



Synthea

Künstliche Patient*innen generieren

Was ist Synthea?



Was ist Synthea?



Synthetische Patientendaten generieren

Was ist Synthea?



Synthetische Patientendaten generieren

Simuliert dabei realistische medizinische Szenarien und Krankheitsverläufe

Was ist Synthea?



Synthetische Patientendaten generieren

Simuliert dabei realistische medizinische Szenarien und Krankheitsverläufe

Für Forschungs- und Testzwecke

Was ist Synthea?



Synthetische Patientendaten generieren

Simuliert dabei realistische medizinische Szenarien und Krankheitsverläufe

Für Forschungs- und Testzwecke

Open-Source-Projekt

Was ist Synthea?



Synthetische Patientendaten generieren

Simuliert dabei realistische medizinische Szenarien und Krankheitsverläufe

Für Forschungs- und Testzwecke

Open-Source-Projekt

Entwickler: MITRE Corporation

Wie nutzt man Synthea?



Wie nutzt man Synthea?



1. (Kostenlose Prebuilt) OpenJDK installieren <https://adoptium.net/>

Wie nutzt man Synthea?



1. (Kostenlose Prebuilt) OpenJDK installieren <https://adoptium.net/>
2. Bash/Terminal öffnen, das Git installiert hat (z. B. GitBash oder IDE Terminals)

Wie nutzt man Synthea?



1. (Kostenlose Prebuilt) OpenJDK installieren <https://adoptium.net/>
2. Bash/Terminal öffnen, das Git installiert hat (z. B. GitBash oder IDE Terminals)

Terminal

Wie nutzt man Synthea?



1. (Kostenlose Prebuilt) OpenJDK installieren <https://adoptium.net/>
2. Bash/Terminal öffnen, das Git installiert hat (z. B. GitBash oder IDE Terminals)

Terminal

```
git clone https://github.com/synthetichealth/synthea.git
```

Wie nutzt man Synthea?



1. (Kostenlose Prebuilt) OpenJDK installieren <https://adoptium.net/>
2. Bash/Terminal öffnen, das Git installiert hat (z. B. GitBash oder IDE Terminals)

Terminal

```
git clone https://github.com/synthetichealth/synthea.git  
cd synthea
```

Wie nutzt man Synthea?



1. (Kostenlose Prebuilt) OpenJDK installieren <https://adoptium.net/>
2. Bash/Terminal öffnen, das Git installiert hat (z. B. GitBash oder IDE Terminals)

Terminal

```
git clone https://github.com/synthetichealth/synthea.git  
cd synthea  
./gradlew build check test
```

Wie nutzt man Synthea?



1. (Kostenlose Prebuilt) OpenJDK installieren <https://adoptium.net/>
2. Bash/Terminal öffnen, das Git installiert hat (z. B. GitBash oder IDE Terminals)

Terminal

```
git clone https://github.com/synthetichealth/synthea.git  
cd synthea  
./gradlew build check test  
  
./run_synthea -p 100 --condition "Sepsis"
```

Wie nutzt man Synthea?



1. (Kostenlose Prebuilt) OpenJDK installieren <https://adoptium.net/>
2. Bash/Terminal öffnen, das Git installiert hat (z. B. GitBash oder IDE Terminals)

Terminal

```
git clone https://github.com/synthetichealth/synthea.git  
cd synthea  
./gradlew build check test  
  
./run_synthea -p 100 --condition "Sepsis"
```

3. Resultat: JSON Daten im FHIR Format

Output Daten im JSON Format im FHIR Standard



Output Daten im JSON Format im FHIR Standard



```
"name": [ {  
  "use": "official",  
  "family": "Halvorson124",  
  "given": [ "Cara845", "Viva686" ]  
} ],  
"telecom": [ {  
  "system": "phone",  
  "value": "555-974-6221",  
  "use": "home"  
} ],  
"gender": "female",  
"birthDate": "2012-07-11",
```

Output Daten im JSON Format im FHIR Standard



```
"name": [ {  
  "use": "official",  
  "family": "Halvorson124",  
  "given": [ "Cara845", "Viva686" ]  
} ],  
"telecom": [ {  
  "system": "phone",  
  "value": "555-974-6221",  
  "use": "home"  
} ],  
"gender": "female",  
"birthDate": "2012-07-11",
```

```
"valueCodeableConcept": {  
  "coding": [ {  
    "system": "http://snomed.info/sct",  
    "code": "266919005",  
    "display": "Never smoked tobacco (finding)"  
  } ],  
  "text": "Never smoked tobacco (finding)"  
}
```

Output Daten im JSON Format im FHIR Standard



```
"name": [ {  
  "use": "official",  
  "family": "Halvorson124",  
  "given": [ "Cara845", "Viva686" ]  
} ],  
"telecom": [ {  
  "system": "phone",  
  "value": "555-974-6221",  
  "use": "home"  
} ],  
"gender": "female",  
"birthDate": "2012-07-11",
```

```
"valueCodeableConcept": {  
  "coding": [ {  
    "system": "http://snomed.info/sct",  
    "code": "266919005",  
    "display": "Never smoked tobacco (finding)"  
  } ],  
  "text": "Never smoked tobacco (finding)"  
}
```

davon z. B.
500.000 Zeilen
für 1 Patient*in
10MB+

Wie werden die Daten under the hood generiert



Wie werden die Daten under the hood generiert

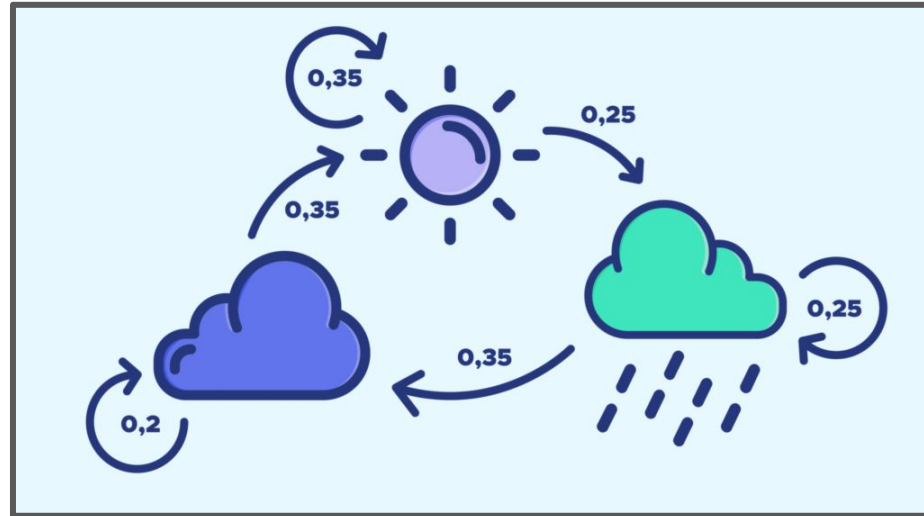


Mit Markov Modellen

Wie werden die Daten under the hood generiert



Mit Markov Modellen



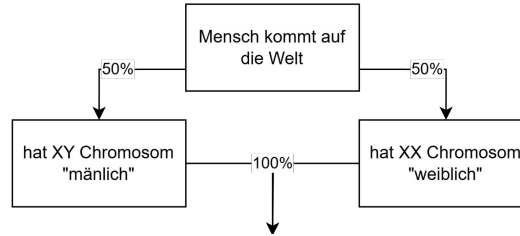
<https://datascientest.com/de/files/2023/02/markov-1-1024x562.webp>

Wie werden die Daten under the hood generiert

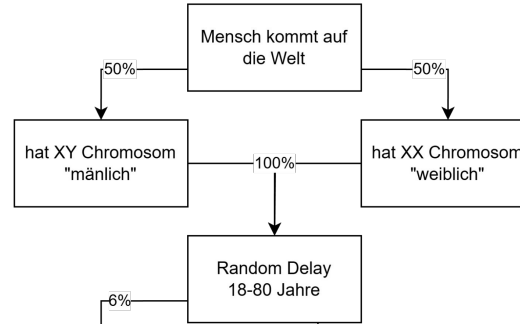


Mensch kommt auf
die Welt

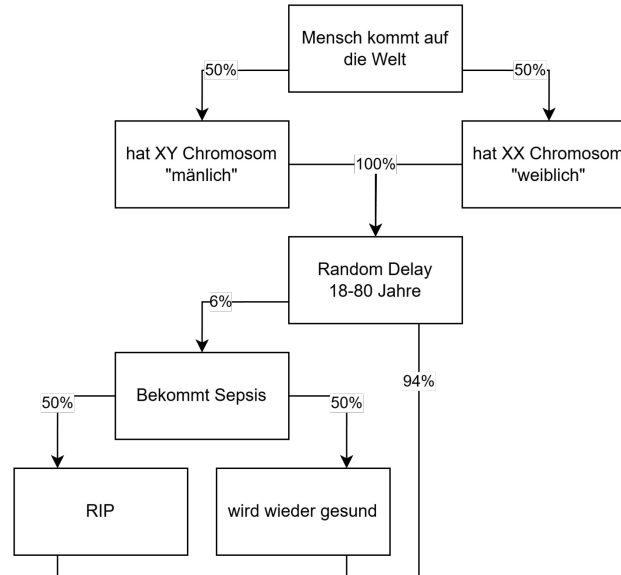
Wie werden die Daten under the hood generiert



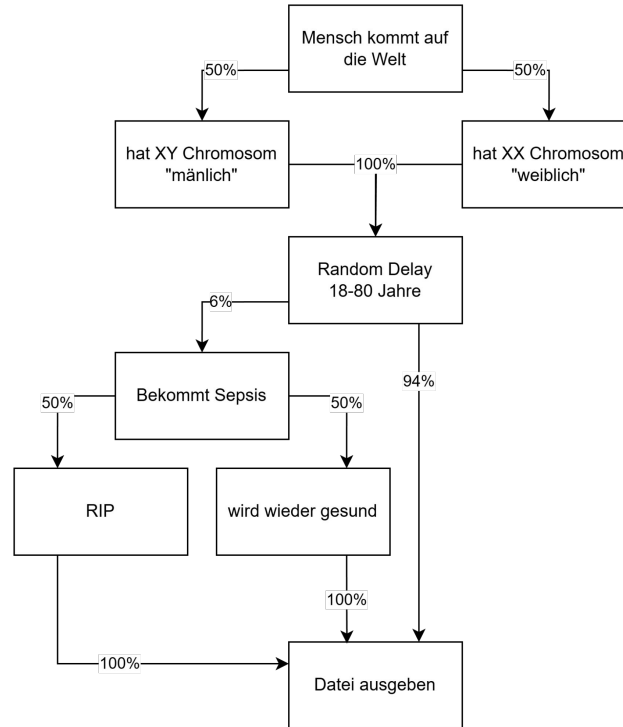
Wie werden die Daten under the hood generiert



Wie werden die Daten under the hood generiert



Wie werden die Daten under the hood generiert

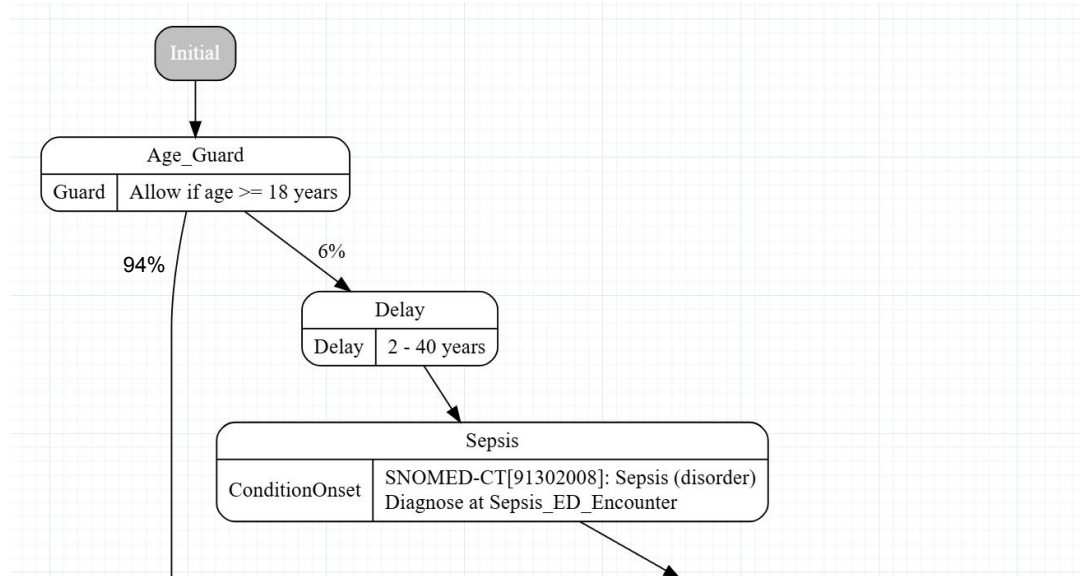


Wie kann man eigene Modelle erstellen?



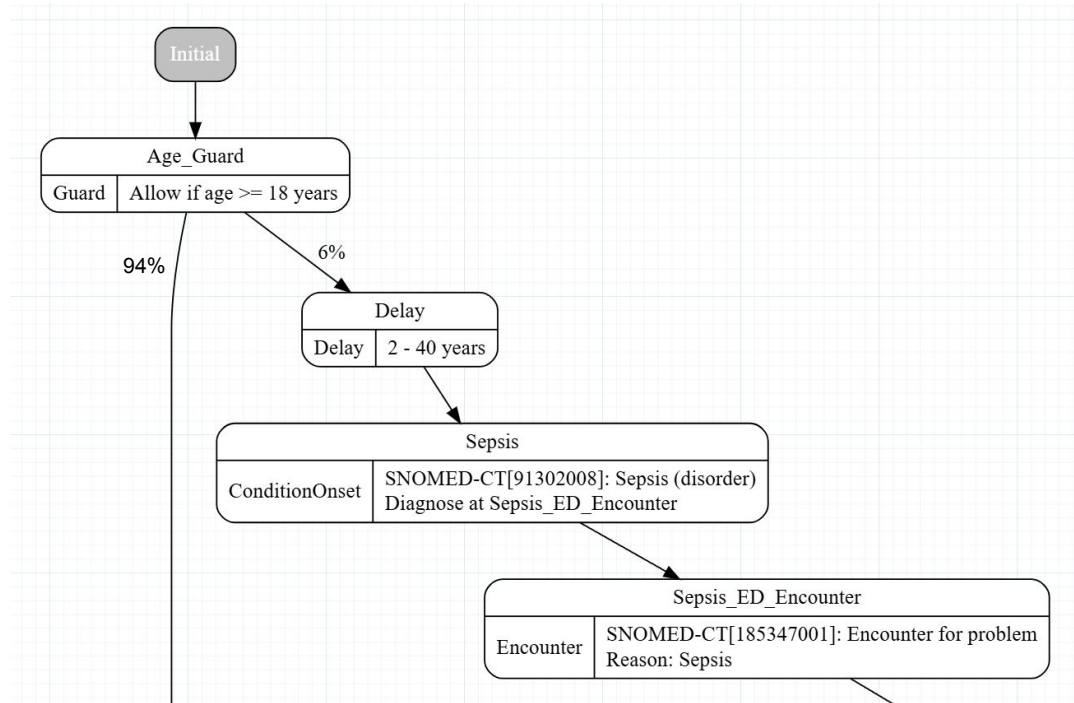
<https://synthetichealth.github.io/module-builder/#sepsis>

Wie kann man eigene Modelle erstellen?



<https://synthetichealth.github.io/module-builder/#sepsis>

Wie kann man eigene Modelle erstellen?



<https://synthetichealth.github.io/module-builder/#sepsis>

Standort ist (normalerweise nur) US



Standort ist (normalerweise nur) US



> `synthea/src/main/resources/geography/zipcodes.csv`

Standort ist (normalerweise nur) US



> `synthea/src/main/resources/geography/zipcodes.csv`

Alabama,AL,Adamsville,35005,33.59595,-87.000649	
Alabama,AL,Adamsville,35060,33.559629,-86.95346500000001	
Alabama,AL,Adamsville,35062,33.7249200000000004,-87.010252000000001	



&



Wie bekommt man die Daten in Java (Klassen)?



Wie bekommt man die Daten in Java (Klassen)?



Wie bekommt man die Daten in Java (Klassen)?



```
<dependency>  
  <groupId>com.fasterxml.jackson.core</groupId>  
  <artifactId>jackson-databind</artifactId>  
  <version>2.13.3</version>  
</dependency>
```

Wie bekommt man die Daten in Java (Klassen)?



Wie bekommt man die Daten in Java (Klassen)?



1. Eine Library für JSON Parsen (“verarbeiten”) installieren

Wie bekommt man die Daten in Java (Klassen)?



1. Eine Library für JSON Parsen (“verarbeiten”) installieren
2. Eine Klasse mit den Variablen erstellen die aus JSON ausgelesen werden

Wie bekommt man die Daten in Java (Klassen)?



1. Eine Library für JSON Parsen (“verarbeiten”) installieren
2. Eine Klasse mit den Variablen erstellen die aus JSON ausgelesen werden
3. Die JSON Daten den Variablen zuweisen

Wie bekommt man die Daten in Java (Klassen)?



```
"name": "John Doe",  
"age": 45,  
"conditions": [  
  {  
    "name": "Diabetes",  
    "date": "2023-01-01"
```

Wie bekommt man die Daten in Java (Klassen)?



```
"name": "John Doe",  
"age": 45,  
"conditions": [  
  {  
    "name": "Diabetes",  
    "date": "2023-01-01"
```

```
public class Patient {  
    private String name;  
    private int age;  
    private List<Condition> conditions;  
  
    // Getter und Setter  
}  
  
public class Condition {  
    private String name;  
    private String date;  
  
    // Getter und Setter  
}
```

Wie bekommt man die Daten in Java (Klassen)?



Wie bekommt man die Daten in Java (Klassen)?



```
ObjectMapper mapper = new ObjectMapper();

// JSON-Datei laden und in Patient-Objekt konvertieren
Patient patient = mapper.readValue(new File("path/to/synthea-data.json"), Patient.class);

// Daten anzeigen
System.out.println("Name: " + patient.getName());
System.out.println("Alter: " + patient.getAge());
System.out.println("Bedingungen:");
for (Condition condition : patient.getConditions()) {
    System.out.println(" - " + condition.getName() + " (Datum: " + condition.getDate() + ")");
}
```

Wie könnte man die Daten anzeigen?



Wie könnte man die Daten anzeigen?



```
public static void showPatientData(List<Patient> patients) {  
    // Spaltennamen definieren  
    String[] columnNames = {"Name", "Alter", "Bedingung", "Datum"};  
  
    // Daten in die Tabelle einfügen  
    DefaultTableModel model = new DefaultTableModel(columnNames, 0);  
    for (Patient patient : patients) {  
        for (Condition condition : patient.getConditions()) {  
            model.addRow(new Object[]{patient.getName(), patient.getAge(), condition.getName(), condition.getDate()});  
        }  
    }  
  
    // Tabelle erstellen und anzeigen  
    JTable table = new JTable(model);  
    JScrollPane scrollPane = new JScrollPane(table);  
}
```



```

import javax.swing.*;
import javax.swing.table.DefaultTableModel;
import java.util.List;

public class TableVisualizer {

    public static void showPatientData(List<Patient> patients) {
        // Spaltennamen definieren
        String[] columnNames = {"Name", "Alter", "Bedingung", "Datum"};

        // Daten in die Tabelle einfügen
        DefaultTableModel model = new DefaultTableModel(columnNames, 0);
        for (Patient patient : patients) {
            for (Condition condition : patient.getConditions()) {
                model.addRow(new Object[]{patient.getName(), patient.getAge(), condition.getName(), condition.getDate()});
            }
        }

        // Tabelle erstellen und anzeigen
        JTable table = new JTable(model);
        JScrollPane scrollPane = new JScrollPane(table);

        JFrame frame = new JFrame("Patientendaten");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.add(scrollPane);
        frame.pack();
        frame.setVisible(true);
    }
}

```



```

import javax.swing.*;
import javax.swing.table.DefaultTableModel;
import java.util.List;

public class TableVisualizer {

    public static void showPatientData(List<Patient> patients) {
        // Spaltennamen definieren
        String[] columnNames = {"Name", "Alter", "Bedingung", "Datum"};

        // Daten in die Tabelle einfügen
        DefaultTableModel model = new DefaultTableModel(columnNames, 0);
        for (Patient patient : patients) {
            for (Condition condition : patient.getConditions()) {
                model.addRow(new Object[]{patient.getName(), patient.getAge(), condition.getName(), condition.getDate()});
            }
        }

        // Tabelle erstellen und anzeigen
        JTable table = new JTable(model);
        JScrollPane scrollPane = new JScrollPane(table);

        JFrame frame = new JFrame("Patientendaten");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.add(scrollPane);
        frame.pack();
        frame.setVisible(true);
    }
}

```

```
show(pd.read_json("path/to/synthea-data.json"))
```



Fazit

Eigene
Meinung

Fazit

Eigene
Meinung

Synthea ist für Stresstests und Simulationen im großen Maßstab geeignet

Fazit

Eigene
Meinung

Synthea ist für Stresstests und Simulationen im großen Maßstab geeignet

Für kleine Anwendungen ist es eigentlich overkill, denn es wäre einfacher nur die Sepsis relevanten Werte selbst zu generieren

Fazit

Eigene
Meinung

Synthesa ist für Stresstests und Simulationen im großen Maßstab geeignet

Für kleine Anwendungen ist es eigentlich overkill, denn es wäre einfacher nur die Sepsis relevanten Werte selbst zu generieren

Trotzdem ist es sinnvoll die Technologie zu beherrschen, da sie prinzipiell für jedes Projekt, welches viele Personen verwaltet, interessant sein kann