Hoofdstuk 1: Basis en Datasets

Functies en Methodes (Datasets)

Functie/Methode	Omschrijving	Voorbeeld	Output (voorbeeld)
`read_csv()`	Leest een CSV-bestand in een DataFrame	`titanic = pd.read_csv('https://raw.githubusercontent.com/DataRepo2019/Data-files/master/titanic.csv')`	DataFrame met Titanic-data geladen
`head()`	Toont de eerste rijen van een DataFrame	`titanic.head()`	Eerste 5 rijen van DataFrame
`tail()`	Toont de laatste rijen van een DataFrame	`print(not_null_df.tail(3))`	Laatste 3 rijen
`info()`	Toont informatie over kolommen en datatypes	`titanic.info()`	Overzicht van kolomnamen, niet-null counts, datatypes
`describe()`	Geeft statistische samenvatting	`print(titanic.Survived.describe())`	Count, mean, std, min, max enz.
`count()`	Telt niet-NA waarden per kolom	`print(titanic.count())`	Aantal niet-NA per kolom
`sum()`	Som van waarden	`print(not_null_df.sum())`	Som per kolom of rij
`mean()`	Gemiddelde van waarden	`avg_age = titanic['Age'].mean()`	Eén getal (gemiddelde leeftijd)
`unique()`	Unieke waarden in een kolom	`print(titanic.Embarked.unique())`	Array van unieke waarden
`value_counts()`	Telt hoe vaak unieke waarden voorkomen	`print(titanic.dtypes.value_counts())`	Tabel met telling per uniek type
`astype()`	Zet kolom om naar een ander datatype	`titanic.Survived = titanic.Survived.astype('category')`	Kolomtype aangepast
`drop()`	Verwijdert kolommen of rijen	`titanic.drop("PassengerId", axis="columns")`	DataFrame zonder opgegeven kolom

Functie/Methode	Omschrijving	Voorbeeld	Output (voorbeeld)
`dropna()`	Verwijdert rijen met NA-waarden	`cleaned = titanic.dropna()`	DataFrame zonder NA-rijen
`fillna()`	Vult NA-waarden op	`titanic = titanic.fillna(value={'Age' : avg_age})`	DataFrame met ingevulde waarden
`set_index()`	Zet een kolom als index	`titanic.set_index(['PassengerId'])`	DataFrame met nieuwe index
`map()`	Past een functie toe op een Series	`titanic['Sex'] = titanic['Sex'].map(mapping_dict)`	Kolom met aangepaste waarden
`query()`	Filtert data met een query-string	`titanic.query("(Sex=='male') and (Age < 18)")`	Gefilterde DataFrame
`round()`	Rondt waarden af	`print(f"(Rounded) Average age of passengers: {round(avg_age)}")`	Afgerond getal
`notnull()`	Controleert of waarden niet null zijn	`not_null_df = titanic.notnull()`	DataFrame met True/False waarden

Extra (niet direct pandas maar gezien in notebook)

Functie	Omschrijving	Voorbeeld	Output (voorbeeld)
`print()`	Print tekst naar console	`print(f"Number of rows: {len(titanic)}")`	"Number of rows: 891" (bijv.)
`len()`	Geeft lengte van object	`print(f"Number of rows: {len(titanic)}")`	Getal met aantal rijen
`and`	Logische EN	`titanic.query("(Sex=='male') and (Age < 18)")`	Gefilterde DataFrame

Visualisaties (Grafieken)

Naam Grafiek	Functie	Omschrijving	Voorbeeld	Output (voorbeeld)
Countplot	`countplot()`	Toont telling van categorieën	`sns.countplot(data=titanic, x='Embarked');`	Balkgrafiek met tellingen per categorie

Hoofdstuk 2: Univariate Analysis

Functies en Methodes

Functie/Methode	Omschrijving	Voorbeeld	Output (voorbeeld)
`load_dataset()`	Laadt voorbeelddataset (Seaborn)	`tips = sns.load_dataset("tips")`	DataFrame met tips-data
`head()`	Toont eerste rijen van DataFrame	`tips.head()`	Eerste 5 rijen
`mean()`	Gemiddelde	`print(f"Mean: {tips.tip.mean()}")`	Gemiddelde waarde

Functie/Methode	Omschrijving	Voorbeeld	Output (voorbeeld)
`median()`	Mediaan	`print(f"Median: {tips.tip.median()}")`	Mediaanwaarde
`mode()`	Modus	`tips.mode()`	Waarde(n) die het vaakst voorkomen
`min()`	Minimumwaarde	`print(f"Minimum: {tips.tip.min()}")`	Kleinste waarde
`max()`	Maximumwaarde	`print(f"Maximum: {tips.tip.max()}")`	Grootste waarde
`var()`	Variantie	`print(f"Variance: {tips.tip.var()}")`	Variantie (n-1 in noemer)
`std()`	Standaardafwijking	`print(f"Standard deviation: {tips.tip.std()}")`	Std-dev
`quantile()`	Percentielen	`print(f"Percentiles {percentiles}\n{tips.tip.quantile(percentiles)}")`	Waarden op percentielen
`describe()`	Statistische samenvatting	`tips.day.describe()`	Count, mean, std, min, max enz.
`kurtosis()`	Scheefheid (spitsheid)	`print(f"Kurtosis: {tips.tip.kurtosis()}")`	Kurtosis-waarde
`skew()`	Scheefheid	`print(f"Skewness: {tips.tip.skew()}")`	Skewness-waarde
`iqr()`	Interkwartielafstand	`print(f"Inter Quartile Range: {stats.iqr(tips.tip)}")`	IQR-waarde
`array()`	Maakt numpy-array	`a = np.array([4, 8, 6, 5, 3, 2, 8, 9, 2, 5])`	Numpy-array
`arange()`	Maakt range-array	`population = np.arange(0, 101)`	Array van 0 tot 100
`choice()`	Willekeurige selectie uit array	`sample = np.random.choice(population, size=sample_size)`	Array met random waarden
`sum()`	Som van waarden	`mean = sum(x) / n`	Totale som
`len()`	Lengte van object	`n = len(x)`	Getal met lengte
`sqrt()`	Vierkantswortel	`return np.sqrt(pop_var(x))`	Wortelwaarde

Visualisaties (Grafieken)

Naam Grafiek	Functie	Omschrijving	Voorbeeld	Output (voorbeeld)
Displot	`displot()`	Histogram met verdelingscurve	`sns.displot(data=tips, x='tip');`	Histogram
KDE Plot	`kdeplot()`	Kernel Density Estimate plot	`sns.kdeplot(data=tips, x='tip');`	Gladde verdelingslijn
Boxplot	`boxplot()`	Boxplot	`sns.boxplot(data=tips, x='tip');`	Boxplot met mediaan, IQR enz.
Violinplot	`violinplot()`	Violinplot	`sns.violinplot(data=tips, x='tip');`	Violinvormige plot
Catplot (count)	`catplot(kind='count')`	Categorische tellingen plot	`sns.catplot(data=tips, x='day', kind='count');`	Balkgrafiek met tellingen

Hoofdstuk 3: Probability

Functies en Methodes

Functie/Methode	Omschrijving	Voorbeeld	Output (voorbeeld)

Functie/Methode	Omschrijving	Voorbeeld	Output (voorbeeld)
`arange()`	Maakt een array met stappen	`x = np.arange(4)`	Array [0,1,2,3]
`linspace()`	Verdeelt bereik in evenredige stappen	`x = np.linspace(mu - 4 * sigma, mu + 4 * sigma, num=201)`	Array met 201 waarden
`randint()`	Genereert willekeurige gehele getallen	`random.randint(1,6)`	Getal tussen 1-6
`normal()`	Genereert normale verdeling	`observations = np.random.normal(loc=m, scale=s, size=n)`	Array met normaal verdeelde waarden
`full()`	Array vullen met constante waarde	`y = np.full(6, 1/6)`	Array [1/6,1/6,...]
`append()`	Voegt waarde toe aan lijst	`l_game_1.append(number_of_times_won / index)`	Gewijzigde lijst
`range()`	Genereert reeks getallen	`for index in range(1, number_of_games + 1):`	1 tot number_of_games
`pdf()`	Probability density function (scipy.stats)	`y = stats.norm.pdf(x, mu, sigma)`	Waarden van de kansdichtheid
`cdf()`	Cumulative density function (scipy.stats)	`stats.norm.cdf(1.62, loc=0, scale=1)`	Kanswaarde
`sf()`	Survival function (1 - CDF) (scipy.stats)	`stats.norm.sf(1.62, loc=0, scale=1)`	Complementaire kanswaarde
`isf()`	Inverse survival function (scipy.stats)	`stats.norm.isf(0.05, loc=0, scale=1)`	Waarde bij gegeven overlevingskans
`interval()`	Bereken betrouwbaarheidsinterval	`stats.norm.interval(confidence, loc=m, scale=s/√n)`	Tuple met (lower, upper) grens
`sqrt()`	Vierkantswortel	`lo = m - z * s / np.sqrt(n)`	Wortelwaarde
`subplots()`	Maakt figuur en subplot	`fig, tplot = plt.subplots(1, 1)`	Lege figuur met as
`plot()`	Lijnplot tekenen	`tplot.plot(x, stats.norm.pdf(x, 0, 1))`	Lijngrafiek
`legend()`	Voeg legenda toe aan plot	`tplot.legend(loc='best')`	Legenda op plot

Visualisaties (Grafieken)

Naam Grafiek/Plot	Functie	Omschrijving	Voorbeeld	Output (voorbeeld)
Lijnplot (seaborn)	`lineplot()`	Lijnplot voor trends	`sns.lineplot(ax=ax, x='number of games', y='game 1 die', data=data)`	Lijngrafiek
Staafdiagram (bar)	`bar()`	Staafdiagram	`ax.bar(x, y, 0.35)`	Staafdiagram
Histogram	`histplot()`	Histogram met KDE	`sns.histplot(observations, kde=True)`	Histogram met verdelingslijn
Vulgebied	`fill_between()`	Kleurt gebied onder curve	`plt.fill_between(dist_x, 0, dist_y, where=dist_x <= x, color='lightblue')`	Ingekleurde grafiek
Verticale lijn	`axvline()`	Verticale lijn in grafiek	`plt.axvline(75, color='green')`	Verticale lijn

Naam Grafiek/Plot	Functie	Omschrijving	Voorbeeld	Output (voorbeeld)
Subplots	`subplots()`	Meerdere grafieken naast elkaar	`f, ax = plt.subplots(1, 1)`	Lege subplot
Figure	`figure()`	Nieuwe figuur starten	`fig = plt.figure()`	Lege figuur
Add Axes	`add_axes()`	Handmatig assen toevoegen	`ax = fig.add_axes([0,0,1,1])`	Assen toegevoegd
Show Plot	`show()`	Toon de plot	`plt.show()`	Toont plot op scherm
Plot (lijn)	`plot()`	Lijnplot (matplotlib)	`plt.plot(x, stats.norm.pdf(x, 0, 1))`	Lijngrafiek
X-as label	`xlabel()`	Zet label op x-as	`plt.xlabel('number of games')`	Label op x-as
Y-as label	`ylabel()`	Zet label op y-as	`plt.ylabel('win percentage')`	Label op y-as
Set X Label	`set_xlabel()`	Zet label op x-as via Axes	`ax.set_xlabel('x')`	Label op x-as
Set Y Label	`set_ylabel()`	Zet label op y-as via Axes	`ax.set_ylabel('P(X = x)')`	Label op y-as
Titel toevoegen	`title()`	Voeg titel toe aan plot	`plt.title('Standard Normal Distribution')`	Titel boven plot
X-limieten	`xlim()`	Beperk x-as bereik	`plt.xlim((0,20))`	X-as limiet
Y-limieten	`ylim()`	Beperk y-as bereik	`plt.ylim((0,1))`	Y-as limiet
Legenda toevoegen	`legend()`	Voeg legenda toe	`plt.legend()`	Legenda op plot

Symbolische Notaties

Symbool	Naam	Betekenis
∩	Doorsnede	A én B (snijpunt van gebeurtenissen)
∪	Unie	A of B (alles wat in A of B zit)
A^c	Complement	Alles wat niet in A zit
		Voorwaardelijke kans
μ (mu)	Gemiddelde (verwachtingswaarde)	Gemiddelde van verdeling
σ (sigma)	Standaardafwijking	Spreiding rond het gemiddelde
α (alpha)	Significantieniveau	Kans buiten betrouwbaarheidsinterval (bijv. 0,05)

Kansregels en Formules

Begrip	Betekenis	Formule
Kans	Kans dat gebeurtenis A gebeurt	P(A)
Complementregel	Kans dat A niet gebeurt	$P(A^c) = 1 - P(A)$
Productregel (onafhankelijk)	Kans op A én B als onafhankelijk	$P(A \cap B) = P(A) * P(B)$
Productregel (afhankelijk)	Kans op A én B als afhankelijk	$P(A \cap B) = P(A) * P(B$

Begrip	Betekenis	Formule
Somregel (disjunct)	Kans op A of B (disjunct)	P(A ∪ B) = P(A) + P(B)
Somregel (niet-disjunct)	Kans op A of B	P(A ∪ B) = P(A) + P(B) - P(A ∩ B)
CDF (cumulatief)	Kans dat X kleiner of gelijk aan x	P(X ≤ x) = CDF(x)
SF (survival)	Kans dat X groter is dan x	P(X > x) = SF(x) = 1 - CDF(x)
Z-score (kritieke waarde)	Kritieke waarde uit standaardnormale verdeling	z = isf(α/2)
Standaardfout (SE)	Standaardafwijking van steekproefgemiddelde	SE = s / √n
Bereken z-score voor interval	Stap 1 voor betrouwbaarheidsinterval	z = stats.norm.isf(α/2)
Bereken SE voor interval	Stap 2 voor betrouwbaarheidsinterval	SE = s / √n
Bereken onder- en bovengrens	Stap 3 voor betrouwbaarheidsinterval	lo = m − z _ SE; hi = m + z _ SE
α (alpha)	Kans buiten interval (significantieniveau)	α = 1 − confidence

Formule (handmatig):

- Bereken z-score: `z = stats.norm.isf(α/2)`
- Bereken standaardfout: `SE = s / √n`
- Ondergrens: `lo = m − z * SE`
- Bovengrens: `hi = m + z * SE`

Uitleg: (Standaard)normale verdeling

De **standaardnormale verdeling** is een speciale normale verdeling met gemiddeld **μ = 0** en standaardafwijking **σ = 1**. Het is een klokvormige symmetrische curve die wordt gebruikt om veel natuurlijke verschijnselen te modelleren.

Eigenschappen:

- Symmetrisch rond het gemiddelde **μ = 0**
- 68% van de waarden ligt binnen 1 standaardafwijking (σ)
- 95% van de waarden ligt binnen 2 standaardafwijkingen (σ)
- 99,7% van de waarden ligt binnen 3 standaardafwijkingen (σ)

Python-voorbeeld:

```
x = np.linspace(-4, 4, 100)
y = stats.norm.pdf(x, 0, 1)
plt.plot(x, y)
plt.title("Standaardnormale verdeling")
plt.show()
```

Uitleg: Wat is een z-score?

Een **z-score** geeft aan hoeveel standaardafwijkingen een waarde van het gemiddelde af ligt.

Formule:

> z = (x − μ) / σ

Wat doet het? Het zet elke waarde uit een normale verdeling om naar de standaardnormale schaal, zodat je kanswaarden kunt opzoeken met bijvoorbeeld `cdf()`, `sf()` of `isf()`.

Voorbeeld: Stel je hebt een waarde **x = 80**, met een gemiddelde **μ = 70** en een standaardafwijking **σ = 5**, dan geldt:

> z = (80 − 70) / 5 = 10 / 5 = 2

Dit betekent dat de waarde 2 standaardafwijkingen boven het gemiddelde ligt.

Wat is α (alpha)? De **α-waarde** (significantieniveau) geeft aan hoeveel kans je buiten een betrouwbaarheidsinterval laat. Bijvoorbeeld:

- Bij 95% betrouwbaarheidsinterval: α = 1 − 0,95 = 0,05
- Bij 99% betrouwbaarheidsinterval: α = 1 − 0,99 = 0,01

Omdat de interval symmetrisch verdeeld is, gebruik je vaak **α/2** voor de bovenste of onderste kant.

Python-voorbeeld:

```
confidence = 0.95
m = 50
s = 10
n = 100
z = stats.norm.isf((1 - confidence) / 2)
lo = m - z * s / np.sqrt(n)
hi = m + z * s / np.sqrt(n)
print(f"Confidence interval: [{lo:.2f}, {hi:.2f}]")
```

Hypothesis Testing: Overzicht en Voorwaarden

Wanneer gebruik je welke test?

- **z-test** → als je een grote steekproef hebt (n ≥ 30), de standaardafwijking σ van de populatie kent, en de data normaal verdeeld zijn.
- **t-test** → als je een kleine steekproef hebt (n < 30) of σ onbekend is.

Stappenplan Hypothesis Testing

1 Formuleer de hypotheses

- H_0: nulhypothese (bijvoorbeeld: er is geen verschil)
- H_1: alternatieve hypothese (bijvoorbeeld: er is wél verschil)

2 Kies significantieniveau

- Meestal α = 0,05 (5%)

3 Bereken de teststatistiek

- $z = (\bar{x} - \mu) / (\sigma / \sqrt{n})$ → bij z-test
- $t = (\bar{x} - \mu) / (s / \sqrt{n})$ → bij t-test

4 Bepaal het kritieke gebied

- Right-tailed: kijk naar bovenkant (H_1: $\mu > \mu_0$)
- Left-tailed: kijk naar onderkant (H_1: $\mu < \mu_0$)
- Two-tailed: kijk naar beide kanten (H_1: $\mu \neq \mu_0$)

5 Maak een beslissing

- Als p-waarde ≤ α → verwerp H_0
- Als p-waarde > α → behoud H_0

Python formules voor variabelen right tailed z test

Symbool	uitleg	Python formule
n	grootte van steekproef	`len(array)`
μ (mu)	gemiddelde (niet van steekproef)	/
m_sample	gemiddelde (steekproef)	`np.mean(array)`
p	/	`stats.norm.sf(m_sample, loc=mu, scale=sigma/np.sqrt(n))`

Symbool	uitleg	Python formule
g	/	`stats.norm.isf(1-alpha, loc=mu, scale=(sigma/ np.sqrt(n)))`

Python formules voor variabelen left tailed z test

Symbool	uitleg	Python formule
n	grootte van steekproef	`len(array)`
μ (mu)	gemiddelde (niet van steekproef)	/
m_sample	gemiddelde (steekproef)	`np.mean(array)`
p	/	`stats.norm.cdf(m_sample, loc=mu, scale=sigma/np.sqrt(n))`
g	/	`stats.norm.isf(1-alpha, loc=mu, scale=(sigma/ np.sqrt(n)))`

Python formules voor variabelen two-tailed z test

Symbool	uitleg	Python formule
n	grootte van steekproef	`len(array)`
μ (mu)	gemiddelde (niet van steekproef)	/
m_sample	gemiddelde (steekproef)	`np.mean(array)`
p	/	`stats.norm.sf(m_sample, loc=mu, scale=sigma/np.sqrt(n))`
g1	linkerkant	`mu - stats.norm.isf(1-alpha, loc=mu, scale=(sigma/ np.sqrt(n)))`
g2	rechterkant	`mu + stats.norm.isf(1-alpha, loc=mu, scale=(sigma/ np.sqrt(n)))`

Voorbeeld: Right-tailed z-test

```
n = 50
x_bar = 105
mu = 100
sigma = 10
z = (x_bar - mu) / (sigma / np.sqrt(n))
p_value = stats.norm.sf(z)
```

Voorbeeld: Left-tailed z-test

```
n = 50
x_bar = 95
mu = 100
sigma = 10
z = (x_bar - mu) / (sigma / np.sqrt(n))
p_value = stats.norm.cdf(z)
```

Voorbeeld: Two-tailed z-test

```
n = 50
x_bar = 105
mu = 100
sigma = 10
z = (x_bar - mu) / (sigma / np.sqrt(n))
p_value = 2 * stats.norm.sf(abs(z))
```

Voorbeeld: Right-tailed t-test

```
n = 15
x_bar = 105
mu = 100
s = 10
t = (x_bar - mu) / (s / np.sqrt(n))
p_value = stats.t.sf(t, df=n-1)
```

Voorbeeld: Left-tailed t-test

```
n = 15
x_bar = 95
mu = 100
s = 10
t = (x_bar - mu) / (s / np.sqrt(n))
p_value = stats.t.cdf(t, df=n-1)
```

Voorbeeld: Two-tailed t-test (met `ttest_1samp`)

```
sample = np.array([98, 102, 100, 105, 97])
mu = 100
t_stat, p_value = stats.ttest_1samp(sample, popmean=mu)
```

Symbolen en Formules

Symbool	Betekenis
H_0	Nulhypothese
H_1	Alternatieve hypothese
α	Significantieniveau (bijv. 0,05)
p	p-waarde, kans op verkregen resultaat onder H_0
z	z-score bij z-test
t	t-score bij t-test
μ	Populatiegemiddelde
$\bar{x}$	Steekproefgemiddelde
σ	Populatiestandaardafwijking
s	Steekproefstandaardafwijking
n	Steekproefgrootte

Wat is een Type I en Type II fout?

- **Type I fout (α):** je verwerpt H_0 terwijl die waar is → vals alarm.
- **Type II fout (β):** je behoudt H_0 terwijl H_1 waar is → je mist een effect.

Gevolgen:

- Type I fout: je denkt dat er effect is, maar dat is er niet.
- Type II fout: er is wél effect, maar je ziet het niet.

Hoofdstuk 4: Chi-Squared & Associaties

Welke test gebruik je wanneer?

Onafhankelijke Variabele (Independent)	Afhankelijke Variabele (Dependent)	Test/Metric
Kwalitatief	Kwalitatief	χ^2-test, Cramér's V
Kwalitatief	Kwantitatief	two-sample t-test, Cohen's d
Kwantitatief	Kwantitatief	Regressie, correlatie

Uitleg per test

- **χ^2-test:** Verband tussen twee categorische variabelen.
- **Goodness-of-Fit test:** Vergelijkt verdeling van één categorische variabele met een verwachte verdeling.
- **Cramér's V:** Sterkte van het verband tussen categorische variabelen.
- **Two-sample t-test:** Vergelijkt gemiddelden tussen twee onafhankelijke groepen.
- **Gepaalde t-test:** Vergelijkt gemiddelden tussen twee gekoppelde metingen (voor/na).
- **Cohen's d:** Effectgrootte van verschil tussen twee groepen.
- **Regressie (least squares):** Voorspelt waarden van een afhankelijke kwantitatieve variabele uit een onafhankelijke kwantitatieve variabele.
- **Covariantie:** Meet hoe twee kwantitatieve variabelen samen variëren.
- **Correlatie (Pearson):** Meet de sterkte en richting van een lineair verband tussen twee kwantitatieve variabelen.
- **Coëfficiënt van determinatie (R^2):** Geeft aan hoeveel van de variantie verklaard wordt door het model.

Wat doet de χ^2-test?

De **χ^2-test** controleert of er een verband is tussen twee categorische variabelen door de geobserveerde frequenties in een kruistabel te vergelijken met de verwachte frequenties als er géén verband zou zijn.

Wat doet de Goodness-of-Fit test?

De **Goodness-of-Fit test** controleert of de verdeling van één categorische variabele overeenkomt met een verwachte verdeling (bijvoorbeeld een uniforme verdeling of een andere theoretische verdeling).

Beste grafieken (met voorbeeld)

Grafiektype	Omschrijving	Voorbeeldcode
Mozaïekdiagram	Visualiseert frequenties tussen categorieën	`mosaic(data, ['col1', 'col2'])`
Gestapelde staafdiagram (met percentages)	Vergelijkt proporties per categorie	`pd.crosstab(df['col1'], df['col2'], normalize='index').plot(kind='bar', stacked=True)`

Functies voor χ^2, Goodness-of-Fit en Cramér's V

Functie/Methode	Omschrijving	Voorbeeld
`pd.crosstab()`	Maakt een kruistabel (contingency table)	`table = pd.crosstab(df['col1'], df['col2'])`
`chi2_contingency()`	Voert de χ^2-test uit op een kruistabel	`chi2, p, dof, expected = stats.chi2_contingency(table)`
`chisquare()`	Voert de goodness-of-fit test uit	`chi2, p = stats.chisquare(observed, f_exp=expected)`
Handmatige Cramér's V	Berekent Cramér's V	zie functie hieronder

Handmatige berekening van Cramér's V

```
dof = min(observed.shape) - 1
cramers_v = np.sqrt(chi_squared / (n * dof))
```

```
print(cramers_v)
```

Cramér's V	Interpretation
0	No association
0.1	Weak association
0.25	Moderate association
0.50	Strong association
0.75	Very strong association
1	Complete association

Stappenplan χ^2-test (contingentie)

1 Formuleer de hypotheses

- H_0: Er is geen verband tussen de variabelen.
- H_1: Er is wel een verband tussen de variabelen.

2 Maak een kruistabel

- Gebruik bijvoorbeeld `pd.crosstab()`.

3 Voer de test uit

- Gebruik `stats.chi2_contingency()`.

4 Bekijk de resultaten

- χ^2-waarde, p-waarde, vrijheidsgraden, expected counts.

5 Maak een beslissing

- Als $p \leq \alpha \rightarrow$ verwerp H_0.
- Als $p > \alpha \rightarrow$ behoud H_0.

Stappenplan Goodness-of-Fit test

1 Formuleer de hypotheses

- H_0: De geobserveerde verdeling komt overeen met de verwachte.
- H_1: De verdeling wijkt af van de verwachte.

2 Bereken de teststatistiek

- Gebruik `stats.chisquare(observed, f_exp=expected)`.

3 Maak een beslissing

- Als $p \leq \alpha \rightarrow$ verwerp H_0.
- Als $p > \alpha \rightarrow$ behoud H_0.

Voorbeeld χ^2 Goodness-of-Fit test in Python

```
import numpy as np
from scipy import stats

observed = np.array([30, 50, 20])
expected = np.array([33.3, 33.3, 33.3])
chi2, p = stats.chisquare(observed, f_exp=expected)
print(f"Chi²: {chi2:.2f}, p-waarde: {p:.4f}")
```

Symbolen en Formules

Symbool	Betekenis
χ^2	Chi-kwadraat statistiek
p	p-waarde, kans onder nulhypothese
dof	Degrees of freedom (vrijheidsgraden)
Cramér's V	Sterkte van het verband tussen variabelen

Hoofdstuk 5: t-tests & Effectgrootte

Beste grafieken (met voorbeeld)

Grafiektype	Omschrijving	Voorbeeldcode
Density plot	Toont verdeling van een continue variabele	`sns.kdeplot(data, x='value', hue='group')`
Staafdiagram met foutbalken	Vergelijkt gemiddelden met onzekerheid (error bars)	`sns.barplot(data=df, x='group', y='score', ci='sd')`

Functies voor t-tests en Cohen's d

Functie/Methode	Omschrijving	Voorbeeld
`ttest_ind()`	t-test voor twee onafhankelijke steekproeven	`stats.ttest_ind(group1, group2)`
`ttest_rel()`	t-test voor gepaarde steekproeven	`stats.ttest_rel(before, after)`
`cohen_d()` (zelf gedefinieerd)	Berekent Cohen's d (effectgrootte) handmatig	Zie functie hieronder

Formule van Cohen's d

```
def cohen_d(a, b):
    na = len(a)
    nb = len(b)
    pooled_sd = np.sqrt(((na - 1) * np.var(a, ddof=1) + (nb - 1) * np.var(b, ddof=1)) / (na + nb - 2))
    return (np.mean(b) - np.mean(a)) / pooled_sd
```

Interpretatie van Cohen's d (Effect Size)

|d|	Effectgrootte
0,01	Zeer klein
0,20	Klein
0,50	Gemiddeld
0,80	Groot
1,20	Zeer groot
2,00	Enorm

Stappenplan t-test

1 Formuleer de hypotheses

- H_0: Er is geen verschil tussen de groepen.
- H_1: Er is wel een verschil tussen de groepen.

2 Kies de juiste test

- Onafhankelijke groepen → `ttest_ind()`
- Gepaalde metingen (voor/na) → `ttest_rel()`

3 Bereken de teststatistiek

- t-waarde en p-waarde met `stats.ttest_ind()` of `stats.ttest_rel()`

4 Bereken de effectgrootte (optioneel)

- Cohen's d met de handmatige `cohen_d()` functie

5 Maak een beslissing

- Als $p \leq \alpha \rightarrow$ verwerp H_0.
- Als $p > \alpha \rightarrow$ behoud H_0.

Voorbeeld: t-test voor twee onafhankelijke steekproeven

```
from scipy import stats
t_stat, p_value = stats.ttest_ind(group1, group2)
print(f"t: {t_stat:.3f}, p: {p_value:.4f}")
```

Voorbeeld: t-test voor gepaarde steekproeven

```
from scipy import stats
t_stat, p_value = stats.ttest_rel(before, after)
print(f"t: {t_stat:.3f}, p: {p_value:.4f}")
```

Voorbeeld: Cohen's d

```
d = cohen_d(group1, group2)
print(f"Cohen's d: {d:.3f}")
```

Symbolen en Formules

Symbool	Betekenis
t	t-statistiek
p	p-waarde, kans onder nulhypothese
μ	Populatiegemiddelde
$\bar{x}$	Steekproefgemiddelde
s	Steekproefstandaardafwijking
n	Steekproefgrootte
Cohen's d	Effectgrootte, verschil in standaarddeviaties

Hoofdstuk 6: Regressie, Covariantie & Correlatie

Beste grafieken (met voorbeeld)

Grafiektype	Omschrijving	Voorbeeldcode
Rel plot	Scatterplot met regressielijn of trends	`sns.relplot(data=df, x='x', y='y', kind='scatter')`

Grafiektype	Omschrijving	Voorbeeldcode
Regressieplot	Lijnfit met betrouwbaarheidsbanden	`sns.regplot(x='x', y='y', data=df)` of `sns.lmplot(x='x', y='y', data=df)`

Methode van de kleinste kwadraten (Method of Least Squares)

De kleinste-kwadratenmethode vindt de lijn (of regressievergelijking) die de som van de kwadraten van de residuen (afstanden tussen werkelijke waarden en voorspelde waarden) minimaliseert.

Formules:

- Regressielijn: $y = \beta_0 + \beta_1 \times x$
- Hellingscoëfficiënt (β_1): $\beta_1 = \text{Cov}(X, Y) / \text{Var}(X)$
- Intercept (β_0): $\beta_0 = \bar{y} - \beta_1 \times \bar{x}$

Voorbeeld β_0 en β_1 berekenen

```
import numpy as np
x = np.array([1, 2, 3, 4, 5])
y = np.array([2, 4, 5, 4, 5])

beta_1 = np.cov(x, y, ddof=1)[0, 1] / np.var(x, ddof=1)
beta_0 = np.mean(y) - beta_1 * np.mean(x)

print(f"β₁ (helling): {beta_1:.2f}")
print(f"β₀ (intercept): {beta_0:.2f}")
```

Covariantie

Begrip	Omschrijving	Voorbeeldcode
Covariantie	Meet de mate waarin twee variabelen samen variëren; positief = samen omhoog/omlaag, negatief = tegengesteld	`np.cov(x, y)[0, 1]`

Correlatie

- Meet de sterkte en richting van een lineair verband tussen twee variabelen.
- Waarden tussen -1 en 1; dicht bij 0 = geen lineair verband.

Functie en voorbeeld:

```
corr = np.corrcoef(x, y)[0, 1]
print(f"Correlatie: {corr:.2f}")
```

Coëfficiënt van determinatie (R^2)

- Geeft het percentage verklaarde variantie aan (hoe goed past het model?).
- $R^2 = r^2$, waar r de correlatiecoëfficiënt is.

Voorbeeld:

```
r = np.corrcoef(x, y)[0, 1]
r_squared = r ** 2
print(f"R²: {r_squared:.2f}")
```

R	R^2	Verklaarde variantie	Lineair verband

R	R^2	Verklaarde variantie	Lineair verband
< 0,3	< 0,1	< 10%	zeer zwak
0,3 - 0,5	0,1 - 0,25	10% - 25%	zwak
0,5 - 0,7	0,25 - 0,5	25% - 50%	matig
0,7 - 0,85	0,5 - 0,75	50% - 75%	sterk
0,85 - 0,95	0,75 - 0,9	75% - 90%	zeer sterk
> 0,95	> 0,9	> 90%	uitzonderlijk sterk

Tabel met gebruikte termen

Term	Betekenis
Regressielijn	De beste lijn door de data (voorspelling)
Intercept (β_0)	Snijpunt met de y-as
Hellingscoëfficiënt (β_1)	Hoeveel y verandert als x met 1 stijgt
Covariantie	Samenhang in variatie tussen x en y
Correlatie (R)	Sterkte en richting lineair verband
Coëfficiënt van determinatie (R^2)	Hoeveel variantie door model verklaard wordt
Residuen	Verschil tussen geobserveerde en voorspelde waarden

Hoofdstuk 7: Time Series Analyse

Belangrijkste componenten

Component	Definitie
Level	Het gemiddelde of basisniveau van de reeks over tijd
Trend	Langetermijnbeweging (stijgend of dalend patroon)
Seasonal	Terugkerende patronen of fluctuaties over vaste perioden

Wanneer gebruik je welke methode?

Methode	Wanneer gebruiken?	Voorbeeldcode
Simple Moving Average (SMA)	Als je kortetermijnschommelingen wilt gladstrijken	`series.rolling(window=3).mean()`
Simple Exponential Smoothing (SES)	Voor tijdreeksen met alleen level, zonder trend of seizoen	`SimpleExpSmoothing(series).fit()`
Double Exponential Smoothing (DES)	Voor tijdreeksen met level én trend, maar zonder seizoen	`ExponentialSmoothing(series, trend='add').fit()`
Triple Exponential Smoothing (TES, Holt-Winters)	Voor tijdreeksen met level, trend én seizoen (add => mul om van additief naar multiplicatief verband)	`ExponentialSmoothing(series, trend='add', seasonal='add', seasonal_periods=12).fit()` * add = schommelingen zijn even groot mul = schommelingen worden groter

Uitgewerkte voorbeelden

Simple Moving Average (SMA)

```
import pandas as pd
sma = series.rolling(window=3).mean()
print(sma.head())
```

Dit berekent het gemiddelde van telkens 3 opeenvolgende waarden om ruis weg te filteren.

Simple Exponential Smoothing (SES)

```
from statsmodels.tsa.holtwinters import SimpleExpSmoothing
model = SimpleExpSmoothing(series).fit()
pred = model.predict(start=0, end=len(series)-1)
print(pred.head())
```

Gebruik SES als er geen trend of seizoen is, enkel een stabiel gemiddelde.

Double Exponential Smoothing (DES)

```
from statsmodels.tsa.holtwinters import ExponentialSmoothing
model = ExponentialSmoothing(series, trend='add').fit()
pred = model.predict(start=0, end=len(series)-1)
print(pred.head())
```

Gebruik DES als de data een duidelijke stijgende of dalende trend vertoont.

Triple Exponential Smoothing (TES / Holt-Winters)

```
model = ExponentialSmoothing(series, trend='add', seasonal='add', seasonal_periods=12).fit()
pred = model.predict(start=0, end=len(series)-1)
print(pred.head())
```

Gebruik TES als je ook seizoensinvloeden wilt meenemen (bijvoorbeeld maandelijkse data met jaarcyclus).

Evaluatiematen

Maat	Definitie	Voorbeeldcode
MAE	Gemiddelde absolute fout tussen voorspelde en werkelijke waarden	`from sklearn.metrics import mean_absolute_error` `mae = mean_absolute_error(y_true, y_pred)`
MSE	Gemiddelde van de kwadraten van de fouten (straft grotere fouten sterker)	`from sklearn.metrics import mean_squared_error` `mse = mean_squared_error(y_true, y_pred)`

Decomposing a Time Series

Decompositie splitst een tijdreeks op in zijn **level**, **trend**, **seasonal** en **residual** componenten om patronen beter te begrijpen.

Voorbeeld decompositie

```
from statsmodels.tsa.seasonal import seasonal_decompose
import matplotlib.pyplot as plt

result = seasonal_decompose(series, model='additive', period=12)
result.plot()
plt.show()
```

Samengevat

- **SMA**: gladstrijken van korte termijn fluctuaties.
- **SES**: voor stabiele reeksen zonder trend/seizoen.
- **DES**: voor reeksen met trend.
- **TES**: voor reeksen met trend én seizoenspatronen.
- **MAE/MSE**: beoordelen hoe nauwkeurig je voorspellingen zijn.
- **Decompositie**: inzicht krijgen in de bouwstenen van je tijdreeks.