

expand\_more

**Erstelle eine einfache Simulation in NetLogo, bei der Bürger und Behörden eine eID verwenden können. Die Bürger können eine eID haben und haben zu dieser eine normalverteilte Einstellung von 0 bis 100. Anfangs haben 10% der Bürger eine eID. Die Behörden können Dienste mit eID anbieten, die Schwierigkeit der Nutzung der eID kann von 0 bis 100 gehen. Anfangs bieten 10% aller Behörden Dienste mit eID an. Die Bürger laufen umher und treffen gelegentlich auf eine Behörde. Können sie dort ihre eID nicht einsetzen, so sinkt ihre Einstellung zur eID. Können sie ihre eID einsetzen, so entwickelt sich ihre Einstellung entsprechend der Schwierigkeit der Nutzung: Ist der Einsatz leicht, so steigt ihre Einstellung, ist der Einsatz schwer, so sinkt ihre Einstellung.**

expand\_more

volume\_up

**NetLogo-Simulation: Bürger und Behörden mit eID**

**Modellbeschreibung:**

Diese Simulation modelliert die Interaktion zwischen Bürgern und Behörden in einer Welt, in der die elektronische Identitätskarte (eID) eingeführt wurde. Bürger können eine eID besitzen und haben dazu eine normalverteilte Einstellung von 0 bis 100. Behörden können Dienste mit eID anbieten, und die Schwierigkeit der Nutzung der eID für diese Dienste kann von 0 bis 100 variieren.

**Modellsetup:**

* **Agenten:**
  + Bürger: Repräsentieren die Bürger in der Simulation.
  + Behörden: Repräsentieren die Behörden, die Dienste anbieten.
* **Variablen:**
  + eID-Einstellung: Die Einstellung eines Bürgers zur eID (normalverteilt, 0-100).
  + eID-Besitz: True/False, ob ein Bürger eine eID besitzt.
  + eID-Dienste: True/False, ob eine Behörde Dienste mit eID anbietet.
  + Nutzungs-Schwierigkeit: Die Schwierigkeit der Nutzung der eID für einen Dienst (0-100).
* **Parameter:**
  + Anfangsproportion\_eID\_Bürger: Der anfängliche Anteil der Bürger mit eID (z. B. 0,1).
  + Anfangsproportion\_eID\_Behörden: Der anfängliche Anteil der Behörden, die Dienste mit eID anbieten (z. B. 0,1).
* **Verhalten:**
  + **Bürger:**
    - Laufen zufällig umher in der Umgebung.
    - Interagieren mit Behörden, wenn sie ihnen begegnen.
    - Aktualisieren ihre eID-Einstellung basierend auf ihren Erfahrungen mit der eID.
  + **Behörden:**
    - Bieten Dienste an, unabhängig von der Interaktion mit Bürgern.
    - Beeinflussen die eID-Einstellung der Bürger durch die Schwierigkeit der Nutzung der eID in ihren Diensten.

**Modelldynamik:**

* **Setup:**
  + Initialisiere die Welt mit Bürgern und Behörden.
  + Verteile eID-Besitz und eID-Dienste zufällig gemäß den Anfangsparametern.
  + Weise jedem Bürger eine eID-Einstellung zu.
  + Weise jeder Behörde eine Nutzungs-Schwierigkeit zu.
* **Laufzeit:**
  + Bürger laufen umher und interagieren mit Behörden.
  + Bei jeder Interaktion:
    - Prüfe, ob der Bürger eine eID besitzt und ob die Behörde Dienste mit eID anbietet.
    - Wenn ja, nutze die eID für den Dienst.
    - Aktualisiere die eID-Einstellung des Bürgers basierend auf der Nutzungs-Schwierigkeit.
* **Datenanalyse:**
  + Beobachte die Entwicklung der eID-Einstellungen der Bürger im Laufe der Zeit.
  + Analysiere den Einfluss von eID-Besitz, eID-Diensten und Nutzungs-Schwierigkeit auf die eID-Einstellungen.

**Erweiterungen:**

* Implementieren Sie verschiedene Arten von eID-Diensten mit unterschiedlichen Auswirkungen auf die Bürger.
* Modellieren Sie die Kosten für die Beschaffung und Nutzung einer eID.
* Berücksichtigen Sie die Rolle von Vertrauen und Sicherheit in der Akzeptanz der eID.
* Fügen Sie ein Modell der öffentlichen Meinung und des politischen Drucks hinzu.

**Beispielcode:**

globals

num-citizens

num-agencies

agents

citizens

patches

eid-attitude

has-eid

agencies

patches

offers-eid-services

usage-difficulty

patches

xcor

ycor

setup

set num-citizens 100

set num-agencies 20

create-citizens num-citizens

create-agencies num-agencies

;; Initialisiere eID-Besitz und eID-Dienste

;; ...

;; Weise jedem Bürger eine eID-Einstellung zu

;; ...

;; Weise jeder Behörde eine Nutzungs-Schwierigkeit zu

;; ...

end

go

;; Bürger laufen umher und interagieren mit Behörden

;; ...

;; Aktualisiere die eID-Einstellungen der Bürger

;; ...

end

**Hinweis:** Dies ist nur ein einfaches Beispiel. Die konkrete Implementierung der Modelllogik und der Datenanalyse kann je nach Ihren spezifischen Forschungsfragen variieren.

share

more\_vert



**erstelle den netlogo code**

volume\_up

**NetLogo-Code für die eID-Simulationsmodell**

globals

num-citizens

num-agencies

agents

citizens

patches

eid-attitude

has-eid

agencies

patches

offers-eid-services

usage-difficulty

patches

xcor

ycor

setup

set num-citizens 100

set num-agencies 20

create-citizens num-citizens

create-agencies num-agencies

;; Initialisiere eID-Besitz (10% der Bürger haben anfangs eine eID)

set initial-eid-proportion 0.1

foreach citizen in citizens

if random-float < initial-eid-proportion

set citizen.has-eid true

end

end

;; Initialisiere eID-Dienste (10% der Behörden bieten anfangs eID-Dienste an)

set initial-eid-services-proportion 0.1

foreach agency in agencies

if random-float < initial-eid-services-proportion

set agency.offers-eid-services true

end

end

;; Weise jedem Bürger eine eID-Einstellung zu (normalverteilt, 0-100)

set mean-eid-attitude 50

set stddev-eid-attitude 20

foreach citizen in citizens

set citizen.eid-attitude random-normal mean-eid-attitude stddev-eid-attitude

end

;; Weise jeder Behörde eine Nutzungs-Schwierigkeit zu (0-100)

foreach agency in agencies

set agency.usage-difficulty random 101

end

end

go

;; Bürger laufen umher und interagieren mit Behörden

foreach citizen in citizens

let neighbor-agencies patches-with agentset agencies in 1

let agency-with-eid-services one-of neighbor-agencies with offers-eid-services = true

if agency-with-eid-services != null and citizen.has-eid = true

;; Interaktion mit Behörde, die eID-Dienste anbietet

let usage-difficulty agency-with-eid-services.usage-difficulty

;; Aktualisiere die eID-Einstellung des Bürgers

if usage-difficulty <= 50

set citizen.eid-attitude citizen.eid-attitude + (100 - usage-difficulty)

else

set citizen.eid-attitude citizen.eid-attitude - usage-difficulty

end

;; Begrenze die eID-Einstellung auf 0-100

set citizen.eid-attitude min(max(citizen.eid-attitude, 0), 100)

end

end

;; Behörden bieten weiterhin Dienste an

; ...

end

observations

;; Beobachte die Entwicklung der eID-Einstellungen der Bürger im Laufe der Zeit

monitor plot citizen.eid-attitude

;; Analysiere den Einfluss von eID-Besitz, eID-Diensten und Nutzungs-Schwierigkeit auf die eID-Einstellungen

; ...

end

**Erläuterung:**

* **globals:** Deklariert globale Variablen, die in der gesamten Simulation verwendet werden.
* **agents:** Deklariert Agententypen, die in der Simulation verwendet werden: citizens und agencies.
* **patches:** Deklariert Patch-Typen, die die Umgebung der Simulation darstellen.
* **setup:** Initialisiert die Simulation, einschließlich der Erstellung von Agenten, der Initialisierung von Variablen und der Zuweisung von Werten zu Agenten.
* **go:** Lässt die Simulation laufen. In diesem Schritt werden die Agenten aktualisiert und ihre Interaktionen untereinander simuliert.
* **observations:** Definiert Beobachtungen, die während der Simulation aufgezeichnet werden sollen. In diesem Fall wird die Entwicklung der eID-Einstellungen der Bürger über die Zeit beobachtet.

**Wichtige Punkte:**

* Die Simulation modelliert die Interaktion zwischen Bürgern und Behörden in Bezug auf die Nutzung der eID.
* Bürger haben eine normalverteilte Einstellung zur eID und können diese Einstellung basierend auf ihren Erfahrungen mit der eID aktualisieren.
* Behörden bieten Dienste mit unterschiedlicher Nutzungs-Schwierigkeit an, was die eID-Einstellungen der Bürger beeinflusst.
* Die Simulation kann verwendet werden, um den Einfluss von Faktoren wie eID-Besitz, eID-Diensten und Nutzungs-Schwierigkeit auf die Akzeptanz der eID zu analysieren.

**Erweiterungen:**

* Implementieren Sie verschiedene Arten von e

share

more\_vert



**erstelle das programm in python**

volume\_up

**Python-Code für die eID-Simulationsmodell (mit NetLogo-Anbindung)**

**Hinweis:** Die Implementierung der eID-Simulationsmodell in Python erfordert die Verwendung der pynetlogo-Bibliothek. Diese Bibliothek ermöglicht die Interaktion zwischen Python-Code und NetLogo-Modellen.

Python

import pynetlogo as pn

# Initialisierung der Verbindung zu NetLogo-Modell

netlogo = pn.NetLogo('eid\_simulation.nlogo')

# Setzen von Parametern

initial\_eid\_proportion = 0.1

initial\_eid\_services\_proportion = 0.1

mean\_eid\_attitude = 50

stddev\_eid\_attitude = 20

# Initialisierung der Simulation in NetLogo

netlogo.command('setup')

# Senden von Parametern an NetLogo

netlogo.set\_param('initial-eid-proportion', initial\_eid\_proportion)

netlogo.set\_param('initial-eid-services-proportion', initial\_eid\_services\_proportion)

netlogo.set\_param('mean-eid-attitude', mean\_eid\_attitude)

netlogo.set\_param('stddev-eid-attitude', stddev\_eid\_attitude)

# Ausführen der Simulation in NetLogo

netlogo.run(100) # Simulieren Sie für 100 Ticks

# Abrufen von Daten aus NetLogo

eid\_attitudes = netlogo.get\_agent\_data('citizens', 'eid-attitude')

# Analyse der eID-Einstellungen

# ... (z. B. Daten visualisieren, Statistiken berechnen)

# Beenden der Verbindung zu NetLogo

netlogo.end()

Verwende den Code [mit Vorsicht](/faq#coding).

content\_copy

**Erläuterung:**

* Die pynetlogo-Bibliothek wird importiert, um die Interaktion mit NetLogo zu ermöglichen.
* Eine Verbindung zu dem NetLogo-Modell eid\_simulation.nlogo wird hergestellt.
* Die Parameter für die Simulation werden initialisiert.
* Die Simulation wird in NetLogo initialisiert und die Parameter werden an NetLogo gesendet.
* Die Simulation wird in NetLogo für 100 Ticks ausgeführt.
* Die eID-Einstellungen der Bürger werden aus NetLogo abgerufen.
* Die eID-Einstellungen werden analysiert (z. B. Datenvisualisierung, Berechnung von Statistiken).
* Die Verbindung zu NetLogo wird beendet.

**Wichtige Punkte:**

* Dieser Python-Code dient als Schnittstelle zwischen Python und dem NetLogo-Modell.
* Die eigentlichen Simulationslogik befindet sich im NetLogo-Code.
* Python kann verwendet werden, um Parameter zu setzen, Daten aus NetLogo abzurufen und die Ergebnisse der Simulation zu analysieren.

**Erweiterungen:**

* Implementieren Sie zusätzliche Funktionen in Python, um die Datenanalyse zu erweitern oder Visualisierungen zu erstellen.
* Verbinden Sie das Python-Programm mit anderen Tools oder Bibliotheken für erweiterte Analysefunktionen.
* Verwenden Sie Parameteroptimierungsverfahren in Python, um die optimalen Parameterwerte für die Simulation zu finden.

share

more\_vert



expand\_more

**Erstelle eine einfache Simulation in Python, bei der Bürger und Behörden eine eID verwenden können. Die Bürger können eine eID haben und haben zu dieser eine normalverteilte Einstellung von 0 bis 100. Anfangs haben 10% der Bürger eine eID. Die Behörden können Dienste mit eID anbieten, die Schwierigkeit der Nutzung der eID kann von 0 bis 100 gehen. Anfangs bieten 10% aller Behörden Dienste mit eID an. Die Bürger laufen umher und treffen gelegentlich auf eine Behörde. Können sie dort ihre eID nicht einsetzen, so sinkt ihre Einstellung zur eID. Können sie ihre eID einsetzen, so entwickelt sich ihre Einstellung entsprechend der Schwierigkeit der Nutzung: Ist der Einsatz leicht, so steigt ihre Einstellung, ist der Einsatz schwer, so sinkt ihre Einstellung.**

expand\_more

volume\_up

**Python-Simulation der eID-Nutzung**

Python

import numpy as np

import random

# Anzahl der Bürger und Behörden

num\_citizens = 100

num\_agencies = 20

# Initialisierung der eID-Einstellungen

initial\_eid\_proportion = 0.1

citizen\_eid\_attitudes = np.random.normal(50, 20, num\_citizens)

# Initialisierung der eID-Dienste

initial\_eid\_services\_proportion = 0.1

agency\_eid\_services = np.random.choice([True, False], size=num\_agencies, p=[initial\_eid\_services\_proportion, 1-initial\_eid\_services\_proportion])

# Simulations-Parameter

usage\_difficulty\_range = (0, 100)

interaction\_probability = 0.1

# Simulationsschleife

for \_ in range(100):

# Bürger laufen umher und interagieren mit Behörden

for citizen\_id, citizen\_attitude in enumerate(citizen\_eid\_attitudes):

if random.random() < interaction\_probability:

agency\_id = random.randint(0, num\_agencies - 1)

if agency\_eid\_services[agency\_id] and citizen\_eid\_attitudes[citizen\_id] > 0:

usage\_difficulty = random.randint(\*usage\_difficulty\_range)

citizen\_eid\_attitudes[citizen\_id] += 100 - usage\_difficulty if usage\_difficulty <= 50 else -usage\_difficulty

# Begrenzen Sie die Einstellung auf 0-100

citizen\_eid\_attitudes[citizen\_id] = max(0, min(citizen\_eid\_attitudes[citizen\_id], 100))

# Analyse der Ergebnisse

print("Durchschnittliche eID-Einstellung der Bürger:", citizen\_eid\_attitudes.mean())

Verwende den Code [mit Vorsicht](/faq#coding).

content\_copy

**Erläuterung:**

* Die Simulation verwendet numpy für numerische Berechnungen und random für Zufallszahlen.
* Die Anzahl der Bürger und Behörden wird definiert.
* Die anfänglichen eID-Einstellungen der Bürger werden normalverteilt initialisiert.
* Die Verfügbarkeit von eID-Diensten bei Behörden wird zufällig festgelegt.
* Die Simulationsschleife iteriert über 100 Zeitschritte.
* In jedem Zeitschritt laufen die Bürger umher und interagieren mit Behörden mit einer bestimmten Wahrscheinlichkeit.
* Wenn ein Bürger eine Behörde mit eID-Diensten trifft und der Bürger eine eID besitzt, wird seine Einstellung zur eID basierend auf der Schwierigkeit der Nutzung der eID aktualisiert.
* Die durchschnittliche eID-Einstellung der Bürger wird am Ende der Simulation ausgegeben.

**Wichtige Punkte:**

* Diese Simulation ist eine vereinfachte Darstellung der eID-Nutzung in einer Gesellschaft.
* Die Parameter und die Interaktionslogik können angepasst werden, um verschiedene Szenarien zu modellieren.
* Die Simulation kann mit weiteren Analysefunktionen erweitert werden, z. B. der Visualisierung der eID-Einstellungen im Laufe der Zeit.

**Erweiterungen:**

* Implementieren Sie verschiedene Arten von eID-Diensten mit unterschiedlichen Auswirkungen auf die eID-Einstellungen der Bürger.
* Modellieren Sie die Kosten für die Beschaffung und Nutzung einer eID.
* Berücksichtigen Sie die Rolle von Vertrauen und Sicherheit in der Akzeptanz der eID.
* Fügen Sie ein Modell der öffentlichen Meinung und des politischen Drucks hinzu.

share

more\_vert



**ergänze das programm um die darstellung der eID-Einstellung über die Zeit**

volume\_up

**Python-Simulation der eID-Nutzung mit Zeitverlauf**

Python

import numpy as np

import random

import matplotlib.pyplot as plt

# Anzahl der Bürger und Behörden

num\_citizens = 100

num\_agencies = 20

# Initialisierung der eID-Einstellungen

initial\_eid\_proportion = 0.1

citizen\_eid\_attitudes = np.random.normal(50, 20, num\_citizens)

# Initialisierung der eID-Dienste

initial\_eid\_services\_proportion = 0.1

agency\_eid\_services = np.random.choice([True, False], size=num\_agencies, p=[initial\_eid\_services\_proportion, 1-initial\_eid\_services\_proportion])

# Simulations-Parameter

usage\_difficulty\_range = (0, 100)

interaction\_probability = 0.1

# Speichern der eID-Einstellungen über die Zeit

citizen\_eid\_attitudes\_over\_time = []

# Simulationsschleife

for time\_step in range(100):

# Bürger laufen umher und interagieren mit Behörden

for citizen\_id, citizen\_attitude in enumerate(citizen\_eid\_attitudes):

if random.random() < interaction\_probability:

agency\_id = random.randint(0, num\_agencies - 1)

if agency\_eid\_services[agency\_id] and citizen\_eid\_attitudes[citizen\_id] > 0:

usage\_difficulty = random.randint(\*usage\_difficulty\_range)

citizen\_eid\_attitudes[citizen\_id] += 100 - usage\_difficulty if usage\_difficulty <= 50 else -usage\_difficulty

# Begrenzen Sie die Einstellung auf 0-100

citizen\_eid\_attitudes[citizen\_id] = max(0, min(citizen\_eid\_attitudes[citizen\_id], 100))

# Speichern Sie die durchschnittliche eID-Einstellung für den aktuellen Zeitschritt

citizen\_eid\_attitudes\_over\_time.append(citizen\_eid\_attitudes.mean())

# Visualisierung der eID-Einstellungen über die Zeit

plt.plot(citizen\_eid\_attitudes\_over\_time)

plt.xlabel("Zeitschritt")

plt.ylabel("Durchschnittliche eID-Einstellung")

plt.title("Entwicklung der eID-Einstellungen im Laufe der Zeit")

plt.show()

Verwende den Code [mit Vorsicht](/faq#coding).

content\_copy

**Erläuterung:**

* Die citizen\_eid\_attitudes\_over\_time-Liste wird hinzugefügt, um die durchschnittliche eID-Einstellung für jeden Zeitschritt zu speichern.
* In der Simulationsschleife wird die durchschnittliche eID-Einstellung nach jedem Zeitschritt in der Liste gespeichert.
* Nach der Simulation wird ein Liniendiagramm erstellt, um die Entwicklung der eID-Einstellungen im Laufe der Zeit zu visualisieren.

**Wichtige Punkte:**

* Diese Erweiterung ermöglicht die Visualisierung der Entwicklung der eID-Einstellungen der Bürger über die Zeit.
* Dies kann verwendet werden, um zu analysieren, wie sich die eID-Einstellungen in Abhängigkeit von Faktoren wie der Interaktionshäufigkeit, der Nutzungs-Schwierigkeit und der Verfügbarkeit von eID-Diensten ändern.
* Die Visualisierung kann auch verwendet werden, um verschiedene Simulationsszenarien miteinander zu vergleichen.

**Weitere Erweiterungen:**

* Fügen Sie verschiedene Farben oder Linienstile für verschiedene Gruppen von Bürgern (z. B. mit oder ohne eID) hinzu.
* Markieren Sie wichtige Ereignisse in der Simulation (z. B. Änderungen der Parameter) im Diagramm.
* Erstellen Sie interaktive Visualisierungen, die es Benutzern ermöglichen, die Daten zu erkunden.

share

more\_vert



edit

**schreibe das programm als agentanbasierte simulation**

edit

Vorschläge anzeigen

volume\_up

**Agentenbasierte Simulation der eID-Nutzung mit Python**

**Hinweis:** Diese Implementierung verwendet die massim-Bibliothek für die agentenbasierte Simulation. Diese Bibliothek bietet Werkzeuge zur Modellierung und Simulation komplexer Systeme mit interagierenden Agenten.

Python

import massim as sim

import numpy as np

import random

# Anzahl der Bürger und Behörden

num\_citizens = 100

num\_agencies = 20

# Initialisierung der eID-Einstellungen

initial\_eid\_proportion = 0.1

citizen\_eid\_attitudes = np.random.normal(50, 20, num\_citizens)

# Initialisierung der eID-Dienste

initial\_eid\_services\_proportion = 0.1

agency\_eid\_services = np.random.choice([True, False], size=num\_agencies, p=[initial\_eid\_services\_proportion, 1-initial\_eid\_services\_proportion])

# Simulations-Parameter

usage\_difficulty\_range = (0, 100)

interaction\_probability = 0.1

# Agentenklassen

class Citizen(sim.Agent):

def \_\_init\_\_(self, eid\_attitude):

super().\_\_init\_\_()

self.eid\_attitude = eid\_attitude

def update(self):

if random.random() < interaction\_probability:

agency = self.environment.find\_closest\_agent(Agency)

if agency is not None and agency.offers\_eid\_services and self.eid\_attitude > 0:

usage\_difficulty = random.randint(\*usage\_difficulty\_range)

self.eid\_attitude += 100 - usage\_difficulty if usage\_difficulty <= 50 else -usage\_difficulty

# Begrenzen Sie die Einstellung auf 0-100

self.eid\_attitude = max(0, min(self.eid\_attitude, 100))

class Agency(sim.Agent):

def \_\_init\_\_(self, offers\_eid\_services):

super().\_\_init\_\_()

self.offers\_eid\_services = offers\_eid\_services

# Simulationsumgebung

environment = sim.Environment(width=100, height=100)

# Erstellen und platzieren von Agenten

for \_ in range(int(num\_citizens \* initial\_eid\_proportion)):

citizen = Citizen(random.choice(citizen\_eid\_attitudes))

environment.add\_agent(citizen)

for \_ in range(num\_citizens - int(num\_citizens \* initial\_eid\_proportion)):

citizen = Citizen(0)

environment.add\_agent(citizen)

for \_ in range(int(num\_agencies \* initial\_eid\_services\_proportion)):

agency = Agency(True)

environment.add\_agent(agency)

for \_ in range(num\_agencies - int(num\_agencies \* initial\_eid\_services\_proportion)):

agency = Agency(False)

environment.add\_agent(agency)

# Simulationsschleife

for \_ in range(100):

environment.run\_all()

# Analyse der Ergebnisse

print("Durchschnittliche eID-Einstellung der Bürger:", np.mean([citizen.eid\_attitude for citizen in environment.agents if isinstance(citizen, Citizen)]))

Verwende den Code [mit Vorsicht](/faq#coding).

content\_copy

**Erläuterung:**

* Die massim-Bibliothek wird importiert.
* Die Anzahl der Bürger und Behörden wird definiert.
* Die anfänglichen eID-Einstellungen der Bürger und die Verfügbarkeit von eID-Diensten bei Behörden werden initialisiert.
* Die Simulations-Parameter werden definiert.
* Die Citizen- und Agency-Klassen werden definiert, um das Verhalten von Bürgern und Behörden zu modellieren.
* Die Simulationsumgebung wird erstellt und Agenten werden darin platziert.
* Die Simulationsschleife lässt die Agenten interagieren und aktualisiert ihre Zustände.
* Die durchschnittliche eID-Einstellung der Bürger wird am Ende der Simulation ausgegeben.

**Wichtige Punkte:**

* Diese Implementierung verwendet eine agentenbasierte Modellierungsmethode, um das Verhalten von Bürgern und Behörden in einer Welt mit eID zu simulieren.
* Die massim-Bibliothek bietet eine flexible und leistungsstarke Plattform für die Entwicklung und Analyse komplexer agentenbasierter Modelle.
* Die Simulation kann mit weiteren Funktionen erweitert werden, z. B. der Modellierung verschiedener Arten von eID-Diensten, der Berücksichtigung von Kosten und Nutzen der eID-Nutzung oder der Untersuchung des Einflusses von sozialen Normen und Meinungen.