[Date]

python-docx

[company name]

Statistic

**Deze samenvatting is gebaseerd op de slides ‘Lecture 1’ van Dennis Fok (Statistics for Data Science).**

Inhoudsopgave

# Lecture 1 — Statistics for Data Science (Volledige Samenvatting)

Erasmus School of Economics — Prof. Dennis Fok  
Periode: September – Oktober 2024

Deze samenvatting bevat een uitgebreide uitleg van alle slides uit Lecture 1, inclusief theorie, kernbegrippen, formules en Python‑voorbeelden.

# 1 Inleiding & Doel

De cursus vormt een (her)introductie tot de basis van statistiek met nadruk op kritisch denken: niet enkel **\*hoe\*** maar ook **\*waarom\*** analyses worden uitgevoerd.

Alles wordt toegepast in Python of R, met als leidraad het boek ‘Practical Statistics for Data Scientists’ (Bruce, Bruce & Gedeck, hoofdstukken 1–5).

Naast het technisch toepassen van methoden, wordt in deze cursus sterk de nadruk gelegd op het ontwikkelen van een kritische houding ten opzichte van data en statistische uitkomsten. Je leert niet alleen formules gebruiken, maar vooral ook onderliggende aannames doorzien en resultaten interpreteren binnen de context van het vraagstuk. Deze vaardigheden zijn essentieel om in de praktijk als data scientist valkuilen te vermijden en met vertrouwen beslissingen te nemen op basis van data.

# 2 Cursusstructuur en aanpak

De cursus bestaat uit zeven onderdelen:  
1. Basis van statistiek en inferentie  
2. Distributies, beschrijvende statistiek en hypothesetoetsing  
3. Toetsen van verschillen  
4. Lineaire regressie  
5. Diagnostiek en modelselectie  
6. Logistische regressie (GLM)  
7. Bayesian statistiek.  
  
Werkwijze: lezen → college → oefenen → wekelijkse opdrachten → deelopdrachten voor eindproject.

# 3 Doelen van statistiek

De drie hoofddoelen van statistiek:  
1. Data samenvatten (descriptieve statistiek)  
2. Verschillen toetsen (hypothese‑toetsing)  
3. Eigenschappen schatten van het datagenererend proces (inferentiële statistiek).  
  
Belangrijk: onzekerheid is altijd aanwezig. Resultaten berusten op aannames; wanneer die niet gelden, kunnen conclusies misleidend zijn.

Statistiek is geen trucendoos; het draait om logisch redeneren en het stellen van de juiste vragen bij elke stap van het analyseproces. Dit betekent dat je als data scientist altijd kritisch moet nagaan of de gekozen methode past bij de aard van je data en de vraagstelling. Bovendien leer je binnen deze cursus wanneer je voorzichtig moet zijn met het trekken van conclusies en hoe je valkuilen zoals bias, overfitting en misinterpretatie kunt herkennen en vermijden.

# 4 Belangrijke kernbegrippen

1. Data en variabelen:verzameling van gemeten kenmerken (features). Rol in analyse: afhankelijk (Y) of onafhankelijk (X).  
2. Steekproef en populatie:steekproef is subset van populatie; representativiteit cruciaal.  
3. Variatie en onzekerheid:toevalsvariatie, meetfouten, verschillen tussen respondenten.  
4. Modellen:vereenvoudigde weergaven van het datagenererend proces met expliciete aannames.

# 5 Typen data en variabelen

• Numeriek: continu (temperatuur) of discreet (aantal studenten)  
• Categoraal: binair (ja/nee), nominaal (kleur), ordinaal (eens/oneens)  
• Zonder duidelijke afhankelijke variabele → exploratieve statistiek In Python/R bepaalt het datatype vaak welke bewerkingen mogelijk zijn.

# 6 Steekproef vs populatie

Data kunnen afkomstig zijn uit experimenten of observaties. Controleer steeds:  
– zijn observaties onafhankelijk?  
– is er clustering (bijv. studenten binnen scholen)?  
– is er sampling bias?  
Voorbeeld van slechte steekproef: enquête op lokale markt of lage responsgraad in online survey.

# 7 Variatie, onzekerheid en significantie

Variatie is normaal. Significantie toetst of een waargenomen verschil groter is dan toevalsvariatie.  
  
\*\*Significant verschil:\*\* onwaarschijnlijk onder nulhypothese (H₀).  
\*\*Niet‑significant:\*\* verschil kan door toeval komen, maar sluit echt verschil niet uit.

# 8 Modellen en aannames

Een model beschrijft aannames over hoe data ontstaan. Cruciale quote: “All models are wrong, but some are useful.” (Box & Draper, 1987). Belangrijk: toets aannames (lineariteit, normaliteit, homoskedasticiteit).

# 9 Python‑implementatie (StatsModels)

Voorbeeldcode python:

|  |
| --- |
| import statsmodels.api as sm import statsmodels.formula.api as smf m = smf.ols('y ~ x1 + x2', data=df) r = m.fit() print(r.summary()) |

Beschikbare modellen: OLS, GLM, MixedLM, etc.

Met StatsModels kun je eenvoudig modellen specificeren door formules te gebruiken die vergelijkbaar zijn met de notatie in statistische literatuur. Bijvoorbeeld, een lineair regressiemodel kan worden opgezet met `**ols('Y ~ X1 + X2', data=df**)`. Vervolgens kun je de fit van het model beoordelen aan de hand van samenvattende statistieken zoals R², F-statistiek en p-waarden. Het is aan te raden om na elke analyse de aannames te evalueren via diagnostische plots, zoals **residualenplots en Q-Q plots**, zodat je zeker weet dat je interpretaties betrouwbaar zijn en niet berusten op foutieve aannames.

# 10 Workflow voor data‑analyse in Python

1. Importeren: `import pandas as pd`  
2. Data inlezen met `pd.read\_csv()`  
3. Verkennen met `df.describe()` en `df.plot()`  
4. Statistische berekeningen uitvoeren  
5. Resultaten documenteren en reproduceerbaar maken.

Een goede workflow begint vaak met het helder formuleren van je onderzoeksvraag. Daarna verzamel en structureer je de benodigde data, bijvoorbeeld door datasets te combineren of ontbrekende waarden te controleren. Vervolgens kies je geschikte analysemethoden, waarbij je steeds kritisch kijkt naar de aannames en eventuele beperkingen van je model. Tot slot is het belangrijk om je code en resultaten zó te documenteren dat anderen je stappen kunnen volgen en reproduceren – dat is echt goud waard in de wetenschap.

# 11 Beschrijvende statistiek

Gebruik grafische en numerieke samenvattingen om data te begrijpen:  
• `plot.scatter()`, `hist()`, `density()`, `boxplot()`  
Let op spreiding, vorm, correlatie en uitschieters (outliers).

# 12 entrale tendentie

Voor n waarnemingen X₁,…,Xₙ:  
• Gemiddelde: X̄ = (ΣXi)/n  
• Mediaan: 50%‑kwantiel  
• Modus: meest voorkomende waarde.  
Mediaan is robuuster dan gemiddelde bij uitschieters.

# 13 Spreidingsmaten

• Range = max − min  
• Interkwartielafstand (IQR) = Q₃ − Q₁  
• Outlier: < Q₁ − 1.5×IQR of > Q₃ + 1.5×IQR  
• Variantieschatting: s² = 1/(n−1) Σ(Xi−X̄)²  
• Standaardafwijking: s = √s²  
• Graden van vrijheid = aantal observaties − aantal geschatte parameters.

# 14 Hogere‑orde momenten

• 1e moment (verwachting): E[X] = μ  
• 2e moment (variantie): E[(X−μ)²] = σ²  
• 3e moment (scheefheid): E[(X−μ)³/σ³]  
• 4e moment (kurtosis): E[(X−μ)⁴/σ⁴]  
Voor normaalverdeling geldt: skew = 0, kurtosis = 3 (excess kurtosis = 0).

# 15 Toepassing in Python

Gebruik `.mean()`, `.median()`, `.var()`, `.std()`, `.skew()`, `.kurtosis()` op DataFrame‑kolommen voor directe berekening.

# 16 Samenvatting formules

• Gemiddelde: X̄ = (ΣXi)/n  
• Variantie: s² = (1/(n−1)) Σ(Xi−X̄)²  
• Standaardafwijking: s = √s²  
• IQR = Q₃ − Q₁  
• Outliers: < Q₁−1.5·IQR of > Q₃+1.5·IQR  
• Skewness = E[(X−μ)³/σ³]  
• Kurtosis = E[(X−μ)⁴/σ⁴]

# 17 Oefening en voorbereiding volgende week

Lees hoofdstuk 1–2 van het boek, maak opdrachten 1.1–1.4 en oefen met eigen datasets. Gebruik pandas.describe(), eenvoudige grafieken, en controleer verdelingen en relaties tussen variabelen.

# Lecture 2 – Statistics for Data Science

# 2.1. Distributies en basisbegrippen

Bij het werken met **statistische distributies** is het essentieel om te begrijpen **wat voor type variabele** men onderzoekt discreet of continu en om de bijbehorende **kansfuncties** correct te onderscheiden.  
Een distributie beschrijft namelijk **hoe waarschijnlijk** bepaalde waarden van een willekeurige variabele zijn. Dit inzicht vormt de basis voor bijna alle statistische analyses, van beschrijvende statistiek tot inferentiële methoden zoals hypothesetoetsing en regressieanalyse.

**Discrete variabelen**

Discrete variabelen kunnen slechts **afgebakende waarden** aannemen, zoals de uitkomst van een dobbelsteen (1–6) of het aantal kinderen in een gezin.  
Voor dergelijke variabelen worden twee fundamentele functies gebruikt:

* **Probability Mass Function (PMF)** –   
  Geeft de kans dat de variabele exact een bepaalde waarde aanneemt.
* **Cumulative Distribution Function (CDF)** –   
  Toont de kans dat de variabele kleiner dan of gelijk aan een bepaalde waarde is.

De CDF is dus de opgetelde kans tot en met een bepaalde waarde, terwijl de PMF de kans op één specifieke waardeweergeeft.  
Bijvoorbeeld: voor een eerlijke dobbelsteen is  voor , en de cumulatieve kans stijgt in stappen van .

**Continue variabelen**

Continue variabelen kunnen **elke waarde binnen een interval** aannemen, zoals lengte, gewicht of temperatuur. Omdat het aantal mogelijke waarden oneindig is, is de kans dat  precies één specifieke waarde aanneemt exact nul ().  
Daarom wordt niet met afzonderlijke kansen gewerkt, maar met **kansdichtheden**:

* **Probability Density Function (PDF)** – , zodat

De kans dat  binnen het interval  valt, is dus de oppervlakte onder de PDF tussen  en .

* **Cumulative Distribution Function (CDF)** –   
  De CDF is de integraal (oppervlakte) van de PDF en geeft de kans dat de variabele kleiner of gelijk is aan .  
  De afgeleide van de CDF levert de PDF: .

Interpretatie en betekenis

Het onderscheid tussen discrete en continue variabelen is cruciaal bij:

* het kiezen van de juiste **statistische toets** (bijv. binomiaal vs. normaal),
* het berekenen van **verwachtingswaarden en varianties**, en
* het visualiseren van data met histograms of dichtheidsplots.

Een goed begrip van kansfuncties voorkomt verkeerde interpretatie van data — bijvoorbeeld het verwarren van dichtheid (hoogte van de curve) met kans (oppervlakte onder de curve).

Vormkenmerken van verdelingen

Naast de locatie (gemiddelde) en spreiding (variantie) zijn er twee belangrijke **vormmaten**:

* **Scheefheid (Skewness):** meet de asymmetrie van de verdeling.  
  Positieve skewness → rechterstaart langer; negatieve → linkerstaart langer.
* **Kurtosis:** meet de ‘spitsheid’ of concentratie van de verdeling rond het gemiddelde.  
  Een normale verdeling heeft kurtosis ≈ 3; hogere waarden duiden op dikkere staarten.

Het analyseren van deze kenmerken helpt vast te stellen of een dataset **afwijkt van de normale verdeling** — een cruciale stap vóór het toepassen van parametrische statistische methoden.

Samenvattend:

Bij het werken met statistische distributies is het van groot belang om te begrijpen wat voor type variabele je onderzoekt — een **discrete** of een **continue** variabele — en om de bijbehorende kansfuncties correct te interpreteren. Een distributie geeft namelijk weer **hoe waarschijnlijk** bepaalde uitkomsten van een willekeurige variabele zijn. Dit vormt de basis voor het correct analyseren, interpreteren en modelleren van data.

Bij **discrete variabelen** gaat het om grootheden die slechts een beperkt aantal waarden kunnen aannemen, zoals het aantal keren dat een munt op “kop” valt of de uitkomst van een dobbelsteen. De verdeling wordt dan beschreven door de *probability mass function* (PMF), die de kans geeft dat de variabele een specifieke waarde aanneemt, aangeduid als . Daarnaast wordt vaak de *cumulative distribution function* (CDF) gebruikt, , die de kans weergeeft dat de uitkomst kleiner of gelijk is aan een bepaalde waarde.

Bij **continue variabelen** — zoals lengte, gewicht of temperatuur — kan de variabele oneindig veel waarden binnen een bepaald interval aannemen. Omdat de kans op één specifieke waarde precies nul is (), wordt de verdeling beschreven met een *probability density function* (PDF), aangeduid als . De kans dat een waarneming tussen twee waarden valt, wordt berekend via de oppervlakte onder de curve:

De *cumulative distribution function* is in dit geval de integraal van de PDF, en de afgeleide van de CDF is opnieuw de PDF: .

Het onderscheid tussen discrete en continue variabelen is essentieel bij het kiezen van de juiste analysemethode en het trekken van correcte conclusies uit data. Een misverstand tussen kans en kansdichtheid kan leiden tot verkeerde interpretaties van resultaten.

Naast het type kansfunctie zijn ook de **vormkenmerken van de verdeling** belangrijk. De **scheefheid (skewness)** geeft aan of de verdeling asymmetrisch is: een positieve scheefheid duidt op een langere rechterstaart, een negatieve op een langere linkerstaart. De **kurtosis** meet hoe spits of vlak de verdeling is ten opzichte van een normale verdeling: een hoge kurtosis wijst op dikkere staarten, een lage op plattere toppen.

Samen helpen deze concepten — het onderscheid tussen discrete en continue kansfuncties, en de vormkenmerken van de verdeling — bij het **begrijpen van de structuur van data** en het **toepassen van geschikte statistische modellen**. Ze vormen daarmee een onmisbare basis voor iedere datawetenschappelijke analyse.

# 2.2. Grafische illustraties

Voorbeelden van discrete en continue kansverdelingen

Een eenvoudig maar krachtig voorbeeld van een **discrete kansverdeling** is de **dobbelsteen**. Wanneer een eerlijke dobbelsteen wordt gegooid, zijn er zes mogelijke uitkomsten: 1, 2, 3, 4, 5 of 6. Elke uitkomst heeft een gelijke kans van één op zes.  
De probability mass function (PMF) wordt dan gegeven door:

De cumulative distribution function (CDF)  neemt toe in sprongen van  telkens wanneer een volgende waarde wordt bereikt. Deze trapvormige structuur is kenmerkend voor discrete verdelingen.

Bij een **continue kansverdeling** liggen de dingen anders. De variabele kan oneindig veel waarden aannemen binnen een interval, en de kans op één specifieke waarde is precies nul. De verdeling wordt dan beschreven met een probability density function (PDF), waarvan de oppervlakte onder de curve gelijk is aan 1.

Een eenvoudig voorbeeld van een continue verdeling is:

De bijbehorende cumulatieve verdelingsfunctie is:

De controle op de geldigheid van de PDF gebeurt door het berekenen van de integraal:

waaruit blijkt dat de totale kans inderdaad gelijk is aan 1.

Het verschil tussen deze twee voorbeelden maakt het onderscheid tussen discrete en continue kansverdelingen goed zichtbaar. In het discrete geval bestaat de verdeling uit afzonderlijke kanspunten, terwijl in het continue geval de kans zich over een oneindig aantal mogelijke waarden verspreidt en de oppervlakte onder de curve de kans vertegenwoordigt.

# 2.3. Python implementatie

Gebruik scipy.stats voor standaardverdelingen:  
from scipy import stats  
stats.norm.cdf(1, loc=0, scale=1) # standaardnormaal  
stats.binom.pmf(15, n=30, p=0.5) # binomiale kans  
stats.expon.ppf(0.95, scale=10) # 95e percentiel exponentieel

# 2.4. Belangrijke standaardverdelingen

Vier centrale verdelingen:  
1. Normaal: X ~ N(μ, σ²), f(x)=1/(σ√(2π)) exp(−½((x−μ)/σ)²)  
2. t-verdeling (tₙ)  
3. Chi-kwadraat (χ²ₖ)  
4. F-verdeling (F(d₁,d₂))

# 2.5. Schatting van distributies

Histogram = benadering van echte verdeling.  
Gebruik scipy.stats.ecdf(data) en fit():  
fit = stats.fit(stats.norm, data, bounds={'loc':(-4,4),'scale':(0,1)})  
fit.plot(plottype='qq')

# 2.6. Schattingsonzekerheid

We schatten μ en σ² uit steekproeven.  
Var(X̄) = σ²/n → onzekerheid daalt bij grotere n.  
Standaardfout: SE(X̄) = √(s²/n).

# 2.7. Betrouwbaarheidsintervallen

Als σ² bekend: X̄ ± 1.96√(σ²/n)  
Als σ² onbekend: X̄ ± t₀.₉₇₅,ₙ₋₁ √(s²/n)  
Voorbeeld (huisprijsdata): [4967.998, 5332.533].

# 2.8. Central Limit Theorem (CLT)

Wat als het N aantal waarnemingen klein is?

Wanneer het aantal waarnemingen (n) klein is, kunnen we niet zomaar aannemen dat het steekproefgemiddelde normaal verdeeld is. De **Centrale Limietstelling (CLT)** stelt weliswaar dat de verdeling van steekproefgemiddelden bij toenemende steekproefgrootte steeds beter de normale verdeling benadert, maar deze eigenschap geldt slechts **ongeveer**en **pas bij voldoende grote steekproeven**.

Bij kleine steekproeven (bijvoorbeeld n < 30) — zeker wanneer de onderliggende data afwijkt van een normale verdeling — kan dit leiden tot **onbetrouwbare of misleidende conclusies**. De gebruikelijke betrouwbaarheidsintervallen en hypothesetoetsen die zijn gebaseerd op de normale of t-verdeling, gaan dan mogelijk uit van verkeerde aannames. Hierdoor kunnen de berekende betrouwbaarheden of p-waarden afwijken van hun werkelijke waarden.

In de praktijk betekent dit dat het geschatte **betrouwbaarheidsinterval** niet altijd de bedoelde dekking heeft: een zogenaamd 95%-interval kan in werkelijkheid bijvoorbeeld maar 90% of juist 98% van de tijd het echte populatiegemiddelde bevatten. Ook kunnen statistische toetsen **te streng of te soepel** zijn, wat de kans op verkeerde beslissingen (type-I- of type-II-fouten) vergroot.

Wanneer de steekproefgrootte beperkt is, is het daarom belangrijk om **de verdeling van de data zorgvuldig te onderzoeken**. Als de verdeling duidelijk afwijkt van normaal, moet men extra voorzichtig zijn met het toepassen van standaard parametische methoden.

Een oplossing is het gebruik van **niet-parametrische methoden**, die geen specifieke veronderstellingen doen over de verdelingsvorm. Voorbeelden hiervan zijn de **Wilcoxon-toets** (als alternatief voor de t-toets) en de **Mann–Whitney-U-toets** voor onafhankelijke steekproeven. Deze toetsen baseren zich op de rangorde van de data in plaats van op de ruwe waarden, waardoor ze robuuster zijn bij niet-normale verdelingen.

Een andere veelgebruikte aanpak is de **bootstrap-techniek**. Hierbij wordt herhaaldelijk met teruglegging uit de steekproef getrokken om een empirische verdeling van de schatter (zoals het gemiddelde) op te bouwen. Op basis van deze herhalingen kunnen betrouwbaarheidsintervallen worden berekend zonder aan te nemen dat de data normaal verdeeld is. Dit maakt bootstrap-methoden bijzonder geschikt voor situaties met kleine steekproeven of onbekende verdelingen.

Samenvattend is het bij kleine steekproeven verstandig om:

* niet blind te vertrouwen op de Centrale Limietstelling,
* eerst de verdeling van de data te analyseren, en
* indien nodig alternatieve methoden te gebruiken die minder afhankelijk zijn van de aanname van normaliteit.

Voor **grote steekproeven** geldt wél dat:

en dus dat het steekproefgemiddelde zelfs bij niet-normale data **ongeveer normaal verdeeld** zal zijn.  
Maar zolang n klein blijft, zijn **de aannames van de CLT te zwak** om volledig op te vertrouwen, en moet men overschakelen op aangepaste of niet-parametrische methoden.

# 2.9. Samenvattend overzicht

**Stap 1: Probability Mass Function (PMF)**

**Definitie**  
De PMF beschrijft de kans dat een **discrete variabele** een specifieke waarde aanneemt:

Voorbeeld: bij een dobbelsteen .

**Belangrijke conclusie**  
De PMF vormt de basis van kansrekening bij discrete data.  
🡺 **Zonder PMF geen inzicht in de totale kansverdeling.**  
🡺Dit inzicht is nodig om over te stappen naar continue variabelen (PDF’s) en integralen als vervanging van sommen.

**Stap 2: Probability Density Function (PDF)**

**Definitie**  
Voor **continue variabelen** wordt de kansdichtheid beschreven door een PDF:

De kans op één exacte waarde is nul (), maar de kans over een interval wordt bepaald door de oppervlakte onder de curve.

**Belangrijke conclusie**  
PDF’s maken het mogelijk om kansen te berekenen bij continue data.  
🡺 **De focus verschuift van afzonderlijke waarden naar oppervlakten (integralen).**  
🡺 Deze gedachte is essentieel voor de stap naar de cumulatieve verdeling (CDF).

**Stap 3: Cumulative Distribution Function (CDF)**

**Definitie**  
De CDF geeft de opgetelde kans tot en met een bepaalde waarde:

en haar afgeleide is de PDF: .

**Belangrijke conclusie**  
De CDF biedt een volledig overzicht van de verdeling en vormt de schakel tussen kansdichtheid en realisaties in data.  
🡺 **Met F(x) kunnen we kwantielen, percentielen en overschrijdingskansen berekenen.**  
🡺 Dit begrip is cruciaal om straks betrouwbaarheidsintervallen te kunnen opstellen.

**Stap 4: Standaardfout (SE)**

**Definitie**  
De standaardfout meet de variatie van een schatter (zoals het steekproefgemiddelde):

**Belangrijke conclusie**  
De SE geeft aan **hoe precies** een steekproef het populatiegemiddelde benadert.  
🡺**Kleinere SE = grotere betrouwbaarheid van de schatting.**  
🡺 Deze maat vormt de bouwsteen voor het berekenen van betrouwbaarheidsintervallen.

**Stap 5: Betrouwbaarheidsintervallen (Confidence Intervals, CI)**

**Definitie**  
Op basis van de standaardfout kunnen we een interval schatten waarbinnen het populatiegemiddelde met een bepaalde waarschijnlijkheid ligt.

* Als σ bekend:
* Als σ onbekend:

**Belangrijke conclusie**  
CI’s vertalen de variantie van een schatting naar een interpreteerbaar interval.  
🡺**Ze vormen de brug tussen beschrijvende en inferentiële statistiek.**  
🡺Zonder CI’s weten we niet hoe betrouwbaar onze schatting van μ werkelijk is.

**Stap 6: Centrale Limietstelling (CLT)**

**Definitie**  
De CLT stelt dat voor grote steekproeven:

Zelfs als de onderliggende data niet normaal verdeeld is, wordt het steekproefgemiddelde bij grote n ongeveer normaal verdeeld.

**Belangrijke conclusie**  
De CLT rechtvaardigt het gebruik van normale en t-verdelingen bij inferentie.  
🡺**Bij grote n mogen we normaalmodellen gebruiken, zelfs bij niet-normale data.**  
🡺Bij kleine n moeten we echter alternatieven (zoals bootstrap of niet-parametrische methoden) overwegen.

**Eindconclusie**

Deze opeenvolgende stappen vormen samen het fundament van statistische inferentie:

1. PMF/PDF → beschrijven hoe kansen verdeeld zijn.
2. CDF → begrijpen hoe cumulatieve kansen zich opbouwen.
3. SE → kwantificeren hoeveel onzekerheid een schatting bevat.
4. CI → vertalen die onzekerheid naar interpreteerbare grenzen.
5. CLT → rechtvaardigen waarom deze aanpak wiskundig geldig is bij grote steekproeven.

# Lecture 3 – Hypothesetoetsing, t-testen & Correlatie

Statistics for Data Science (Erasmus University Rotterdam, Dennis Fok)

# 3.1 Inleiding: Waarom hypothesen toetsen?

In de statistiek onderzoeken we of de waargenomen verschillen tussen groepen of variabelen het resultaat zijn van toeval of werkelijk bestaan. Een hypothesetoets biedt ons de benodigde ondersteuning bij deze analyse.

Typische vraag: zijn twee “dingen” (groepen, gemiddelden, parameters) verschillend?  
Stappen in een toetsingsprocedure:

1. **Formuleer hypothesen**:

* **Nulhypothese (H₀)**: Dit is het uitgangspunt dat er geen effect of verschil is. Het stelt dat waargenomen verschillen te wijten zijn aan toeval.
* **Alternatieve hypothese (Hₐ)**: Dit is wat we proberen te bewijzen. Het stelt dat er wel een effect of verschil is tussen de groepen of variabelen.

1. **Verzamel data**: Dit kan gedaan worden door middel van experimenten, observaties of enquêtes, waarbij je ervoor zorgt dat de data representatief en betrouwbaar zijn.
2. **Bereken een teststatistiek**: Afhankelijk van het type hypothesetoets (bijvoorbeeld t-toets, chi-kwadraattoets) bereken je een waarde die de gegevens samenvat en vergelijkt met wat we onder de nulhypothese verwachten.
3. **Beoordeel extremiteit**: Bepaal de p-waarde of vergelijk de teststatistiek met een kritieke waarde uit een statistische distributie. De p-waarde geeft de kans aan om een resultaat te observeren dat even extremen of extremer is dan wat we hebben geobserveerd, als de nulhypothese waar is. Als deze kans klein is (meestal onder een gekozen alpha-niveau, zoals 0,05), dan is er onvoldoende bewijs om H₀ te behouden, en verwerpen we H₀.

**Belangrijk**: De conclusies van een hypothesetoets zijn probabilistisch. Een lage p-waarde betekent niet dat de nulhypothese absoluut onwaar is; het betekent simpelweg dat de gegevens niet goed passen bij de nulhypothese. Dit benadrukt het belang van het formuleren van hypotheses en het interpreteren van resultaten in de context van het onderzoek.

Of anders gezegd: we bewijzen nooit dat H₀ waar of onwaar is; we toetsen enkel of er voldoende bewijs is om het te verwerpen.

# 3.2 Significantie, fouten en betrouwbaarheid

Elke statistische toets kan fouten bevatten, die van invloed zijn op de resultaten en conclusies.

- **Type I-fout**: Dit gebeurt wanneer de nulhypothese (H₀) waar is, maar ten onrechte wordt verworpen. Dit staat bekend als een valse positieve bevinding.

- **Type II-fout**: Dit treedt op wanneer de nulhypothese (H₀) onwaar is, maar niet wordt verworpen. Dit wordt aangeduid als een valse negatieve bevinding.

**Significantieniveau (α)** Het significantieniveau is de kans op een Type I-fout en wordt meestal vastgesteld op 5%. Dit betekent dat we bereid zijn om in 5% van de gevallen een waarachtige nulhypothese ten onrechte te verwerpen.

**Power van een test**

De power van een test wordt gedefinieerd als \(1 - β\) en geeft de kans aan dat we de nulhypothese (H₀) correct verwerpen wanneer deze onwaar is. De power van een toets neemt toe bij:

- Een grotere steekproefgrootte (n↑), wat leidt tot nauwkeurigere schattingen.

- Een groter werkelijk effect (\(μ - μ₀↑\)), waardoor het makkelijker wordt om het verschil te detecteren.

- Een kleinere variantie (\(σ²↓\)), die de precisie van de schattingen vergroot.

Dus:  
Type I-fout: H₀ is waar, maar wordt verworpen (valse positieven).  
Type II-fout: H₀ is onwaar, maar wordt niet verworpen (valse negatieven).  
  
Significantieniveau (α): kans op Type I-fout, meestal 5%.  
Power van een test = 1 − β: kans dat we H₀ correct verwerpen.  
Power neemt toe bij grotere steekproef (n↑), groter werkelijk effect (μ − μ₀↑) en kleinere variantie (σ²↓).

# 3.3 Toetsingsstrategieën

Bij het toetsen van een hypothese zijn er twee belangrijke benaderingen: het gebruik van kritieke waarden en de bepaling van de p-waarde.

**A. Kritieke Waarden**

1. **Kies α**: Bepaal het significantieniveau, dat meestal 0,05 of 5% is.
2. **Bepaal de Kritieke Grens**: Afhankelijk van het gekozen α en het type toets (bijvoorbeeld éénzijdig of tweezijdig), bepaal je de kritieke grenswaarden die de kritieke zone definiëren.
3. **Bereken Teststatistiek**: Voer de toets uit en bereken de teststatistiek op basis van de verzamelde data.
4. **Verwerp H₀**: Als de berekende teststatistiek buiten de kritieke zone valt, verwerp dan de nulhypothese (H₀).

**B. p-Waarde**

1. **Bereken de p-waarde**: Bepaal de kans op een observatie die gelijk is aan of extremer is dan de berekende waarde, op voorwaarde dat de nulhypothese (H₀) waar is.
2. **Vergelijk met α**: Als de p-waarde kleiner is dan het gekozen α, verwerp dan de nulhypothese (H₀).
3. **Interpretatie**: De p-waarde geeft een indicatie van de sterkte van het bewijs tegen de nulhypothese. Hoe kleiner de p-waarde, des te sterker het bewijs tegen H₀.

# 3.4 Toetsen van gemiddelden – de t-test

Bij het toetsen van gemiddelden wordt vaak de one-sample t-test gebruikt. Deze test helpt bepalen of het gemiddelde van een steekproef significant verschilt van een bekend of verondersteld populatiegemiddelde. De nulhypothese (H₀) stelt dat het populatiegemiddelde (μ) gelijk is aan een specifieke waarde (μ₀), terwijl de alternatieve hypothese (Hₐ) stelt dat zij ongelijk zijn (μ ≠ μ₀).

De formule voor het berekenen van de t-statistiek is als volgt:  
t = (X̄ − μ₀) / (s / √n), met df = n − 1.  
  
**Poweranalyse**

Om de steekproefgrootte te bepalen die nodig is voor een bepaalde power van de t-test, kun je gebruikmaken van de functie statsmodels.stats.TTestPower. De power van een toets, dat is de kans om de nulhypothese correct te verwerpen wanneer deze onwaar is, neemt toe bij een grotere steekproefgrootte (n), een groter werkelijk effect en een kleinere variantie. Dit betekent dat je met een grotere steekproef meer vertrouwen hebt in het detecteren van significante verschillen indien deze bestaan.

# 3.5 Vergelijken van twee steekproeven

Variantietoets (F-test): H₀: σ₁² = σ₂² → verwerp als p < α.  
Gelijkheid van gemiddelden (t-test): H₀: μ₁ = μ₂ → gebruik gepoolde of Welch-versie afhankelijk van varianties.  
Meer dan 2 groepen → gebruik ANOVA of regressie.

# 3.6 Afhankelijke steekproeven

Bij gepaarde metingen (bijv. voor/na): bereken verschil Dᵢ = Xᵢ − Yᵢ en toets H₀: μ\_D = 0.  
Python: stats.ttest\_rel(X, Y).

# 3.7 Wanneer aannames niet gelden

Niet-normale data:  
 - Grote n → CLT geldt → t-test blijft geldig.  
 - Kleine n → gebruik bootstrap, permutatietest of niet-parametrische test.

# 3.8 Niet-parametrische toetsen

Wilcoxon signed-rank test: één steekproef of gepaarde metingen.  
Mann–Whitney U-test: twee onafhankelijke steekproeven.  
Binomiaaltest: aantal keren Xi > Yi.  
Voordeel: geen normaliteitsaanname. Nadeel: minder power bij normale data.

# 3.9 Bivariate beschrijvende statistiek

Covariantie: Cov(X,Y) = E[(X−E[X])(Y−E[Y])].  
Correlatie: Corr(X,Y) = Cov(X,Y) / √(Var[X]Var[Y]) ∈ [−1, +1].  
Gebruik scipy.stats.pearsonr(x, y) voor r en p-waarde.  
Let op: correlatie ≠ causaliteit.  
Andere maten: Spearman’s ρ, Kendall’s τ.

# 3.10 Oefeningen en voorbereiding

In-class 3.1: simulatie met N(0.05,1), t-test, poweranalyse.  
In-class 3.2: huizen met/zonder airco, variantie- en t-test (ook op log(price)).  
Volgende keer: Hoofdstuk 4 lezen en scatterplots + correlatieanalyse uitvoeren.

**Hypothesetoetsing, t-testen en Correlatie**

In de statistiek gebruiken we hypothesetoetsen om vast te stellen of verschillen tussen waargenomen waarden het gevolg zijn van toeval of dat ze wijzen op een werkelijk effect. Het uitgangspunt is steeds de **nulhypothese (H₀)**, die stelt dat er geen verschil of effect is, tegenover de **alternatieve hypothese (Hₐ)**, die een afwijking of verandering veronderstelt. We verzamelen data, berekenen een geschikte teststatistiek, en beoordelen vervolgens of deze waarde uitzonderlijk is als H₀ waar zou zijn. Wanneer de kans op zo’n extreme uitkomst klein genoeg is, concluderen we dat het onwaarschijnlijk is dat H₀ klopt en verwerpen we die. Toch bewijst een toets nooit dat H₀ waar of onwaar is – we spreken enkel over voldoende of onvoldoende bewijs om haar te verwerpen.

Elke statistische beslissing kan twee soorten fouten bevatten. Een **Type I-fout** treedt op wanneer we H₀ verwerpen terwijl ze eigenlijk waar is; de kans hierop noemen we het **significantieniveau (α)**, vaak ingesteld op 5 %. Een **Type II-fout** betekent dat we H₀ niet verwerpen terwijl ze onwaar is. De kans dat we wél correct verwerpen heet de **power** van de toets (1 − β). Die power wordt groter bij grotere steekproeven, bij grotere werkelijke verschillen en bij kleinere varianties. Het doel is een lage α en een hoge power te combineren.

Om te beslissen of een resultaat significant is, kunnen we twee methoden volgen. De klassieke methode werkt met **kritieke waarden**: we bepalen de grenzen die bij een gekozen α horen, en verwerpen H₀ wanneer de teststatistiek buiten dat gebied valt. De modernere benadering gebruikt **p-waarden**, die aangeven hoe waarschijnlijk de waargenomen of extremere waarde is als H₀ waar zou zijn. Een kleine p-waarde wijst op sterk bewijs tegen H₀; een grote p-waarde betekent dat de data goed verenigbaar zijn met H₀. In de praktijk is de p-waarde het handigst, omdat ze ook toelaat dat de lezer zelf zijn grens kiest.

Een veelgebruikte toets is de **t-test** voor gemiddelden. Bij een *one-sample t-test* vergelijken we het steekproefgemiddelde X̄ met een verondersteld populatiegemiddelde μ₀. De t-statistiek berekent hoe ver X̄ van μ₀ ligt in aantal standaardfouten. Is die afwijking groter dan verwacht op basis van de t-verdeling, dan verwerpen we H₀. De toets wordt uitgevoerd in Python met stats.ttest\_1samp(). Wanneer de variantie onbekend is, wordt ze geschat uit de data.

De kracht van een t-test hangt af van het **effect size** (het relatieve verschil tussen μ en μ₀ in standaarddeviaties), de steekproefgrootte n en het gekozen α. Via een poweranalyse (bijv. met statsmodels.stats.TTestPower) kan men vooraf berekenen hoeveel waarnemingen nodig zijn om met voldoende zekerheid een effect te kunnen detecteren.

Wanneer we twee steekproeven vergelijken, willen we eerst nagaan of hun varianties gelijk zijn. Dat doen we met de **Fisher-F-test**, die de verhouding tussen beide steekproefvarianties vergelijkt met de F-verdeling. Als de varianties niet significant verschillen, gebruiken we de gewone t-test met gepoolde variantie; anders nemen we de **Welch-t-test**, die robuuster is bij ongelijke varianties. Beide toetsen onderzoeken of de gemiddelden μ₁ en μ₂ gelijk zijn. Bij meer dan twee groepen schakelen we over op **ANOVA** of een lineair regressiemodel.

Soms zijn observaties **afhankelijk**, bijvoorbeeld bij voor- en nametingen van dezelfde personen. In dat geval toetsen we het gemiddelde van de verschillen (D = X − Y) tegen nul, wat neerkomt op een *paired t-test*. In Python gebeurt dit met stats.ttest\_rel().

De klassieke toetsen veronderstellen normaal verdeelde data. Als die aanname niet houdbaar is en de steekproef groot genoeg is, vangt de **centrale limietstelling** dit vaak op: het steekproefgemiddelde volgt dan toch ongeveer een normale verdeling. Bij kleine steekproeven gebruiken we liever alternatieve methoden, zoals **bootstrap-toetsen**, **permutatietests** of **niet-parametrische toetsen**. Deze laatste maken geen aannames over de verdeling en zijn daardoor robuuster, al hebben ze iets minder onderscheidend vermogen wanneer de normale aannames wel gelden.

Belangrijke niet-parametrische toetsen zijn de **Wilcoxon signed-rank test** voor één of gepaarde steekproeven en de **Mann–Whitney U-test** voor twee onafhankelijke groepen. Een eenvoudige **binomiaaltest** kan ook worden toegepast om te tellen hoe vaak waarden stijgen of dalen ten opzichte van elkaar. Al deze methoden zijn beschikbaar in scipy.stats.

Naast het toetsen van gemiddelden is het ook essentieel om de **relatie tussen twee variabelen** te beschrijven. De **covariantie** meet of twee grootheden samen toenemen of afnemen, maar is afhankelijk van de schaal. Daarom gebruiken we meestal de **correlatiecoëfficiënt (r)**, die wordt berekend als de gestandaardiseerde covariantie. De waarde van r ligt tussen −1 en +1: een positieve waarde wijst op een stijgend verband, een negatieve op een dalend verband, en nul op geen lineaire samenhang. In Python kan dit met stats.pearsonr(x, y), dat tegelijk een p-waarde geeft om te toetsen of de correlatie verschilt van nul.

Het is echter belangrijk om correlatie voorzichtig te interpreteren: **correlatie impliceert geen causaliteit**. Een verband kan veroorzaakt worden door een derde factor of een toevallige trend. Bekende voorbeelden tonen dat margarineverbruik sterk kan correleren met echtscheidingscijfers, of dat uitgaven aan wetenschap toevallig samenlopen met zelfmoordstatistieken – duidelijke gevallen van schijncorrelatie. Voor situaties waarin het verband niet lineair is, gebruiken we alternatieve maten zoals **Spearman’s ρ** of **Kendall’s τ**, beide gebaseerd op rangordes in plaats van ruwe waarden.

De colleges sluiten af met oefenopgaven. In *Assignment 3.1* wordt de t-test geoefend via simulatie en poweranalyse; in *Assignment 3.2* worden huizenprijzen met en zonder airconditioning vergeleken, zowel in originele waarden als na log-transformatie om scheefheid te verminderen. Als voorbereiding op de volgende les wordt gevraagd correlaties grafisch te verkennen met scatterplots, ze kwantitatief te berekenen en daarover een hypothesetoets uit te voeren.

**Kort samengevat:** hypothesetoetsen vormen het fundament van statistische besluitvorming. Ze helpen onderscheid maken tussen toevallige variatie en betekenisvolle effecten. De keuze voor de juiste toets hangt af van steekproefgrootte, variantie, afhankelijkheid en aannames over de verdeling. Niet-parametrische alternatieven en correlatie-analyse vullen dit kader aan, mits we de resultaten altijd in hun inhoudelijke en economische context interpreteren.

Kernboodschap:  
Hypothesetoetsen vormen de basis van statistische besluitvorming. Kies de juiste toets afhankelijk van steekproefgrootte, varianties, afhankelijkheid en verdelingsaanname. Gebruik altijd contextuele interpretatie en economische relevantie.

# Lecture 4 – OLS Regression & Diagnostics

## 1. Inleiding

Deze les bouwt voort op eerdere colleges over kansrekening en correlatie. Het doel is te begrijpen hoe we afhankelijkheid tussen variabelen kunnen modelleren met behulp van regressie, en hoe we daarna toetsen of het model betrouwbaar is.

## 2. Univariate Regression

Een regressie beschrijft het verband tussen één afhankelijke variabele (y) en één verklarende variabele (x).  
  
Model: y\_i = a + b x\_i + e\_i  
- a: intercept (verwachte waarde van y als x=0)  
- b: helling (verandering in y bij stijging van x met 1)  
- e\_i: residu (voorspellingsfout)

## 3. Ordinary Least Squares (OLS)

OLS kiest a en b zodat de som van de gekwadrateerde residuen minimaal is:  
min Σ(y\_i − a − b x\_i)²  
Oplossing: b = Σ(x\_i−x̄)(y\_i−ȳ)/Σ(x\_i−x̄)² en a = ȳ − b x̄

## 4. Statistische Eigenschappen van OLS

Voordelen: eenvoudig te berekenen, optimale eigenschappen onder aannames. Nadelen: gevoelig voor outliers en foutieve aannames.

## 5. De 6 Klassieke Aannames (Gauss–Markov)

A1: Variatie in x\_i  
A2: E[ε\_i]=0  
A3: Lineariteit  
A4: Homoskedasticiteit (gelijke variantie)  
A5: Geen autocorrelatie  
A6: Normaliteit ε\_i ~ N(0,σ²)  
→ Onder deze aannames is OLS de Best Linear Unbiased Estimator (BLUE).

## 6. Goodness-of-Fit

R² = 1 − SSE/SSY = SSR/SSY  
R² toont welk deel van de variantie in y door het model wordt verklaard.

## 7. Varianties en Standaardfout

s² = SSE/(n−k)  
SE\_b = s/√SSX  
Kleine SE ⇒ hoge precisie.

## 8. Hypothesetoetsing

H₀: β = 0  
t = b / SE\_b ~ t\_(n−k)  
Een hoge t-waarde ⇒ significante invloed van x op y.

## 9. Niet-lineaire modellen

Transformaties van x: x², log(x), √x, 1/x  
Transformaties van y: log(y)  
Interpretaties:  
- y = α + β log(x): 1% ↑x ⇒ β/100 ↑y  
- log(y) = α + β log(x): 1% ↑x ⇒ β% ↑y  
- log(y) = α + βx: 1 ↑x ⇒ ±100β% ↑y

## 10. Python Implementatie

import statsmodels.api as sm  
import statsmodels.formula.api as smf  
  
model = smf.ols(formula='price ~ lotsize', data=df)  
res = model.fit()  
print(res.summary())

## 11. Meervoudige Regressie

Model: y = Xβ + ε  
β̂ = (X'X)⁻¹X'y  
Gebruik t-test (individuele β) en F-test (meerdere β’s samen)  
Adjusted R² = 1 − [SSE/(n−k)] / [SSY/(n−1)]  
AIC balanceert model-fit en complexiteit.

## 12. Interpretatie (Ceteris Paribus)

Voorbeeld: log(income) = 7 + 0.01age + 0.025educ  
Leeftijd +1 ⇒ +1% inkomen; Educatie +1 ⇒ +2.5% inkomen.

## 13. Interactie-effecten

y = α + β₁x₁ + β₂x₂ + β₃x₁x₂ + ε  
De invloed van x₁ hangt af van x₂.  
Python: smf.ols(formula='y ~ x1 \* x2', data=df).fit()

## 14. Diagnostiek & Aannames Controleren

Gebruik olsdiagnostics:  
from olsdiagnostics import \*  
influence = OLSInfluence(model)  
diagnosticplots(influence)  
  
Vier plots: Residuals vs Fitted, QQ-plot, Scale-Location, Residuals vs Leverage.

## 15. Normaliteitstesten

Shapiro-Wilk: scipy.stats.shapiro(x)  
Jarque-Bera: scipy.stats.jarque\_bera(x)  
QQ-plot: statsmodels.api.qqplot(x, line='45')

## 16. Praktische Opdrachten

4.1 Housing data: price ~ lotsize  
4.2 Correlation & log-transformatie  
4.3 Murder-rate: Income, Population (+Interaction)  
Take-home: Beste model vinden via laagste AIC, hoogste AdjR² en diagnostische controle.

## 17. Samenvatting

OLS minimaliseert residuen.  
Aannames essentieel voor betrouwbaarheid.  
Gebruik R², t- en F-testen voor evaluatie.  
Diagnostiek controleert lineariteit, homoskedasticiteit, normaliteit en outliers.

# Lecture 6 – Statistics for Data Science

De zesde lecture van \*Statistics for Data Science\* bouwt voort op de klassieke lineaire regressie en introduceert drie belangrijke uitbreidingen: het werken met categorische variabelen, het toepassen van Generalized Linear Models (GLM) – in het bijzonder de logistische regressie – en het gebruik van de bootstrapmethode om onzekerheid te schatten. Samen vormen deze onderdelen een brug tussen traditionele lineaire modellen en modernere statistische technieken die geschikt zijn voor uiteenlopende datatypes.

## 1. Categorical explanatory variables

Tot nu toe werd regressie gebruikt om relaties te analyseren tussen kwantitatieve variabelen. In veel praktische situaties bevat een dataset echter ook kwalitatieve (categorische) kenmerken, zoals geslacht, merk, regio of opleidingsniveau. Om zulke variabelen te gebruiken in een regressiemodel moeten ze eerst numeriek worden weergegeven met behulp van dummyvariabelen.

Een dummyvariabele neemt enkel de waarden 0 of 1 aan. Zo kan bijvoorbeeld D = 1 voor “man” en D = 0 voor “vrouw” worden gedefinieerd. Wanneer we vervolgens een regressie schatten van de vorm y = α + βD + ε, dan geeft het coëfficiënt β het verschil weer tussen de gemiddelden van de twee groepen. Het toetsen van H0: β = 0 komt volledig overeen met de klassieke two-sample t-test. Beide benaderingen leveren identieke resultaten en steunen op dezelfde aannames, zoals gelijke varianties en (bij kleine steekproeven) normaliteit.

Bij categorische variabelen met meer dan twee categorieën worden meerdere dummies aangemaakt: voor K categorieën maken we K − 1 dummyvariabelen. Elke coëfficiënt drukt dan het verschil uit ten opzichte van een gekozen basiscategorie. In Python wordt dit proces grotendeels automatisch uitgevoerd via smf.ols() en de functie C(varname).

Om te beoordelen of een factor als geheel invloed heeft op de afhankelijke variabele, wordt een F-toets gebruikt die het model met en zonder de factor vergelijkt. Deze aanpak is equivalent aan de klassieke ANOVA-methoden (Analysis of Variance), zoals one-way en two-way ANOVA. In feite zijn ANOVA en regressie met dummyvariabelen slechts twee verschillende formuleringen van hetzelfde principe.

## 2. Generalized Linear Models (GLM)

Niet alle verschijnselen laten zich vangen in een continu afhankelijke variabele. Denk aan binaire uitkomsten (ja/nee), tellingen (aantal klanten), of duurmetingen (wachttijd). Voor zulke data is de normale verdeling ongeschikt, en dat vraagt om een meer algemene modelstructuur: de Generalized Linear Model (GLM).

Een GLM bestaat uit vier bouwstenen: (1) een geschikte verdelingsfunctie (bijvoorbeeld Bernoulli voor binaire data of Poisson voor tellingen); (2) een lineaire voorspeller η = Xβ; (3) een linkfunctie g(·) die de verwachte waarde koppelt aan de lineaire voorspeller; en (4) een variantiestructuur die afhangt van de verwachting.

Voor binaire uitkomsten wordt meestal de logistische linkfunctie gebruikt: P(Y = 1) = e^(Xβ) / (1 + e^(Xβ)), waarbij de kans automatisch tussen 0 en 1 ligt. Alternatieven zijn de probitlink (met de standaardnormale verdelingsfunctie Φ(Xβ)) en de complementary log-log link. In de praktijk leveren logit en probit vergelijkbare resultaten, al zijn de coëfficiënten verschillend geschaald.

Het schatten van een GLM gebeurt via Maximum Likelihood Estimation (MLE). Hierbij wordt de kans berekend om de waargenomen data te observeren, gegeven een set parameterwaarden. De parameters worden zo gekozen dat deze waarschijnlijkheid maximaal is. De bijbehorende maat voor modelkwaliteit is de deviance, die kleiner wordt naarmate het model beter past. Voor modelvergelijking worden vaak de Akaike Information Criterion (AIC) en de Likelihood Ratio Test (LRT) gebruikt.

## 3. Voorspellen en beoordelen van logistische modellen

Een logistisch of probitmodel voorspelt kansen P̂(Y=1). Wanneer een binaire voorspelling nodig is, worden deze kansen omgezet naar 0/1-waarden op basis van een drempel c. De standaardkeuze is c = 0,5, maar deze kan worden aangepast om de balans tussen fout-positieven en fout-negatieven te verbeteren.

De kwaliteit van voorspellingen wordt samengevat in een confusion matrix, waarin de correcte en foutieve classificaties worden geteld. Belangrijke maatstaven zijn de accuracy (het percentage juiste voorspellingen) en de sensitivity of recall (het aandeel correct voorspelde positieven).

Omdat de parameters in een logistisch model niet direct de omvang van effecten aangeven, worden ze vaak vertaald naar marginale effecten: de verandering in kans op Y=1 bij een kleine verandering in een verklarende variabele. Voor de logistische link geldt: ∂P(Y=1)/∂xj = P(Y=1)(1−P(Y=1))βj. Gemiddeld over alle observaties spreken we van de Average Marginal Effect (AME), die veel intuïtiever te interpreteren is.

## 4. Bootstrap – schatten van onzekerheid door simulatie

In de klassieke statistiek wordt onzekerheid meestal benaderd via asymptotische theorie: men veronderstelt dat steekproeven groot genoeg zijn zodat de schatters normaal verdeeld zijn. In werkelijkheid zijn steekproeven echter eindig, en deze benadering kan onnauwkeurig blijken.

De bootstrapmethode biedt een elegant alternatief. In plaats van te vertrouwen op theoretische formules, gebruikt bootstrap herhaaldelijke steekproeftrekking met teruglegging uit de bestaande data. Voor elke nieuwe steekproef (de bootstrap sample) wordt de gewenste statistiek opnieuw berekend, bijvoorbeeld het gemiddelde of een regressiecoëfficiënt.

Door dit proces duizenden keren te herhalen ontstaat een empirische verdeling van de schatter. Deze kan vervolgens worden gebruikt om varianties te schatten of betrouwbaarheidsintervallen te construeren, bijvoorbeeld op basis van de 2,5e en 97,5e percentielen. Het resultaat is een praktisch, simulatie-gebaseerd alternatief voor ingewikkelde wiskundige afleidingen — al werkt de methode niet altijd, bijvoorbeeld bij afhankelijke observaties of bij extreme-waarde-schattingen.

## 5. Conclusie

Lecture 6 introduceert drie essentiële uitbreidingen van lineaire regressie. Ten eerste leren we hoe we categorische informatie kunnen integreren via dummyvariabelen, waardoor regressie en ANOVA in feite één en dezelfde methode blijken. Ten tweede biedt de GLM-familie, met name de logistische regressie, een krachtig raamwerk om binaire of niet-normale uitkomsten te modelleren op basis van kansfuncties in plaats van lineaire voorspellingen. Ten slotte maakt de bootstrap het mogelijk om betrouwbaarheidsintervallen en varianties te schatten zonder te vertrouwen op theoretische aannames. Samen maken deze technieken het statistische gereedschap veelzijdiger, robuuster en beter toepasbaar op de complexe datavormen waarmee moderne datawetenschappers werken.

# Lecture 7 – Bayesian Statistics & Wrap-Up

De zevende en laatste lecture van \*Statistics for Data Science\* vormt de afsluiting van het vak en bouwt voort op de concepten uit vorige lessen, in het bijzonder Generalized Linear Models (GLM) en Bootstrap. Het centrale thema is de introductie van de Bayesiaanse statistiek – een fundamenteel andere manier van denken over kans, onzekerheid en leren uit data.

## 1. Van frequentist naar Bayesiaans denken

Tot nu toe werd gewerkt binnen het frequentistische paradigma, waarin kans wordt opgevat als een langetermijnfrequentie van gebeurtenissen. Parameters in dat kader worden beschouwd als vaste, maar onbekende waarden die men probeert te schatten op basis van steekproefdata.

De Bayesiaanse benadering daarentegen ziet kans als een graad van overtuiging (degree of belief). Parameters zijn in deze visie zelf ook willekeurige variabelen, waarvoor we onze onzekerheid uitdrukken via een kansverdeling. Waar de frequentist zich afvraagt ‘wat zou een andere steekproef hebben opgeleverd?’, vraagt de Bayesiaan zich af ‘hoeveel leert mijn huidige steekproef mij over mijn overtuiging?’.

Bayesiaanse statistiek draait om het bijwerken van kennis: wat we vooraf geloven (prior) wordt aangepast met nieuwe informatie uit de data (likelihood) om te komen tot een geüpdatete overtuiging, de posterior.

## 2. Parameterleren: prior, likelihood en posterior

In frequentistische statistiek schatten we parameters via procedures zoals OLS of Maximum Likelihood en bepalen we onzekerheid via asymptotische benaderingen of bootstrap. Bayesianen volgen vier stappen:

1. Start met een priorverdeling – een formele weergave van wat we vooraf geloven over de parameter.  
2. Gebruik de data om de likelihood te bepalen: de kans dat we de waargenomen gegevens zien, gegeven een bepaalde parameterwaarde.  
3. Combineer prior en likelihood volgens de stelling van Bayes:  
 f(β | data) ∝ f(data | β) × f(β)  
4. Gebruik de posterior om puntschattingen, varianties en credible intervals te berekenen.

De posterior bevat volledige informatie over de parameteronzekerheid. Daarmee kan men bijvoorbeeld de kans berekenen dat een parameter groter is dan nul, of dat één parameter groter is dan een andere – iets wat binnen de frequentistische school formeel niet kan.

## 3. Intuïtie via voorbeelden

Het college illustreert Bayesiaans redeneren met eenvoudige voorbeelden zoals het dobbelsteenvoorbeeld en het Monty Hall-probleem, waaruit blijkt hoe nieuwe informatie overtuigingen bijwerkt.

Een concreet voorbeeld betreft een munt met een Beta(2,2)-prior. Na het gooien van twee keer kop ontstaat een posterior Beta(4,2), met een gemiddelde kans op kop van ongeveer 2/3. De frequentistische schatting zou 1 zijn, terwijl de Bayesiaanse aanpak nuance behoudt door de priorinformatie te verwerken.

## 4. Eigenschappen, voordelen en beperkingen

Voordelen van de Bayesiaanse aanpak:  
• Exact en geldig bij kleine steekproeven.  
• Intuïtieve interpretatie: kans dat een hypothese waar is.  
• Eenvoudig bijwerken van kennis bij nieuwe data.  
• Integratie van expertinformatie via de prior.

Nadelen:  
• Keuze van prior is subjectief.  
• Rekentechnisch intensief bij complexe modellen.  
• Foutieve prior of modelaanname kan tot verkeerde conclusies leiden.

## 5. De rol van de prior

De prior heeft vooral meerwaarde bij beperkte informatie, zoals kleine steekproeven, veel parameters of weinig observaties per individu. Een vaak gebruikte prior is N(μ, σ²), waarbij μ de verwachte waarde uitdrukt (vaak 0 bij geen voorkennis) en σ² de sterkte van de overtuiging weergeeft. Een kleine σ² betekent een sterke overtuiging (informatieve prior), een grote σ² wijst op een zwakke overtuiging (ongeïnformeerde prior).

## 6. Praktische toepassingen

Bayesiaanse methoden worden breed toegepast in situaties met beperkte data of dynamische informatie. Voorbeelden zijn:  
• Nieuwe productontwikkeling – modelleren van marktsucces met weinig data.  
• Product ranking – combineren van klantreviews met onzekerheid.  
• A/B-testing – real-time updates van resultaten.  
• Financiële voorspellingen – omgaan met onverwachte gebeurtenissen.  
• Medische diagnostiek – combineren van testresultaten met voorafkansen.

## 7. Berekenen van de posterior

Wanneer prior en likelihood goed bij elkaar passen spreken we van een conjugate prior, zoals het voorbeeld van de Beta-Binomiale combinatie. In veel gevallen is echter geen analytische oplossing mogelijk en wordt gebruikgemaakt van simulatiemethoden, met name Markov Chain Monte Carlo (MCMC). Door iteratief te simuleren uit de voorwaardelijke verdelingen van parameters wordt de posterior benaderd.

## 8. Bayesiaanse analyse in Python

Het college introduceert de bibliotheek Bambi als toegankelijke interface voor PyMC. Deze stelt automatisch priors in en voert Bayesiaanse analyses uit. Een voorbeeld:

import arviz as az  
import bambi as bmb  
  
model = bmb.Model('y ~ x1 + x2', data)  
fitted = model.fit(random\_seed=1234)  
az.summary(fitted)  
az.plot\_trace(fitted)

Met Bambi kunnen ook niet-lineaire modellen worden uitgevoerd, zoals logistische of Poisson-regressies, via het argument family='bernoulli' of 'poisson'.

## 9. Conclusie

De Bayesiaanse benadering biedt een krachtig alternatief voor traditionele statistiek. In plaats van vaste, onbekende parameters gebruikt men kansverdelingen die zich voortdurend aanpassen aan nieuwe informatie. Waar frequentistische methoden puntschattingen en asymptotische aannames gebruiken, biedt de Bayesiaanse aanpak een volledig probabilistisch beeld van onzekerheid – intuïtiever, flexibeler en realistischer.