistentie binnen een groep.

In de eerste case werd de AI gevraagd om de lichaamsmassa van drie pinguïnsoorten te vergelijken op basis van een tabel met beschrijvende statistieken (Figuur 6). Alle vier de tools gaven correct aan dat de Gentoo-pinguïn gemiddeld het zwaarst is, en dat dit overeenkomt met de hoogste mediaan. Claude, Grok en DeepSeek voegden daarbij elk een bredere bespreking toe over het verschil tussen gemiddelde en mediaan, en betrokken ook andere statistieken zoals SD en IQR om uitspraken te doen over de verdelingsvorm. ChatGPT leverde een beknoptere interpretatie, maar wees aan het einde van zijn antwoord wel expliciet op de mogelijkheid om verder te analyseren op basis van SD of IQR — wat functioneel neerkomt op dezelfde reflectie, zij het als optioneel voorstel. De inhoudelijke correctheid was bij alle tools in orde, en de redenering over symmetrie of lichte scheefheid werd telkens juist toegepast.

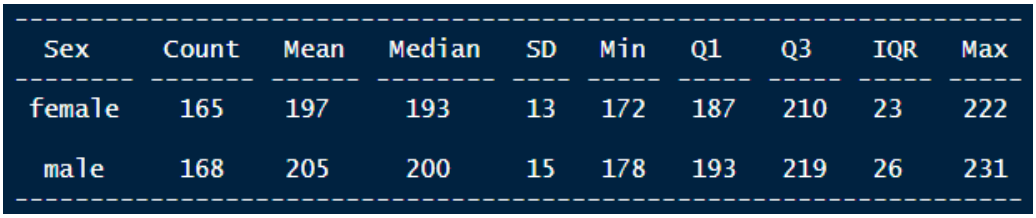
De tweede case ging over flipperlengte bij mannetjes en vrouwtjes (Figuur 7). Ook hier gaven alle tools correct aan dat mannetjes gemiddeld langere flippers hebben dan vrouwtjes, zowel op basis van



Species	Count	Mean	Median	SD	Min	Q1	Q3	IQR	Max
Adelie	146	3706	3700	459	2850	3362	4000	638	4775
Chinstrap	68	3733	3700	384	2700	3488	3950	462	4800
Gentoo	119	5092	5050	501	3950	4700	5500	800	6300

Figuur 6: Beschrijvende statistieken van lichaamsmassa per pinguïnsoort (in g)

het gemiddelde als de mediaan. Daarnaast werd de spreiding bij mannetjes consequent als groter geïnterpreteerd, wat ondersteund werd door hogere SD- en IQR-waarden. Claude en Grok benoemden daarnaast ook het min–maxbereik als aanvullende maatstaf voor variabiliteit. ChatGPT en DeepSeek beperkten zich tot de kernstatistieken, maar maakten verder geen fouten in hun interpretatie. De verdelingen werden in alle gevallen als redelijk symmetrisch omschreven, met iets grotere spreiding bij mannetjes.



Sex	Count	Mean	Median	SD	Min	Q1	Q3	IQR	Max
female	165	197	193	13	172	187	210	23	222
male	168	205	200	15	178	193	219	26	231

Figuur 7: Beschrijvende statistieken van flipperlengte per geslacht (in mm)

Hoewel alle tools in beide cases de maximale score op nauwkeurigheid van 2 op 2 behaalden, kwamen er inhoudelijk toch verschillen naar voren. Claude en Grok leverden beide bijzonder genuanceerde interpretaties, met aandacht voor meerdere statistieken, het relatieve verschil tussen maten, en extra toelichting waar relevant. DeepSeek leverde in case één een uitgebreide analyse met sterke argumentatie, maar bleef in case twee iets meer aan de oppervlakte. ChatGPT was in beide gevallen correct, maar bondiger. In case één gaf hij wel een gerichte follow-upvraag die wees op verder inzicht, maar hij koos er niet voor om die uitwerking zelf al mee te geven.

Deze eerste subsectie toont aan dat AI-tools goed in staat zijn om basale beschrijvende statistiek te interpreteren. Ze herkennen groepsverschillen correct, begrijpen het onderscheid tussen gemiddelde en mediaan, en kunnen spreiding zinvol duiden. De belangrijkste verschillen tussen de tools situeren zich op het vlak van volledigheid, nuance en statistische breedte in het antwoord — met Claude en Grok als duidelijk sterkste voorbeelden binnen deze categorie.

4.3.2 Multivariabele groepsvergelijking

Een verdiepende numerieke interpretatievaardigheid binnen EDA is het kunnen vergelijken van groepen op basis van meer dan één groeperingsvariabele [12]. In deze context worden gegevens meestal gegroepeerd per combinatie van soort en geslacht, wat leidt tot complexere tabellen met meerdere rijen. De AI moet dan niet alleen numerieke waarden correct interpreteren, maar ook logisch redeneren over verschillen binnen en tussen groepen, spreiding inschatten, en soms relatieve metingen uitvoeren. In deze subsectie worden drie cases (case 3-5) besproken die deze vaardigheid op verschillende manieren toetsen. De prompts met bijhorende tabellen zijn opgenomen in Bijlage A.2.2.

Een correcte interpretatie in deze context vereist dat de AI:

- verschillen tussen subgroepen (bv. mannetjes en vrouwtjes binnen een soort) correct kan berekenen, zowel in absolute als relatieve termen;
- verdelingskenmerken (zoals min–maxafstand of Q1/Q3) juist kan ordenen en interpreteren;

- rekening houdt met de juiste referentiegroep voor relatieve uitspraken;
- semantisch correcte en cijfermatig onderbouwde conclusies formuleert.

In case drie kregen de tools een tabel met gemiddelde lichaamsmassa per combinatie van soort en geslacht (Figuur 8). De opdracht was om te bepalen bij welke soort het verschil tussen mannetjes en vrouwtjes het grootst was, zowel in absolute als in relatieve termen. Alle vier de AI-tools beantwoordden beide deelvragen correct. Gentoo werd overal aangeduid als de soort met het grootste absolute verschil (805 g), en Adelie als de soort met het grootste relatieve verschil (20.0%). De tools voerden de berekeningen expliciet uit, gaven inzicht in de formule voor relatieve verschillen en verwezen correct naar de onderliggende waarden. Grok en DeepSeek legden de logica van absolute vs. relatieve verschillen ook duidelijk uit. Claude en ChatGPT waren beknopter, maar stelde hun analyses goed op in twee overzichtelijke stappen.

Species	Sex	Count	Mean	Median	SD	Min	Q1	Q3	IQR	Max
Adelie	female	73	3369	3400	269	2850	3175	3550	375	3900
Adelie	male	73	4044	4000	347	3325	3800	4300	500	4775
Chinstrap	female	34	3527	3550	285	2700	3362	3694	331	4150
Chinstrap	male	34	3939	3950	362	3250	3731	4100	369	4800
Gentoo	female	58	4680	4700	282	3950	4462	4875	412	5200
Gentoo	male	61	5485	5500	313	4750	5300	5700	400	6300

Figuur 8: Beschrijvende statistieken van lichaamsmassa per pinguïnsoort en geslacht (in g)

De vierde case vroeg naar de groep met het grootste verschil tussen minimum en maximum snavellengte, opnieuw zowel in absolute als in relatieve termen (Figuur 9). De informatie was gegroepeerd per soort en geslacht, en de tools moesten dus rijen vergelijken. Alle tools kwamen correct tot het besluit dat Chinstrap–vrouwelijk zowel het grootste absolute verschil (17.1 mm) als het grootste relatieve verschil (41.8%) vertoonde. ChatGPT en Claude gaven de samenvattingen beknopt en duidelijk weer. Grok en DeepSeek beschreven stap voor stap hoe de berekeningen werden uitgevoerd, met extra uitleg bij het belang van het verschil in verhouding tot de minimumwaarde. Opnieuw vielen geen fouten op, en de redeneringen waren correct en helder.

Species	Sex	Count	Mean	Median	SD	Min	Q1	Q3	IQR	Max
Adelie	female	73	37.3	37.0	2.0	32.1	35.9	38.8	2.9	42.2
Adelie	male	73	40.4	40.6	2.3	34.6	39.0	41.5	2.5	46.0
Chinstrap	female	34	46.6	46.3	3.1	40.9	45.4	47.4	2.0	58.0
Chinstrap	male	34	51.1	51.0	1.6	48.5	50.0	52.0	1.9	55.8
Gentoo	female	58	45.6	45.5	2.1	40.9	43.8	46.9	3.0	50.5
Gentoo	male	61	49.5	49.5	2.7	44.4	48.1	50.5	2.4	59.6

Figuur 9: Beschrijvende statistieken van snavellengte per pinguïnsoort en geslacht (in mm)

Case vijf vroeg naar de op één na laagste Q1-waarde en op één na hoogste Q3-waarde binnen een tabel met snaveldiepte per soort en geslacht (Figuur 10). Dit vereiste dat de tools de waarden sorteerden en rangmatig vergeleken. Ook hier kwamen alle vier de tools tot de juiste antwoorden: Gentoo–mannelijk werd correct aangeduid als de groep met de op één na laagste Q1 (15.2 mm), en Adelie–mannelijk als de groep met de op één na hoogste Q3 (19.6 mm). Alle vier de modellen voerden

die sortering correct uit, verwerkten de juiste cijfers en formuleerden foutloze conclusies. Grok gaf daarnaast extra context bij de betekenis van kwartielwaarden in verdelingen, wat de interpretatie verder verrijkte.

Species	Sex	Count	Mean	Median	SD	Min	Q1	Q3	IQR	Max
Adelie	female	73	17.6	17.6	1.0	15.5	17.0	18.3	1.3	20.7
Adelie	male	73	19.1	18.9	1.0	17.0	18.5	19.6	1.1	21.5
Chinstrap	female	34	17.6	17.6	1.0	16.4	17.0	18.0	1.1	19.4
Chinstrap	male	34	19.3	19.3	1.0	17.5	18.8	19.8	1.0	20.8
Gentoo	female	58	14.2	14.2	1.0	13.1	13.8	14.6	0.8	15.5
Gentoo	male	61	15.7	15.7	1.0	14.1	15.2	16.1	0.9	17.3

Figuur 10: Beschrijvende statistieken van snaveldiepte per pinguïnsoort en geslacht (in mm)

De nauwkeurigheid in alle drie de cases was hoog: elk model gaf telkens correcte antwoorden wat resulteerde in scores van 2 op 2. Grok en DeepSeek vielen op door hun stap-voor-stap-benadering, en maakten telkens ruimte voor semantische uitleg over waarom een verschil relevant is — bijvoorbeeld door het verschil te relateren aan spreiding of verhouding. Claude en ChatGPT waren eerder samenvattend in stijl, maar leverden in alle gevallen correcte en duidelijke conclusies.

Deze subsectie toont aan dat AI-tools goed overweg kunnen met multivariabele tabellen en bijhorende groepsvergelijkingen. Zowel absolute als relatieve redeneringen worden vlot uitgevoerd. De interpretatie van gerangschikte waarden (zoals Q1 en Q3) gebeurt systematisch en foutloos. De belangrijkste nuance zit opnieuw in stijl: sommige tools kiezen voor korte overzichtelijke antwoorden, terwijl anderen de lezer actief meenemen in een meer genuanceerd en onderbouwd redeneerproces — wat zeker een meerwaarde kan zijn bij complexere tabellen.

4.3.3 Correlatie-inzicht en associatie

Binnen EDA is het begrijpen van correlaties tussen numerieke variabelen essentieel om structurele verbanden of biologische samenhang tussen kenmerken te detecteren [12]. Correlatie-inzicht vereist dat een AI-toepassing niet enkel de juiste variabele selecteert op basis van een correlatiecoëfficiënt, maar ook dat ze het verschil tussen positieve en negatieve correlaties begrijpt, het onderscheid maakt tussen correlatie en causaliteit, en op een betekenisvolle manier toelichting kan geven bij mogelijke biologische verklaringen. In deze subsectie worden twee cases (case 6 & 7) besproken waarvan de prompts met bijhorende tabellen zijn opgenomen in Bijlage A.2.3.

Een correcte interpretatie binnen dit domein vereist dat de AI:

- de juiste variabele aanduidt bij een opgegeven correlatievraag (bv. hoogste correlatie met X);
- het numerieke verband correct beschrijft en inschat qua sterkte en richting;
- kan verklaren wat de correlatie wel en niet betekent, inclusief foutieve interpretaties vermijden;
- biologische of ecologische duiding geeft die plausibel en data-gedreven is.

In case zes kregen de tools een klassieke correlatievraag: welke variabele vertoont de sterkste positieve correlatie met lichaamsmassa (Figuur 11)? Alle vier de AI-modellen identificeerden correct snaveldiepte als de variabele met de hoogste positieve correlatie (0.65) met lichaamsmassa. Ze interpreteerden de waarde van de correlatie coherent als "matig sterk" tot "sterk" en beschreven de aard van de lineaire relatie tussen snaveldiepte en lichaamsmassa correct. Alle tools gingen in hun biologische uitleg vrij diep in op mogelijke verklaringen, zoals voedselstrategie, morfologische samenhang of seksuele dimorfie. De interpretatie van de correlatiecoëfficiënt verliep bij alle modellen foutloos, wat resulteerde in een maximale score op nauwkeurigheid.

	Lichaamsmassa (g)	Snavellengte (mm)	Snaveldiepte (mm)	Flipperlengte (mm)
Lichaamsmassa (g)	NA	-0.23	0.65	0.59
Snavellengte (mm)	-0.23	NA	-0.58	-0.47
Snaveldiepte (mm)	0.65	-0.58	NA	0.87
Flipperlengte (mm)	0.59	-0.47	0.87	NA

Figuur 11: Correlatiematrix van numerieke variabelen

In case zeven werd een kritische interpretatie gevraagd van een negatieve correlatie tussen snavellengte en snaveldiepte (-0.58) (Figuur 11). Hier moesten de AI-tools aantonen dat ze het verschil tussen samenhang en oorzakelijkheid correct begrijpen, dat ze een negatieve correlatie betekenisvol kunnen uitleggen, en dat ze mogelijke biologische verklaringen kunnen formuleren. Ook in deze case presteerden alle vier de tools sterk. Ze wezen er telkens op dat de negatieve correlatie niet betekent dat langere snavels oorzaak zijn van minder diepe snavels (of omgekeerd), en dat correlatie geen absolute regel is. Alle modellen verwezen daarbij naar verklaringen zoals functionele trade-offs, ecologische aanpassing of morfologische samenhang. Grok en DeepSeek gingen het verst in hun analyse: ze formuleerden bijkomende verklaringen (zoals genetische factoren of seksuele selectie) en gaven kritische kanttekeningen bij de interpretatie van correlaties, bijvoorbeeld door te wijzen op beperkingen van lineaire correlatiecoëfficiënten of mogelijke multicollineariteit tussen variabelen.

De prestaties in deze subsectie waren niet alleen correct, maar vaak ook inhoudelijk genuanceerd, wat leidde tot maximale scores van 2 op 2. In case zes lagen de antwoorden van de vier tools dicht bij elkaar: ze gaven allen een juiste interpretatie van de correlatiecoëfficiënt en een plausibele biologische verklaring. In case zeven kwamen er meer stijl- en diepgangsverschillen naar voren. Grok en DeepSeek vielen op door hun uitgebreide, gestructureerde reflectie over de beperkingen van correlatieanalyse, mogelijke verklaringsmodellen zoals genetica en biomechanica, en kritische kanttekeningen bij de data zelf. Claude en ChatGPT waren correct in hun uitleg, maar beperkten zich iets meer tot de kernpunten zonder uitgebreide methodologische beschouwing.

Deze subsectie toont aan dat AI-tools correlaties niet alleen correct kunnen lezen, maar ook semantisch en biologisch kunnen duiden. De sterkste modellen blinken hier uit in hun vermogen om verbanden te kaderen binnen bredere concepten zoals ecologische adaptatie, allometrie of statistische beperkingen — een vaardigheid die essentieel is bij het verkennen van data in complexe biologische systemen. Alle modellen behaalden in beide cases een correcte interpretatie en dus een maximale score op nauwkeurigheid.

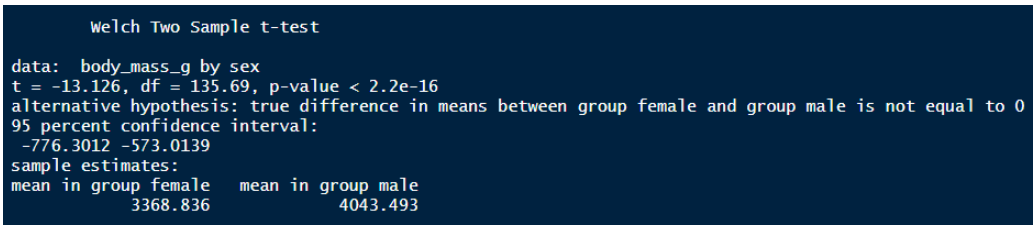
4.3.4 Statistische significantie

Naast het beschrijven van gemiddelden en verbanden tussen variabelen, vereist exploratieve dataanalyse ook inzicht in toetsingsresultaten die aantonen of waargenomen verschillen betekenisvol zijn binnen een statistisch kader [12]. Het correct interpreteren van een t-test is hierbij fundamenteel [12]. Van een AI-tool wordt verwacht dat ze de relevante cijfers accuraat uitleest, de betekenis van de p-waarde begrijpt, het betrouwbaarheidsinterval juist inschat en een semantisch correcte conclusie formuleert. Een correcte interpretatie vereist dat de AI:

- het verschil tussen twee groepen correct beschrijft qua richting en grootte;
- statistische significantie correct afleidt uit p-waarde en betrouwbaarheidsinterval;
- de t-waarde koppelt aan de sterkte van het verschil;
- een praktisch besluit helder en correct formuleert.

In case acht (prompt in Bijlage A.2.4) kregen de tools de resultaten van een t-test op lichaamsmassa bij mannelijke en vrouwelijke Adelie-pinguïns (Figuur 12). Alle vier de AI-modellen interpreteerden het resultaat correct wat een score van 2 op 2 leverde. Ze wezen op het zeer lage significantieniveau ($p < 2.2e\text{-}16$), het 95%-betrouwbaarheidsinterval ([-776.30; -573.01]) dat volledig onder nul lag, en concludeerden dat het verschil in lichaamsmassa tussen de geslachten statistisch

significant was. Elk model gaf aan dat mannetjes gemiddeld zwaarder zijn dan vrouwtjes, met een verschil van ongeveer 675 gram. De interpretatie van de t-waarde (-13.126) en de gevolgtrekking over effectrichting en betrouwbaarheid verliepen bij alle tools foutloos. Ook qua stijl en helderheid van de uitleg was er nauwelijks verschil: alle modellen combineerden statistische juistheid met een duidelijke formulering van de bevindingen.



Figuur 12: T-test lichaamsmassa

Deze case toont aan dat de onderzochte AI-tools betrouwbaar zijn in het interpreteren van toetsingsresultaten. Ze begrijpen de logica van een t-test, maken correct gebruik van alle relevante statistieken, en verwoorden hun conclusies op een duidelijke en technisch correcte manier — wat essentieel is bij het evalueren van groepsverschillen binnen data-analyse.

4.3.5 Tussentijdse conclusie

De resultaten binnen het luik numerieke EDA tonen aan dat de onderzochte AI-tools over een degelijk begrip beschikken van statistische samenvattingen, groepsvergelijkingen en correlatie- en significantieanalyse. In alle acht behandelde cases slaagden de modellen erin om correcte uitspraken te doen over centrale tendens, spreiding, rangordening van groepen, samenhang tussen variabelen en toetsresultaten. Het aantal fouten in de interpretatie was verwaarloosbaar, en op vlak van nauwkeurigheid behaalden de tools vrijwel systematisch de maximale score. Hieronder volgt een samenvattende tabel voor elke tool en elke casegroep (Tabel 3).

Case	Claude	ChatGPT	Grok	DeepSeek
Case 1+2	4	4	4	4
Case 3+4+5	6	6	6	6
Case 6+7	4	4	4	4
Case 8	2	2	2	2
Totaal	**16**	**16**	**16**	**16**

Tabel 3: Overzicht scores numerieke EDA

Wat opvalt, is dat de verschillen tussen de modellen vooral te vinden zijn in de stijl en diepgang van hun antwoorden. Sommige tools, zoals Grok en DeepSeek, kozen regelmatig voor een stapsgewijze uitleg, met bijkomende toelichting over waarom een bepaald verschil of verband betekenisvol is. Zij verwezen spontaan naar concepten zoals relatieve spreiding, statistische beperkingen, of de biologische relevantie van een correlatie. Claude en ChatGPT formuleerden hun conclusies doorgaans compacter, maar zonder aan juistheid in te boeten. Hun stijl was iets meer samenvattend en rechtlijnig, wat in veel gevallen volstaat voor een correcte interpretatie.

In het algemeen bleek dat de AI-modellen sterk presteren op het herkennen van numerieke patronen binnen tabellen en matrices. Ze begrijpen het verschil tussen absolute en relatieve metingen, kunnen kwartielwaarden logisch ordenen, en zijn in staat om correlatie- en t-testresultaten zowel statistisch als semantisch correct te duiden. De analyses waren niet enkel technisch juist, maar ook inhoudelijk plausibel binnen de biologische context van de Palmers Penguins dataset, waarbij ze regelmatig verwezen naar factoren zoals lichaamsgrootte, geslachtsverschillen of ecologische niches.

Deze observaties wijzen erop dat generatieve AI-modellen vandaag al zeer betrouwbare ondersteuning kunnen bieden bij numerieke interpretaties in data-analyse, zeker wanneer het gaat om

basis- en intermediaire statistiek. Ze slagen erin om niet alleen de juiste getallen te vinden, maar ook om daaruit zinvolle conclusies te trekken die relevant zijn binnen het bredere analysekader.

5 Discussie

Dit hoofdstuk bespreekt de belangrijkste bevindingen van het onderzoek, met bijzondere aandacht voor de verschillen in prestaties tussen de onderzochte AI-tools. De evaluatie richtte zich op drie centrale onderdelen binnen de data science workflow: data cleaning, visuele EDA en numerieke EDA. In elk van deze luiken werden meerdere testcases uitgewerkt om de inhoudelijke nauwkeurigheid van de gegenereerde of geïnterpreteerde output te toetsen. De resultaten tonen aan dat alle vier de tools in staat zijn om correcte oplossingen te formuleren, zij het met variatie in bruikbaarheid, volledigheid of diepgang.

De bespreking volgt de driedeling van het onderzoeksontwerp: eerst wordt gereflecteerd per luik op opvallende patronen en verschillen tussen de tools. Vervolgens worden de methodologische beperkingen van het onderzoek besproken, waarna suggesties worden geformuleerd voor toekomstig onderzoek.

5.1 Reflectie per luik

De prestaties op het luik data cleaning lagen over het algemeen hoog: ChatGPT en Claude leverden in alle vier de testcases correcte en functionele oplossingen. Grok en DeepSeek beantwoordden drie van de vier cases accuraat, maar maakten een inhoudelijke fout in case vier (titel-extractie), waarbij de gevraagde teksttransformatie verkeerd werd uitgevoerd. In termen van volledigheid vielen Claude, Grok en DeepSeek op door hun iets uitgebreidere codevoorstellen, vaak aangevuld met extra validatie- of controlemaatregelen. ChatGPT koos doorgaans voor een beknopte, maar correcte aanpak zonder bijkomende toelichting. Op vlak van bruikbaarheid viel op dat Grok in elke case een generiek read_csv-commando gebruikte zonder padverwijzing naar de specifieke testfile, wat de uitvoerbaarheid in een realistische context enigszins beperkte.

Binnen het visuele EDA luik toonden Claude en ChatGPT opnieuw een hoge graad van nauwkeurigheid: zij interpreteerden alle vijf grafieken correct. Grok maakte hier en daar een kleine interpretatiefout, bijvoorbeeld bij het inschatten van relatieve verhoudingen of het benoemen van correlaties. DeepSeek had merkelijk meer moeite met dit luik: slechts één van de vijf grafieken werd correct geïnterpreteerd. In de andere gevallen werden ofwel foutieve observaties gerapporteerd, of werd de grafiek niet geanalyseerd omwille van technische beperkingen in het uitlezen van het beeld. Wat volledigheid betreft, leverde Claude de meest genuanceerde en onderbouwde interpretaties, vaak met biologische of ecologische duiding. ChatGPT formuleerde kernachtige observaties, maar miste bij de correlatiematrix enkele belangrijke verbanden.

Bij de interpretatie van statistische tabellen leverden alle vier de tools over de hele lijn uitstekende prestaties op het vlak van nauwkeurigheid. De cijfers en onderliggende conclusies werden telkens correct geïnterpreteerd. Grok en DeepSeek onderscheidden zich door hun uitgebreidere en vaak methodologisch goed onderbouwde toelichtingen, waarbij ze context boden en mogelijke verklaringen aanhaalden. Claude gaf eveneens gedetailleerde antwoorden, met oog voor nuance en redenering. ChatGPT beperkte zich doorgaans tot bondige en correcte conclusies, maar liet complexere argumentatie vaker achterwege.

De vastgestelde verschillen tussen de tools kunnen deels verklaard worden door bekende architecturale en functionele kenmerken van de onderliggende modellen. Zo is Claude ontworpen met een sterke focus op geavanceerd redeneren, visuele analyse en codegeneratie, wat mogelijk bijdraagt tot de coherente en uitgebreide output in zowel code als visuele EDA [1]. ChatGPT staat bekend om zijn brede toepasbaarheid en efficiënte instructie-afhandeling [25], maar toont ook in andere studies een neiging tot beknopte, taakgerichte antwoorden zonder veel omkadering [5] — wat overeenkomt met de bondige maar correcte output in dit onderzoek. DeepSeek is primair geoptimaliseerd voor

codegeneratie en technische taken, wat zijn degelijke prestaties in cleaning en numerieke EDA ondersteunt [20]. Tegelijk is het een relatief goedkoop model, wat mogelijk bijdraagt tot de beperkte prestaties bij visuele interpretaties — een taak die minder gestructureerde input vereist en meer vraagt van het algemene redeneervermogen van het model [20]. Grok, dat zich profileert als snelle assistent met toegang tot actuele context, bleek inhoudelijk correct bij numerieke en cleaningtaken, maar toonde minder consistentie in visuele analyses, wat mogelijks wijst op beperktere verfijning in patroonherkenning of visuele interpretatie [35]. Deze observaties sluiten aan bij eerdere bevindingen die wijzen op modelafhankelijke sterktes, en onderstrepen het belang van taakgerichte evaluatie bij het beoordelen van AI-ondersteuning in data science.

5.2 Beperkingen

Hoewel dit onderzoek trachtte zo systematisch en reproduceerbaar mogelijk te werk te gaan, zijn er enkele methodologische beperkingen.

Ten eerste werd de evaluatie uitgevoerd door één enkele onderzoeker. Dit verhoogt het risico op beoordelaarsbias, zeker bij de kwalitatieve evaluatiecriteria zoals volledigheid, bruikbaarheid of redeneringsdiepte. Er werd wel gebruikgemaakt van vaste modeloplossingen en scoremodellen om dit risico te beperken, en twijfelgevallen werden herhaaldelijk gecontroleerd. Toch zou intersubjectieve triangulatie via meerdere beoordelaars de betrouwbaarheid verder kunnen verhogen.

Ten tweede is het onderzoek gebaseerd op slechts twee datasets: Titanic voor cleaning en Penguins voor EDA. Hoewel deze datasets representatief zijn en vaak voorkomen in onderwijs en praktijk, blijven het relatief eenvoudige, goed gestructureerde datasets. De generaliseerbaarheid van de bevindingen naar meer complexe of domeinspecifieke datasets (bv. medische of financiële data) blijft hierdoor beperkt.

Ten derde werden de tools geëvalueerd op slechts één specifieke taakvorm per case, telkens op basis van één prompt. Er werd niet geëxperimenteerd met herformuleringen, follow-up vragen of iteratieve interactie. Dit beperkt het inzicht in de robuustheid of interactiekwaliteit van de modellen.

Verder richtte de evaluatie zich uitsluitend op inhoudelijke nauwkeurigheid. Hoewel aspecten zoals gebruiksvriendelijkheid niet formeel werden gescoord, speelden ze onvermijdelijk mee in de kwalitatieve bespreking. In een vroeg stadium werd overwogen om volledigheid en diepgang ook kwantitatief te scoren, maar dit bleek moeilijk objectief uitvoerbaar. Daarom werd gekozen voor een beschrijvende beoordeling op die punten.

Tot slot moet vermeld worden dat het onderzoek enkel generatieve AI-tools evalueerde in het domein van data-analyse in R. Resultaten kunnen verschillen voor andere programmeertalen, datasets of contexten. Daarnaast zijn de prestaties van deze tools onderhevig aan updates, wat de herhaalbaarheid van exacte resultaten in de toekomst kan beïnvloeden.

5.3 Toekomstig Onderzoek

Toekomstig onderzoek kan deze studie op meerdere manieren aanvullen en verdiepen. Een eerste duidelijke aanbeveling betreft het gebruik van meer diverse datasets. In dit onderzoek werd gewerkt met relatief schone en goed gestructureerde benchmarkdatasets (Titanic en Penguins), die typisch zijn voor onderwijssituaties. Om de generaliseerbaarheid van de bevindingen te versterken, zouden vervolgstudies datasets kunnen gebruiken met hogere complexiteit, grotere omvang of domeinspecifieke kenmerken, zoals medische dossiers of financiële transactiegegevens. Zulke data brengen bijkomende uitdagingen met zich mee, waaronder meerlagige variabelen, ruis, onbalans, en contextuele afhankelijkheid — factoren die het cognitieve vermogen van AI-tools verder op de proef stellen.

Een tweede aanbeveling is het inzetten van meerdere beoordelaars om de intersubjectieve betrouwbaarheid van de evaluatie te verhogen. Hoewel dit onderzoek systematisch geëvalueerd werd op basis van modeloplossingen en transparante scoremodellen, blijft er altijd een risico op interpretatieverschillen bij kwalitatieve criteria zoals volledigheid of diepgang.

Verder biedt het domein ruimte om andere AI-tools of architecturen te betrekken. De huidige selectie van vier generatieve LLM's is representatief, maar niet exhaustief. Het zou waardevol zijn om ook gespecialiseerde data science-assistenten of geïntegreerde ontwikkelomgevingen (zoals GitHub Copilot of Amazon CodeWhisperer) in de vergelijking op te nemen. Zulke tools combineren mogelijk generatieve capaciteiten met realtime validatie, syntax-checking of contextuele ondersteuning, wat hun prestaties sterk kan beïnvloeden.

Een bijkomend perspectief is een vergelijking met menselijke baseline-performantie. Door bijvoorbeeld studenten, junior analisten of professionele data scientists dezelfde cases te laten oplossen, kan het relatieve prestatieniveau van AI-tools in kaart worden gebracht. Dit opent de deur naar kost-batenanalyses, pedagogische toepassingen en strategische beslissingen over AI-integratie in data science workflows.

Tot slot zou vervolgonderzoek kunnen inspelen op meer interactieve evaluatievormen. Dit onderzoek werkte bewust met enkelvoudige prompts per case, zonder vervolgvragen of iteraties. In de praktijk verloopt communicatie met AI echter vaak dynamisch, waarbij gebruikers bijsturen, verduidelijking vragen of output aanpassen. Een evaluatie van deze iteratieve interacties zou meer inzicht bieden in de robuustheid, aanpasbaarheid en leereffecten van AI-tools.

6 Conclusie

Deze masterproef onderzocht in welke mate vier generatieve AI-tools — ChatGPT, Claude, DeepSeek en Grok — kunnen functioneren als inhoudelijk correcte assistenten binnen twee cruciale stappen van de data science workflow: data cleaning en EDA. Centraal stond de vraag in hoeverre deze tools accurate en relevante output genereren of interpreteren bij typische voorbereidende analysetaken.

De resultaten tonen aan dat generatieve AI in staat is om voorbereidende analysetaken op een betrouwbare manier uit te voeren, op voorwaarde dat de input duidelijk en gestructureerd wordt aangeleverd. In het luik data cleaning leverden alle tools overwegend correcte oplossingen, met Claude en ChatGPT als de meest consistente. Grok en DeepSeek maakten één fout in de laatste case. Bij visuele EDA viel Claude op door zijn volledigheid en correcte interpretaties; ChatGPT presteerde eveneens sterk, al miste het soms nuance. DeepSeek had het duidelijk moeilijk met visuele input, terwijl Grok wisselende resultaten neerzette. In het luik numerieke EDA waren de prestaties van alle tools zeer sterk, met Grok en DeepSeek die vaak extra toelichting en nuance toevoegden.

Deze bevindingen suggereren dat generatieve AI-tools vandaag al een waardevolle rol kunnen spelen in de voorbereidende fase van data-analyse, zeker wanneer gebruikers heldere prompts formuleren en werken met overzichtelijke input. Claude bleek over de hele lijn de meest consistente en volledige tool, terwijl ChatGPT accuraat maar bondig bleef. Grok en DeepSeek leverden bruikbare output, maar vertoonden grotere variatie afhankelijk van de taak.

Dankzij het gestandaardiseerde en transparante onderzoeksopzet — met vaste prompts, vooraf bepaalde modeloplossingen en systematische scoring per luik — biedt deze studie een robuust en reproduceerbaar kader voor het evalueren van AI in data science-contexten. Hoewel de scope beperkt bleef tot twee datasets en één beoordelaar, levert dit onderzoek een onderbouwd inzicht in de sterktes en beperkingen van actuele AI-tools bij voorbereidende analysetaken.

Generatieve AI is daarmee niet alleen een beloftevolle technologie, maar ook een concreet bruikbare ondersteuning in de dagelijkse praktijk van data-analyse — op voorwaarde dat we haar sterktes benutten, haar beperkingen erkennen, en kritisch blijven in toepassing.

Referenties

Anthropic. (z.d.). Claude. https://www.anthropic.com/claude

Anthropic. (2025). Claude (3.7 Sonnet version) [Large language model]. https://claude.ai/new

Bien, J., & Mukherjee, G. (2024). Generative AI for Data Science 101: Coding Without Learning to Code. *Journal of Statistics and Data Science Education*, *33*(2), 129–142.

Bray, R. (2024). Lessons Learned When Teaching Data Analytics with ChatGPT to MBAs in Spring 2023. *SSRN Electronic Journal*.

Casey, J. C., Dworkin, M., Winschel, J., Molino, J., Daher, M., Katarincic, J. A., Gil, J. A., & Akelman, E. (2024). ChatGPT: A concise Google alternative for people seeking accurate and comprehensive carpal tunnel syndrome information. *Hand Surgery and Rehabilitation*, *43*(5), 101757.

Chu, X., Ilyas, I. F., Krishnan, S., & Wang, J. (2016). Data Cleaning: Overview and Emerging Challenges. *Proceedings of the 2016 International Conference on Management of Data*, 2201–2206.

De Jonge, E., & Van Der Loo, M. (2013). *An introduction to data cleaning with R*. Statistics Netherlands The Hague.

DeepSeek. (2025). DeepSeek (V3 version) [Large language model]. https://chat.deepseek.com/

DeJeu, E. B. (2024). Using Generative AI to Facilitate Data Analysis and Visualization: A Case Study of Olympic Athletes. *Journal of Business and Technical Communication*, *38*(3), 225–241.

Dinov, I. D. (2023). Basic Visualization and Exploratory Data Analytics. In *Data Science and Predictive Analytics: Biomedical and Health Applications using R* (pp. 61–147). Springer.

Ekbote, N., Dhanshetti, P., & Sakhrekar, S. (2023). Techniques of Exploratory Data Analysis. *Madhya Pradesh Journal of Social Sciences*.

Galatro, D., & Dawe, S. (2024). Exploratory Data Analysis. In *Data Analytics for Process Engineers: Prediction, Control and Optimization* (pp. 13–57). Springer.

Hassani, H., & Silva, E. S. (2023). The Role of ChatGPT in Data Science: How AI-Assisted Conversational Interfaces Are Revolutionizing the Field. *Big Data and Cognitive Computing*, *7*(2), 62.

Kalra, V., & Aggarwal, R. (2017). Importance of Text Data Preprocessing & Implementation in RapidMiner. *ICITKM*, *14*, 71–75.

Khatun, A., & Brown, D. G. (2023). Reliability Check: An Analysis of GPT-3's Response to Sensitive Topics and Prompt Wording.

Komorowski, M., Marshall, D. C., Salciccioli, J. D., & Crutain, Y. (2016). Exploratory Data Analysis. In *Secondary Analysis of Electronic Health Records* (pp. 185–203). Springer International Publishing.

Koneva, N., Navarro, A. L. G., Sánchez-Macián, A., Hernández, J. A., Zukerman, M., & González de Dios, Ó. (2025). Introducing Large Language Models as the Next Challenging Internet Traffic Source.

Krishnan, S., Haas, D., Franklin, M. J., & Wu, E. (2016). Towards reliable interactive data cleaning: a user survey and recommendations. *Proceedings of the Workshop on Human-In-the-Loop Data Analytics*, 1–5.

Kumar, P. (2024). Large language models (LLMs): survey, technical frameworks, and future challenges. *Artificial Intelligence Review*, *57*(10), 260.

Lu, M. (2025). The AI revolution triggered by DeepSeek AI: A comprehensive analysis from six perspectives. *Geographical Research Bulletin*, *4*, 120–121.

Lup Low, W., Li Lee, M., & Wang Ling, T. (2001). A knowledge-based approach for duplicate elimination in data cleaning. *Information Systems*, *26*(8), 585–606.

Naeem, Z. A., Ahmad, M. S., Eltabakh, M., Ouzzani, M., & Tang, N. (2024). RetClean: Retrieval-Based Data Cleaning Using LLMs and Data Lakes. *Proceedings of the VLDB Endowment*, *17*(12), 4421–4424.

Oettl, F. C., Oeding, J. F., Feldt, R., Ley, C., Hirschmann, M. T., & Samuelsson, K. (2024). The artificial intelligence advantage: Supercharging exploratory data analysis. *Knee Surgery, Sports Traumatology, Arthroscopy*, *32*(11), 3039–3042.

Oni, S., Chen, Z., Hoban, S., & Jademi, O. (2019). A Comparative Study of Data Cleaning Tools: *International Journal of Data Warehousing and Mining*, *15*(4), 48–65.

OpenAI. (z.d.). ChatGPT. https://openai.com/index/chatgpt/

OpenAI. (2024). ChatGPT (GPT-4o version) [Large language model]. https://chat.openai.com/chat

Pan, W., Liu, D., Meng, J., & Liu, H. (2024). Human–AI communication in initial encounters: How AI agency affects trust, liking, and chat quality evaluation. *New Media & Society*.

Pham, S. T. H., & Sampson, P. M. (2022). The development of artificial intelligence in education: A review in context. *Journal of Computer Assisted Learning*, *38*(5), 1408–1421.

Raj, A., Dubey, A., Rasool, A., & Wadwani, R. (2023). Machine Learning based Missing Data Imputation Techniques. *2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, 1–6.

Rao, A. S., Vardhan, B. V., & Shaik, H. (2021). Role of Exploratory Data Analysis in Data Science. *2021 6th International Conference on Communication and Electronics Systems (ICCES)*, 1457–1461.

Sclar, M., Choi, Y., Tsvetkov, Y., & Suhr, A. (2023). Quantifying Language Models' Sensitivity to Spurious Features in Prompt Design or: How I learned to start worrying about prompt formatting.

Sodano, J. T., & DeFranco, J. F. (2025). Citizen Development, Low-Code/No-Code Platforms, and the Evolution of Generative AI in Software Development. *Computer*, *58*(5), 101–104.

Wang, D., Weisz, J. D., Muller, M., Ram, P., Geyer, W., Dugan, C., Tausczik, Y., Samulowitz, H., & Gray, A. (2019). Human-AI Collaboration in Data Science: Exploring Data Scientists' Perceptions of Automated AI. *Proceedings of the ACM on Human-Computer Interaction*, *3*(CSCW), 1–24.

Wills, S., Poon, S. T. S., Salili-James, A., & Scott, B. (2024). The use of generative AI for coding in academia. *Methods in Ecology and Evolution*, *15*(12), 2189–2191.

xAI. (z.d.). Grok. https://x.ai/grok

xAI. (2025). Grok (3 version) [Large language model]. https://grok.com/chat

Zhecheva, D. (2024). Exploratory Data Analysis and the Rise of Large Language Models - Gaming Industry Insights. *TEM Journal*, 561–569.

A Bijlage – Output AI-tools per luik

A.1 Bijlage Data Cleaning

A.1.1 Case 1 – Verwijderen van duplicaten

```r
# Lees de csv-gegevens in
data <- read.csv("case1_small_raw.csv")

# Controleer oorspronkelijke aantal rijen
original_rows <- nrow(data)
print(paste("Oorspronkelijk aantal rijen:", original_rows))

# Verwijder duplicaatrijen (rijen die volledig identiek zijn op basis van alle kolommen)
deduplicated_data <- unique(data)

# Controleer aantal rijen na verwijderen duplicaten
new_rows <- nrow(deduplicated_data)
print(paste("Aantal rijen na verwijderen van duplicaten:", new_rows))
print(paste("Aantal verwijderde rijen:", original_rows - new_rows))

# Sla de opgeschoonde gegevens op
write.csv(deduplicated_data, "case1_small_cleaned.csv", row.names = FALSE)
```

Code 1: Claude – R-code voor het verwijderen van duplicaten

```r
# Lees het CSV-bestand in
data <- read.csv("case1_small_raw.csv", stringsAsFactors = FALSE)

# Verwijder volledig identieke rijen
data_uniek <- unique(data)

# Optioneel: sla het resultaat op
write.csv(data_uniek, "case1_small_raw_uniek.csv", row.names = FALSE)

# Bekijk resultaat
print(data_uniek)
```

Code 2: ChatGPT – R-code voor het verwijderen van duplicaten

```r
# Lees de dataset in
data <- read.csv("case1_small_raw.csv")

# Verwijder volledig identieke rijen (op basis van alle kolommen)
unique_data <- unique(data)

# Controleer het resultaat
cat("Aantal rijen originele dataset:", nrow(data), "\n")
cat("Aantal rijen na verwijderen duplicaten:", nrow(unique_data), "\n")
cat("Aantal verwijderde duplicaten:", nrow(data) - nrow(unique_data), "\n")

# Optioneel: schrijf het resultaat naar een nieuw CSV-bestand
# write.csv(unique_data, "case1_small_unique.csv", row.names = FALSE)
```

Code 3: DeepSeek – R-code voor het verwijderen van duplicaten

```r
# Laad de dataset (vervang 'titanic.csv' door de juiste bestandsnaam of gebruik een dataframe)
data <- read.csv("titanic.csv", na.strings = "NA", stringsAsFactors = FALSE)

# Verwijder duplicaten op basis van alle kolommen
unique_data <- data[!duplicated(data), ]
```

```r
# Schrijf het resultaat naar een nieuw bestand (optioneel)
write.csv(unique_data, "titanic_unique.csv", row.names = FALSE)

# (Optioneel) Bekijk het aantal rijen voor en na
cat("Oorspronkelijk aantal rijen:", nrow(data), "\n")
cat("Aantal unieke rijen:", nrow(unique_data), "\n")
```

Code 4: Grok – R-code voor het verwijderen van duplicaten

```r
# Lees data en verwijder duplicaten
unique_data <- read.csv("case1_small_raw.csv") %>%
  distinct()

# Optioneel: schrijf het resultaat naar een nieuw CSV-bestand
# write.csv(unique_data, "case1_small_unique.csv", row.names = FALSE)
```

Code 5: DeepSeek – Alternatieve R-code voor het verwijderen van duplicaten

Grok – Beschrijvende uitleg bij gegenereerde code 4

Uitleg:

- **Dataset inladen**: De functie `read.csv` laadt de dataset in. De parameter `na.strings = "NA"` zorgt ervoor dat ”NA” als missing value wordt herkend.
- **Duplicaten verwijderen**: De functie `duplicated()` identificeert rijen die identiek zijn aan een eerdere rij op basis van alle kolommen. Met `!duplicated(data)` selecteren we alleen de eerste voorkomens van unieke rijen.
- **Resultaat opslaan**: Het resultaat wordt opgeslagen in `unique_data`. Optioneel kan dit weggeschreven worden met `write.csv()`.
- **Controle**: `cat()`-statements tonen het aantal rijen vóór en na het verwijderen van duplicaten.

Deze code zorgt ervoor dat alleen unieke rijen overblijven, waarbij duplicaten (rijen die exact gelijk zijn in alle kolommen) worden verwijderd.

DeepSeek – Beschrijvende uitleg bij gegenereerde code 3 en 5

Uitleg:

- `read.csv()` – Leest het CSV-bestand in.
- `unique()` – Behoudt alleen unieke rijen op basis van alle kolommen.
- `cat()` – Geeft een overzicht van hoeveel rijen er zijn verwijderd.

A.1.2 Case 2 – Aanvullen van ontbrekende waarden

```r
# Lees de CSV-bestand in
titanic_data <- read.csv("case2_small_raw.csv")

# Controleer ontbrekende waarden in Age en Embarked
cat("Aantal ontbrekende waarden in Age:", sum(is.na(titanic_data$Age)), "\n")
cat("Aantal ontbrekende waarden in Embarked:", sum(is.na(titanic_data$Embarked)), "\n")
```

```r
# Bereken de mediaan van Age (exclusief NA waarden)
age_median <- median(titanic_data$Age, na.rm = TRUE)

# Vind de meest voorkomende waarde (modus) voor Embarked
embarked_table <- table(titanic_data$Embarked)
embarked_mode <- names(embarked_table)[which.max(embarked_table)]

# Vul de ontbrekende waarden aan
titanic_data$Age[is.na(titanic_data$Age)] <- age_median
titanic_data$Embarked[is.na(titanic_data$Embarked)] <- embarked_mode

# Controleer of de ontbrekende waarden zijn aangevuld
cat("Aantal ontbrekende waarden in Age na aanvulling:", sum(is.na(titanic_data$Age)), "\n")
cat("Aantal ontbrekende waarden in Embarked na aanvulling:", sum(is.na(titanic_data$Embarked))
    , "\n")

# Schrijf het resultaat naar een nieuw CSV-bestand
write.csv(titanic_data, "titanic_data_imputed.csv", row.names = FALSE)

# Toon een samenvatting van de data
summary(titanic_data[, c("Age", "Embarked")])
```

Code 6: Claude – R-code voor het aanvullen van ontbrekende waarden

```r
# Laad de data
df <- read.csv("case2_small_raw.csv", stringsAsFactors = FALSE)

# Vervang NA in 'Age' door de mediaan van 'Age' (zonder NA)
df$Age[is.na(df$Age)] <- median(df$Age, na.rm = TRUE)

# Functie om modus te berekenen
get_mode <- function(x) {
  uniqx <- unique(x[!is.na(x)])
  uniqx[which.max(tabulate(match(x, uniqx)))]
}

# Vervang NA in 'Embarked' door de modus
df$Embarked[is.na(df$Embarked)] <- get_mode(df$Embarked)

# Bekijk een samenvatting om te controleren
summary(df)
```

Code 7: ChatGPT – R-code voor het aanvullen van ontbrekende waarden

```r
# Lees de data in
data <- read.csv("case2_small_raw.csv")

# Vervang NA in 'Age' door de mediaan
data$Age[is.na(data$Age)] <- median(data$Age, na.rm = TRUE)

# Vervang NA in 'Embarked' door de modus (meest voorkomende waarde)
embarked_mode <- names(sort(table(data$Embarked), decreasing = TRUE))[1]
data$Embarked[is.na(data$Embarked)] <- embarked_mode

# Toon de eerste paar rijen om het resultaat te controleren
head(data)
```

Code 8: DeepSeek – R-code voor het aanvullen van ontbrekende waarden

```r
# Laad de data (vervang 'titanic.csv' door de juiste bestandsnaam indien nodig)
data <- read.csv("titanic.csv", na.strings = c("NA", ""))
```

```r
# Functie om de modus te berekenen voor 'Embarked'
get_mode <- function(x) {
  ux <- unique(x[!is.na(x)])
  ux[which.max(tabulate(match(x, ux)))]
}

# Vul ontbrekende waarden in 'Age' met de mediaan
data$Age[is.na(data$Age)] <- median(data$Age, na.rm = TRUE)

# Vul ontbrekende waarden in 'Embarked' met de modus
data$Embarked[is.na(data$Embarked)] <- get_mode(data$Embarked)

# Controleer het resultaat
summary(data[, c("Age", "Embarked")])
```

Code 9: Grok – R-code voor het aanvullen van ontbrekende waarden

Claude – Beschrijvende uitleg bij gegenereerde code 6

Uitleg:

- Leest het CSV-bestand in.
- Controleert hoeveel ontbrekende waarden er zijn in de kolommen `Age` en `Embarked`.
- Berekent de mediaan van `Age` (zonder de `NA`-waarden).
- Bepaalt de meest voorkomende waarde (modus) voor `Embarked`.
- Vult de ontbrekende waarden in beide kolommen aan.
- Controleert of de aanvulling succesvol was.
- Schrijft de aangepaste data naar een nieuw CSV-bestand.
- Toont een samenvatting van de aangepaste kolommen.

De andere kolommen blijven ongewijzigd, zoals gevraagd.

Grok – Beschrijvende uitleg bij gegenereerde code 9

Uitleg:

- De mediaan voor `Age` wordt berekend met `median(data$Age, na.rm = TRUE)` en gebruikt om `NA`-waarden te vervangen.
- Voor `Embarked` wordt een eenvoudige `get_mode`-functie gedefinieerd om de meest voorkomende waarde te vinden, die vervolgens wordt gebruikt om `NA`-waarden te vervangen.
- Andere kolommen blijven ongewijzigd.
- De `summary()`-functie toont een overzicht om te verifiëren dat er geen `NA`-waarden meer zijn in `Age` en `Embarked`.

Deze code zorgt ervoor dat alleen unieke rijen overblijven, waarbij duplicaten (rijen die exact gelijk zijn in alle kolommen) worden verwijderd.

DeepSeek – Beschrijvende uitleg bij gegenereerde code 8

Uitleg:

- **Lezen van de data:** De dataset wordt ingelezen vanuit het CSV-bestand.
- **Aanvullen van `Age`:**
  - `median(data$Age, na.rm = TRUE)` berekent de mediaan van de kolom `Age`, waarbij `na.rm = TRUE` ontbrekende waarden negeert.
  - `is.na(data$Age)` identificeert de ontbrekende waarden in `Age`.
- **Aanvullen van `Embarked`:**
  - `table(data$Embarked)` telt de frequentie van elke waarde.
  - `sort(..., decreasing = TRUE)` sorteert deze frequenties aflopend.
  - `names(...)[1]` selecteert de waarde met de hoogste frequentie (modus).
  - De ontbrekende waarden worden hiermee vervangen.
- **Controle:** `head(data)` toont de eerste rijen van de aangepaste dataset ter verificatie.

Deze code zorgt ervoor dat alleen de kolommen 'Age' en 'Embarked' worden aangepast, terwijl andere kolommen ongewijzigd blijven.

A.1.3 Case 3 – Foutieve invoer detecteren en verwijderen

```r
# Data inladen
data <- read.csv("case3_small_raw.csv")

# Originele aantal rijen bewaren om verwijderingen te kunnen rapporteren
original_rows <- nrow(data)

# Age converteren van character naar numeric
# Eerst vervangen we duidelijk ongeldige waarden door NA
data$Age <- gsub("--|\\?|missing|unknown|NaN|null|###|/", NA, data$Age)

# Converteren naar numeric - R zet automatisch niet-numerieke waarden om naar NA
data$Age <- as.numeric(data$Age)

# Rijen met NA in Age verwijderen
data <- data[!is.na(data$Age), ]

# Afronden naar gehele getallen
data$Age <- round(data$Age)

# Rapporteren hoeveel rijen zijn verwijderd
removed_rows <- original_rows - nrow(data)
cat("Aantal verwijderde rijen:", removed_rows, "\n")
cat("Resterend aantal rijen:", nrow(data), "\n")

# Eerste paar rijen tonen om te controleren
head(data)

# Dataframe opslaan als nieuwe csv
write.csv(data, "case3_cleaned.csv", row.names = FALSE)
```

Code 10: Claude – R-code voor het detecteren en verwijderen van foutieve invoer

```r
# Laad het bestand
data <- read.csv("case3_small_raw.csv", stringsAsFactors = FALSE)

# Bekijk unieke waarden in Age (optioneel)
unique(data$Age)

# Zet Age om naar numeriek en verwijder rijen met NA's
data$Age <- as.integer(data$Age)
data <- data[!is.na(data$Age), ]

# Controleer resultaat (optioneel)
str(data)
summary(data$Age)
```

Code 11: ChatGPT – R-code voor het detecteren en verwijderen van foutieve invoer

```r
# Laad de benodigde packages
library(dplyr)
library(readr)

# Lees de dataset in
data <- read_csv("case3_small_raw.csv")

# Definieer een functie om leeftijd te converteren en ongeldige waarden te identificeren
convert_age <- function(age_str) {
  # Vervang speciale tekens zoals "--", "?", "/", "missing", "unknown" en "###" met NA
  age_str <- ifelse(age_str %in% c("--", "?", "/", "missing", "unknown", "###", "NaN"), NA,
      age_str)

  # Converteer naar numeriek
  age_num <- suppressWarnings(as.numeric(age_str))

  # Rond af naar geheel getal
  age_int <- as.integer(round(age_num))

  return(age_int)
}

# Pas de conversie toe op de Age kolom
data_clean <- data %>%
  mutate(
    Age = convert_age(Age)
  ) %>%
  filter(
    !is.na(Age) & Age >= 0  # Verwijder rijen met NA of negatieve leeftijden
  )

# Toon de samenvatting van de schone data
summary(data_clean$Age)

# Optioneel: bekijk de eerste rijen om het resultaat te controleren
head(data_clean)
```

Code 12: DeepSeek – R-code voor het detecteren en verwijderen van foutieve invoer

```r
# Laad de data (vervang 'titanic.csv' door je bestandspad of gebruik de data direct)
data <- read.csv("titanic.csv", stringsAsFactors = FALSE)

# Zet 'Age' om naar numeriek, ongeldige waarden worden NA
data$Age <- as.numeric(data$Age)

# Verwijder rijen waar Age NA is
```

```r
data <- data[!is.na(data$Age), ]

# Zet Age om naar geheel getal
data$Age <- as.integer(data$Age)

# Bekijk het resultaat
head(data)
```

Code 13: Grok – R-code voor het detecteren en verwijderen van foutieve invoer

Claude – Analyse van onzuivere waarden (boven de code 10)

Ik ga R-code schrijven om de `Age`-kolom van tekst naar gehele getallen om te zetten en ongeldige waarden te verwijderen. Eerst analyseer ik de data om te zien welke soorten ongeldige waarden er zijn. In de dataset zie ik dat `Age` verschillende formaten heeft, waaronder:

- Gewone getallen: `22`, `55`.
- Streepjes: `--`.
- Vraagtekens: `?`.
- Termen als `missing` en `unknown`.
- Ontbrekende waarden zoals `NaN`, `null`, `###`.
- Slash: `/`.

Claude – Beschrijvende uitleg van het stappenplan (onder de code 10)

Deze code:

- Leest de dataset in.
- Houdt bij hoeveel rijen er oorspronkelijk zijn.
- Vervangt alle ongewenste waarden door `NA`.
- Zet de kolom om naar numeriek formaat (`NA` voor foutieve waarden).
- Verwijdert rijen waar `Age` `NA` is geworden.
- Rondt de leeftijden af naar gehele getallen.
- Rapporteert hoeveel rijen verwijderd zijn.
- Toont de eerste rijen ter controle.
- Slaat de opgeschoonde data op.

Deze aanpak zorgt ervoor dat alleen rijen met geldige leeftijdswaarden behouden blijven en dat alle leeftijden worden opgeslagen als gehele getallen.

DeepSeek – Beschrijvende uitleg bij gegenereerde code 12

Inlezen data:
De dataset wordt ingelezen met `read_csv()` uit het `readr` package, wat beter omgaat met verschillende dataformaten dan de standaard R-functies.

Conversiefunctie:

- Speciale tekens die vaak gebruikt worden voor ontbrekende waarden (zoals `"--"`, `"?"`, `"/"`, etc.) worden eerst omgezet naar `NA`.
- Daarna wordt de tekst omgezet naar een numerieke waarde met `as.numeric()`.
- De numerieke waarde wordt afgerond naar een geheel getal met `as.integer(round())`.

Filteren:

- Rijen waar `Age` `NA` is (door ongeldige invoer) worden verwijderd.
- Negatieve leeftijden (zoals `"-"` in de originele data) worden ook verwijderd.

Resultaat:
`summary(data_clean$Age)` geeft een overzicht van de leeftijden in de opgeschoonde dataset. `head(data_clean)` laat de eerste rijen zien om te controleren of de conversie correct is uitgevoerd.

A.1.4 Case 4 – Informatie extraheren uit tekstvelden

```r
# Lees eerst de dataset in (in een echte situatie zou je het bestand inlezen)
# In dit geval gebruiken we de data die al in het geheugen staat
# titanic_data <- read.csv("case4_small_raw.csv")

# Extractie van titels uit de "Name" kolom
# We gebruiken reguliere expressies om alles tussen de komma en een punt te vinden
titanic_data$Title <- gsub("^.*, (.*?)\\. .*$", "\\1", titanic_data$Name)

# Toon de eerste paar rijen om te controleren of de extractie correct werkt
head(titanic_data[, c("Name", "Title")])
```

Code 14: Claude – R-code voor het extraheren van informatie uit tekstvelden

```r
# Laad de benodigde bibliotheek
library(dplyr)

# Laad de data
df <- read.csv("case4_small_raw.csv")

# Extraheer de titel met een regex en voeg toe als nieuwe kolom
df <- df %>%
  mutate(Title = sub(".*, (.*?)\\..*", "\\1", Name))

# Bekijk de eerste paar rijen om het resultaat te controleren
head(df[, c("Name", "Title")])
```

Code 15: ChatGPT – R-code voor het extraheren van informatie uit tekstvelden

```r
# Lees de dataset in
data <- read.csv("case4_small_raw.csv")

# Extraheer de titel uit de 'Name'-kolom met behulp van reguliere expressies
data$Title <- gsub("^.*,\\s(\\w+).*$", "\\1", data$Name)
```

```r
# Toon de eerste paar rijen om het resultaat te controleren
head(data[c("Name", "Title")])

# Optioneel: Sla de dataset op met de nieuwe kolom
# write.csv(data, "case4_small_raw_with_title.csv", row.names = FALSE)
head(data_clean)
```

Code 16: DeepSeek – R-code voor het extraheren van informatie uit tekstvelden

```r
# Laad de data (ervan uitgaande dat de data in een dataframe 'df' zit)
# Als je de data uit een CSV-bestand leest, gebruik dan bijvoorbeeld:
# df <- read.csv("titanic.csv", na.strings = c("NA", ""))

# Extraheer de titel uit de Name kolom en voeg toe als nieuwe kolom
df$Title <- sub(".*?, (\\w+\\.?\\w*).*$", "\\1", df$Name)

# Bekijk het resultaat (optioneel)
head(df[, c("Name", "Title")])
```

Code 17: Grok – R-code voor het extraheren van informatie uit tekstvelden

ChatGPT – Beschrijvende uitleg bij gegenereerde code 15

Deze code gebruikt `sub()` om alles tussen de komma en de punt in de naam te extraheren, wat overeenkomt met de titel.

Claude – Beschrijvende uitleg bij gegenereerde code 14

Dit script doet het volgende:

- Het gebruikt de `gsub()` functie met een reguliere expressie om de titel te extraheren.
- De reguliere expressie `^.*, (.*?)\\. .*$` betekent:
  - `^.*,` – Match alles vanaf het begin tot een komma.
  - `(.*?)` – Vang alles op (niet-greedy) in een groep, dit is de titel.
  - `\\.` – Match een punt.
  - `.*$` – Match alles tot het einde van de string.
- `\\1` in het vervangingspatroon refereert naar de eerste vanggroep, wat de titel is die geëxtraheerd wordt.

Als er speciale gevallen zijn in je dataset (bijvoorbeeld namen met meerdere titels of ongebruikelijke formats), dan kan de reguliere expressie verder verfijnd worden.

Grok – Beschrijvende uitleg bij gegenereerde code 17

Regex-patroon:
`sub(".*?, (\\w+\\.?\\w*).*$", "\\1", df$Name)`

- `.*?` zoekt naar alles vóór de komma (achternaam).
- `(\\w+\\.?\\w*)` vangt de titel:
  - `\\w+` matcht een of meer letters (bijv. `"Mr"`, `"Miss"`).
  - `\\.?` matcht een optionele punt.
  - `\\w*` matcht optionele extra letters (bijv. voor `"Countess"`).
- `.*$` matcht de rest van de string na de titel.
- `\\1` retourneert de gevangen titel.

Resultaat:
De nieuwe kolom `Title` bevat titels zoals `"Mr"`, `"Miss"`, `"Mrs"`, `"Master"`, enz. De originele `Name`-kolom blijft ongewijzigd. De `head()`-functie toont een preview van de kolommen `Name` en `Title` ter controle.

DeepSeek – Beschrijvende uitleg bij gegenereerde code 16

Reguliere expressie:

- `^.*,\\s(\\w+).*$` zoekt naar tekst die begint met een naam gevolgd door een komma en een spatie, dan een woord (de titel), en de rest van de tekst.
- `\\1` behoudt de eerste groep (de titel) uit de reguliere expressie.

Resultaat:

- De nieuwe kolom `Title` wordt toegevoegd aan de dataset met de geëxtraheerde titels (bijv. `Mr`, `Miss`, `Mrs`, enz.).

Controle:

- `head(data[c("Name", "Title")])` toont de eerste paar rijen van de `Name` en `Title` kolommen om te verifiëren of de extractie correct is uitgevoerd.

Opslaan:

- De laatste regel (gecommenteerd) kan worden gebruikt om de aangepaste dataset op te slaan naar een nieuw CSV-bestand.

A.2 Bijlage Numerieke EDA

A.2.1 Centrale tendens en spreiding

Prompt 1: *Bekijk de onderstaande tabel met beschrijvende statistieken van de lichaamsmassa per pinguïnsoort. Welke soort heeft gemiddeld de hoogste lichaamsmassa volgens de cijfers? Komt dit overeen met de mediaan? Wat zegt dat over de verdeling van de datapunten? Leg je antwoord uit en verwijs daarbij expliciet naar de waarden in de tabel.*

Species	Count	Mean	Median	SD	Min	Q1	Q3	IQR	Max
Adelie	146	3706	3700	459	2850	3362	4000	638	4775
Chinstrap	68	3733	3700	384	2700	3488	3950	462	4800
Gentoo	119	5092	5050	501	3950	4700	5500	800	6300

Figuur 13: Beschrijvende statistieken van lichaamsmassa per pinguïnsoort (in g)

Prompt 2: *Bekijk de onderstaande tabel met beschrijvende statistieken van de flipperlengte per geslacht. Welk geslacht heeft gemiddeld langere flippers volgens de gegevens? Gebruik daarnaast de standaarddeviatie en interkwartielafstand (IQR) om te beoordelen hoe groot de spreiding is binnen elk geslacht. Leg je antwoord uit en verwijs daarbij expliciet naar de waarden in de tabel.*

Sex	Count	Mean	Median	SD	Min	Q1	Q3	IQR	Max
female	165	197	193	13	172	187	210	23	222
male	168	205	200	15	178	193	219	26	231

Figuur 14: Beschrijvende statistieken van flipperlengte per geslacht (in mm)

A.2.2 Multivariabele groepsvergelijking

Prompt 3: *Bekijk de onderstaande tabel met beschrijvende statistieken van de lichaamsmassa per combinatie van pinguïnsoort en geslacht. In welke soort is het verschil in gemiddelde lichaamsmassa tussen mannetjes en vrouwtjes het grootst in absolute getallen? In welke soort is het verschil in gemiddelde lichaamsmassa tussen mannetjes en vrouwtjes het grootst in relatieve getallen? Leg je antwoord uit en verwijs daarbij expliciet naar de waarden in de tabel.*

Species	Sex	Count	Mean	Median	SD	Min	Q1	Q3	IQR	Max
Adelie	female	73	3369	3400	269	2850	3175	3550	375	3900
Adelie	male	73	4044	4000	347	3325	3800	4300	500	4775
Chinstrap	female	34	3527	3550	285	2700	3362	3694	331	4150
Chinstrap	male	34	3939	3950	362	3250	3731	4100	369	4800
Gentoo	female	58	4680	4700	282	3950	4462	4875	412	5200
Gentoo	male	61	5485	5500	313	4750	5300	5700	400	6300

Figuur 15: Beschrijvende statistieken van lichaamsmassa per pinguïnsoort en geslacht (in g)

Prompt 4: *Bekijk de onderstaande tabel met beschrijvende statistieken van de snavellengte per pinguïnsoort en geslacht. In welke combinatie van soort en geslacht ligt de minimumwaarde van de snavellengte het verste van de maximumwaarde van de snavellengte in absolute getallen? In welke combinatie van soort en geslacht ligt de minimumwaarde van de snavellengte het verste van de maximumwaarde van de snavellengte in relatieve getallen? Leg je antwoord uit en verwijs daarbij expliciet naar de waarden in de tabel.*

Species	Sex	Count	Mean	Median	SD	Min	Q1	Q3	IQR	Max
Adelie	female	73	37.3	37.0	2.0	32.1	35.9	38.8	2.9	42.2
Adelie	male	73	40.4	40.6	2.3	34.6	39.0	41.5	2.5	46.0
Chinstrap	female	34	46.6	46.3	3.1	40.9	45.4	47.4	2.0	58.0
Chinstrap	male	34	51.1	51.0	1.6	48.5	50.0	52.0	1.9	55.8
Gentoo	female	58	45.6	45.5	2.1	40.9	43.8	46.9	3.0	50.5
Gentoo	male	61	49.5	49.5	2.7	44.4	48.1	50.5	2.4	59.6

Figuur 16: Beschrijvende statistieken van snavellengte per pinguïnsoort en geslacht (in mm)

Prompt 5: *Bekijk de onderstaande tabel met beschrijvende statistieken van de snaveldiepte per pinguïnsoort en geslacht. Welke combinatie van soort en geslacht heeft de op één na laagste Q1-waarde? Welke combinatie van soort en geslacht heeft de op één na hoogste Q3-waarde? Leg je antwoord uit en verwijs daarbij expliciet naar de waarden in de tabel.*

Species	Sex	Count	Mean	Median	SD	Min	Q1	Q3	IQR	Max
Adelie	female	73	17.6	17.6	1.0	15.5	17.0	18.3	1.3	20.7
Adelie	male	73	19.1	18.9	1.0	17.0	18.5	19.6	1.1	21.5
Chinstrap	female	34	17.6	17.6	1.0	16.4	17.0	18.0	1.1	19.4
Chinstrap	male	34	19.3	19.3	1.0	17.5	18.8	19.8	1.0	20.8
Gentoo	female	58	14.2	14.2	1.0	13.1	13.8	14.6	0.8	15.5
Gentoo	male	61	15.7	15.7	1.0	14.1	15.2	16.1	0.9	17.3

Figuur 17: Beschrijvende statistieken van snaveldiepte per pinguïnsoort en geslacht (in mm)

A.2.3 Correlatie-inzicht en associatie

Prompt 6: *Op basis van de onderstaande correlatiematrix van numerieke variabelen: Welke variabele vertoont de sterkste positieve correlatie met lichaamsmassa? Hoeveel bedraagt die correlatie en wat betekent dat? Hoe zou je dit verband biologisch verklaren bij pinguïns?*

	Lichaamsmassa (g)	Snavellengte (mm)	Snaveldiepte (mm)	Flipperlengte (mm)
Lichaamsmassa (g)	NA	-0.23	0.65	0.59
Snavellengte (mm)	-0.23	NA	-0.58	-0.47
Snaveldiepte (mm)	0.65	-0.58	NA	0.87
Flipperlengte (mm)	0.59	-0.47	0.87	NA

Figuur 18: Correlatiematrix van numerieke variabelen

Prompt 7: *De onderstaande correlatiematrix toont een negatieve correlatie tussen snavellengte en snaveldiepte. Hoe interpreteer je dit verband? Wat betekent het wél en wat betekent het níét? Leg kritisch uit en ga in op mogelijke verklaringen.*

	Lichaamsmassa (g)	Snavellengte (mm)	Snaveldiepte (mm)	Flipperlengte (mm)
Lichaamsmassa (g)	NA	-0.23	0.65	0.59
Snavellengte (mm)	-0.23	NA	-0.58	-0.47
Snaveldiepte (mm)	0.65	-0.58	NA	0.87
Flipperlengte (mm)	0.59	-0.47	0.87	NA

Figuur 19: Correlatiematrix van numerieke variabelen

A.2.4 Statistische significantie

Prompt 8: *Bekijk de onderstaande resultaten van een t-test op lichaamsmassa bij mannelijke en vrouwelijke Adelie-pinguïns. Wat kun je besluiten over het verschil tussen de geslachten op basis van deze resultaten? Is dit verschil statistisch significant? Verwijs naar relevante statistieken in je antwoord.*

```
        Welch Two Sample t-test

data:  body_mass_g by sex
t = -13.126, df = 135.69, p-value < 2.2e-16
alternative hypothesis: true difference in means between group female and group male is not equal to 0
95 percent confidence interval:
 -776.3012 -573.0139
sample estimates:
mean in group female   mean in group male
            3368.836             4043.493
```

Figuur 20: T-test lichaamsmassa