



Fachbereich Elektrotechnik und Informatik
Erweiterte Programmiersprachen
SS24

*„Prognose der von aussterben bedrohten Bevölkerung
Chinas im Jahr 2100 mit Julia“*

zur Erlangung des Grades
Bachelor of Science

Vorgelegt von Emre Akarsu

Matrikelnummer: 018345607

Emre.akarsu@stud.hs-bochum.de

Studiengang: Bachelor Informatik

Fachsemester: 6

Abzugeben an: Prof. Dr.-Ing. Edmund Coersmeier

Abgabedatum: 4. Juni 2024

Inhaltsverzeichnis

| | |
|---|-----------|
| Abbildungsverzeichnis | i |
| 1 Einleitung | 1 |
| 1.1 Problemstellung | 1 |
| 1.2 Methodik und Strategie | 2 |
| 2 Technische Grundlagen | 2 |
| 2.1 Julia | 2 |
| 2.2 Pakete und Bibliotheken | 2 |
| 3 Implementierung in Julia | 3 |
| 3.1 Abrufen der Daten | 3 |
| 3.2 Extrahieren der Daten | 5 |
| 3.3 Berechnungen und Prognosen | 6 |
| 3.4 Visualisierung der Ergebnisse | 7 |
| 4 Ergebnisse | 8 |
| 4.1 Ergebnisse der Analyse | 8 |
| 4.1.1 Geburten- und Sterberaten | 8 |
| 4.1.2 Jährliche Wachstumsrate | 8 |
| 4.1.3 Bevölkerungsprognose | 9 |
| 4.2 Fazit | 10 |
| 5 Anhang | 10 |

Abbildungsverzeichnis

| | | |
|---|---|---|
| 1 | Geburtenrate vs. Sterberate in China | 8 |
| 2 | Jährliche Wachstumsrate von China (1960-2022) | 9 |
| 3 | Bevölkerungsprognose für China (1960-2100) | 9 |

Quellcodeverzeichnis

| | | |
|---|--|---|
| 1 | Abrufen der Daten | 3 |
| 2 | Parameter für die Datenabfrage | 4 |
| 3 | Extrahieren der Daten | 5 |
| 4 | Berechnungen und Prognosen | 6 |
| 5 | Visualisierung der Ergebnisse | 7 |

1 Einleitung

China ist das bevölkerungsreichste Land der Welt und spielt eine zentrale Rolle in der globalen Demografie. In den letzten Jahrzehnten hat das Land jedoch eine signifikante demografische Verschiebung erlebt, die durch eine alternde Bevölkerung und eine niedrige Geburtenrate gekennzeichnet ist. Die ein Kind Politik, die von 1979 bis 2015 in China in Kraft war, hat zu einem Ungleichgewicht zwischen den Geschlechtern und einer alternden Bevölkerung geführt. Lebenshaltungskosten und wirtschaftliche Unsicherheit haben auch dazu beigetragen, dass viele Paare sich gegen Kinder entscheiden. Diese demografischen Herausforderungen könnten langfristige Auswirkungen auf die wirtschaftliche und soziale Entwicklung Chinas haben.

Unter Berücksichtigung dieser Aspekte ist es von entscheidender Bedeutung, die demografische Entwicklung Chinas zu analysieren und Prognosen für die Zukunft zu erstellen.

Dieses Projekt zielt darauf ab, die demografische Entwicklung Chinas bis zum Jahr 2100 zu analysieren und zu modellieren. Dazu verwenden wir die Programmiersprache Julia, die sich durch ihre Leistungsfähigkeit und Effizienz bei der Verarbeitung großer Datenmengen auszeichnet.

1.1 Problemstellung

China steht vor einer demografischen Herausforderung, die tiefgreifende Auswirkungen auf die wirtschaftliche und soziale Struktur des Landes hat. Die Hauptprobleme sind:

1. Sinkende Geburtenrate: Die Geburtenrate in China ist seit Jahrzehnten rückläufig und liegt mittlerweile weit unter dem für den Bevölkerungersatz notwendigen Niveau.
2. Alternde Bevölkerung: Der Anteil älterer Menschen in der chinesischen Gesellschaft nimmt stetig zu, was zu einer erhöhten Belastung des Rentensystems und des Gesundheitswesens führt.
3. Ein-Kind-Politik: Obwohl die Ein-Kind-Politik in China offiziell abgeschafft wurde, hat sie langfristige Auswirkungen auf die Bevölkerungsstruktur des Landes.
4. Zu hohe Lebenshaltungskosten: Das verfügbare Einkommen und die Lebenshaltungskosten haben einen direkten Einfluss auf die Familienplanung und die Geburtenrate in China.

Diese Probleme führen zu einem erwarteten Bevölkerungsrückgang, der die wirtschaftliche Stabilität und das soziale Wohlergehen Chinas bedrohen könnte.

1.2 Methodik und Strategie

Die für diese Hausarbeit verwendeten Daten stammen aus der WorldPop Datenbank, welches eine umfassende Sammlung von Bevölkerungsdaten aus verschiedenen Quellen enthält. Diese Datenbank bietet eine API, die es ermöglicht, Bevölkerungsdaten für verschiedene Regionen und Zeiträume abzurufen.

Anhand dieser Datenquellen können wir die demografische Entwicklung Chinas in den letzten Jahrzehnten analysieren und Prognosen für die Zukunft erstellen. Diese Analyse wird folgende Punkte umfassen:

- Das Aufstellen der Geburten- und Sterberaten der letzten 60 Jahre
- Die Berechnung der Wachstumsrate der Bevölkerung
- Prognosen für die Bevölkerungsentwicklung bis 2100

Julia eignet sich in unserem Fall besonders gut, zur Bearbeitung und Analyse großer Datenmengen. Die Programmiersprache bietet eine einfache Syntax und eine umfangreiche Sammlung von Bibliotheken und Paketen, die für die Datenverarbeitung und Modellierung benötigt werden.

2 Technische Grundlagen

Dieses Kapitel beschreibt die technischen Grundlagen und Werkzeuge, die für die Modellierung und Analyse der demografischen Entwicklung Chinas verwendet werden. Dazu gehören die Programmiersprache Julia und die verwendeten Pakete und Bibliotheken.

2.1 Julia

Julia ist eine Hochleistungsprogrammiersprache, die für numerische und wissenschaftliche Berechnungen entwickelt wurde. Sie zeichnet sich durch ihre Geschwindigkeit und Effizienz aus und wird häufig für die Verarbeitung großer Datenmengen verwendet. Julia ist eine Open-Source-Programmiersprache, die auf einer einfachen und intuitiven Syntax basiert. Julia wird häufig in den Bereichen Data Science, Machine Learning und wissenschaftliche Forschung eingesetzt.

In diesem Projekt verwenden wir Julia, um die demografische Entwicklung Chinas zu modellieren und zu analysieren.

2.2 Pakete und Bibliotheken

Die folgenden Julia-Pakete wurden verwendet, um die Daten abzurufen und zu verarbeiten:

- HTTP - Wird verwendet, um HTTP-Anfragen zu senden und Antworten zu empfangen.

- JSON - Wird verwendet, um JSON-Daten zu verarbeiten.
- Plots - Wird verwendet, um Diagramme und Grafiken zu erstellen.

3 Implementierung in Julia

In diesem Abschnitt wird die Implementierung des Modells in Julia beschrieben.

3.1 Abrufen der Daten

Für die Analyse der demografischen Entwicklung Chinas werden Daten zu Geburten-, Sterberaten und Bevölkerung benötigt. Diese Daten können hier mit der Worldbank-API abgerufen werden.

```
1  # Funktion zum Abrufen der Daten für ein bestimmtes Jahr und Indikator
2  function get_indicator_data(country_code::String, indicator::String, start_year::Int,
   ↪  end_year::Int)
3      # URL für den API-Aufruf zur Abrufung der Daten von der Weltbank
4      url = "http://api.worldbank.org/v2/country/$country_code/indicator/"
5      url *= "$indicator?date=$start_year:$end_year&format=json&per_page=1000"
6
7      # HTTP-GET-Anfrage an die URL
8      response = HTTP.get(url)
9      # Überprüfen, ob die Anfrage erfolgreich war
10     if response.status == 200
11         # JSON-Daten aus der Antwort parsen
12         data = JSON.parse(String(response.body))
13         return data
14     else
15         # Fehlermeldung ausgeben, wenn die Anfrage nicht erfolgreich war
16         println("Fehler beim Abrufen der Daten für den Indikator $indicator:
   ↪  HTTP-Statuscode ", response.status)
17         return nothing
18     end
19 end
20
```

Auflistung 1: Abrufen der Daten

Die Funktion `get_indicator_data` bekommt 4 Parameter zugewiesen die zum Abrufen der Daten benötigt werden. Die Funktion gibt ein JSON-Objekt zurück, das die abgerufenen Daten enthält. Wenn ein Fehler auftritt, wird eine entsprechende Fehlermeldung zurückgegeben. Das Abrufen der Daten erfolgt über eine HTTP-GET-Anfrage. Die Funktion nimmt die folgenden Argumente:

- `country_code::String`: Der Ländercode des Landes, für das die Daten abgerufen werden sollen.
- `indicator::String`: Der Indikatorcode des Indikators, für den die Daten abgerufen werden sollen.
- `start_year::Int`: Das Startjahr des Zeitraums, für den die Daten abgerufen werden sollen.
- `end_year::Int`: Das Endjahr des Zeitraums, für den die Daten abgerufen werden sollen.

Die Funktion gibt ein JSON-Objekt zurück, das die abgerufenen Daten enthält. Wenn die `get` Anfrage erfolgreich war gibt diese den Statuscode 200 zurück. Das JSON-Objekt wird im Anschluss in ein String umgewandelt und schließlich als Dictionary-Objekt gespeichert. Wenn ein Fehler auftritt, wird eine entsprechende Fehlermeldung zurückgegeben.

```
1  # Abrufen der Geburten-, Sterberaten und Bevölkerung von China für die Jahre 1960 bis 2022
2  country_code = "CHN" # Ländercode für China
3  start_year = 1960 # Startjahr
4  end_year = 2022 # Endjahr
5
6  # Indikatoren für Geburten-, Sterberaten und Bevölkerung
7  birth_rate_indicator = "SP.DYN.CBRT.IN" # Indikator für die Geburtenrate
8  death_rate_indicator = "SP.DYN.CDRT.IN" # Indikator für die Sterberate
9  population_indicator = "SP.POP.TOTL" # Indikator für die Gesamtbevölkerung
10
11 # Abrufen der Daten für die jeweiligen Indikatoren
12 birth_data = get_indicator_data(country_code, birth_rate_indicator, start_year, end_year)
13 death_data = get_indicator_data(country_code, death_rate_indicator, start_year, end_year)
14 population_data = get_indicator_data(country_code, population_indicator, start_year,
    ↪ end_year)
```

Auflistung 2: Parameter für die Datenabfrage

Den Ländercode den wir für die Abfrage brauchen, lautet nach ISO-3166-1 Alpha-3 für China `CHN`. Die Stammdaten für die Abfrage sind von 1960 bis 2022. Die Indikatoren für Geburtenrate, Sterberate und Bevölkerung sind `SP.DYN.CBRT.IN`, `SP.DYN.CDRT.IN` und `SP.POP.TOTL`, welches von der Worldbank-API dokumentiert wurde. Die Daten werden im Anschluss mit der Funktion `get_indicator_data` abgerufen.

3.2 Extrahieren der Daten

```
1  # Extrahieren der Daten
2  years = start_year:end_year # Jahre von 1960 bis 2022
3  birth_rates = [NaN for _ in years] # Initialisiere Geburtenraten mit NaN
4  death_rates = [NaN for _ in years] # Initialisiere Sterberaten mit NaN
5  historical_population = [NaN for _ in years] # Initialisiere historische Bevölkerung mit
   ↪ NaN
6  println(birth_data)
7  # Extrahieren der Geburtenraten
8  for record in birth_data[2]
9      year = parse{Int, record["date"]} # Jahr aus dem Datensatz parsen
10     if year in years
11         birth_rates[year - start_year + 1] = record["value"] # Geburtenrate für das
   ↪ entsprechende Jahr speichern
12     end
13 end
14
15 # Extrahieren der Sterberaten
16 for record in death_data[2]
17     year = parse{Int, record["date"]} # Jahr aus dem Datensatz parsen
18     if year in years
19         death_rates[year - start_year + 1] = record["value"] # Sterberate für das
   ↪ entsprechende Jahr speichern
20     end
21 end
22
23 # Extrahieren der Bevölkerungsdaten
24 for record in population_data[2]
25     year = parse{Int, record["date"]} # Jahr aus dem Datensatz parsen
26     if year in years
27         historical_population[year - start_year + 1] = record["value"] # Bevölkerungszahl
   ↪ für das entsprechende Jahr speichern
28     end
29 end
```

Auflistung 3: Extrahieren der Daten

Nachdem die Daten abgerufen wurden, werden sie extrahiert und in entsprechenden Arrays gespeichert. Dafür Initialisieren wir Arrays für die Geburtenraten, Sterberaten und historische Bevölkerungszahlen. Die jeweiligen Daten werden dann aus dem Dictionary extrahiert und in die entsprechenden Arrays gespeichert. Jeder Index im Array entspricht einem Jahr von 1960 bis 2022. Die Arrays werden dann für die weitere Analyse und Modellierung verwendet.

3.3 Berechnungen und Prognosen

```

1  # Berechnung der jährlichen Wachstumsrate in Prozent
2  function calculate_growth_rate(birth_rates::Vector{Float64}, death_rates::Vector{Float64})
3      growth_rates = [NaN for _ in birth_rates] # Initialisiere Wachstumsraten mit NaN
4      for i in 1:length(birth_rates)
5          if !isnan(birth_rates[i]) && !isnan(death_rates[i]) # Überprüfen, ob Geburten- und
6              ↪ Sterberate gültige Werte haben
7              growth_rates[i] = (birth_rates[i] - death_rates[i]) / 10 # Berechnung der
8              ↪ Wachstumsrate in Prozent
9          end
10     end
11     return growth_rates
12 end
13
14 growth_rates = calculate_growth_rate(birth_rates, death_rates) # Berechne die jährlichen
15 ↪ Wachstumsraten
16
17 # Prognose der zukünftigen Wachstumsraten mit Annahme einer sinkenden Bevölkerung
18 function extend_growth_rates(growth_rates::Vector{Float64}, future_years::Int)
19     # Nehmen wir an, dass die Wachstumsrate jedes Jahr leicht negativ wird, um eine
20     ↪ moderate Abnahme der Bevölkerung zu erreichen
21     extended_growth_rates = vcat(growth_rates, [0.1 - i*0.03 for i in 1:future_years])
22     return extended_growth_rates
23 end
24
25 # Funktion zur Bevölkerungsprognose
26 function forecast_population(start_population::Float64, growth_rates::Vector{Float64},
27     ↪ future_years::Int)
28     future_population = [start_population] # Initialisiere zukünftige Bevölkerung mit der
29     ↪ Startpopulation
30     for i in 1:future_years
31         # Berechnung der neuen Bevölkerungszahl basierend auf der Wachstumsrate
32         new_population = future_population[end] * (1 + growth_rates[length(growth_rates) -
33             ↪ future_years + i] / 100)
34         push!(future_population, new_population) # Neue Bevölkerungszahl zur Liste
35         ↪ hinzufügen
36     end
37     return future_population
38 end
39
40 start_population = historical_population[end] # Startpopulation ist die letzte bekannte
41 ↪ Bevölkerungszahl
42 future_years = 2022:2100 # Zukunftsjahre von 2022 bis 2100
43 extended_growth_rates = extend_growth_rates(growth_rates, length(future_years)) #
44 ↪ Erweitere die Wachstumsraten um die Zukunftsjahre
45 future_population = forecast_population(start_population, extended_growth_rates,
46     ↪ length(future_years)) # Prognostiziere die zukünftige Bevölkerung

```

Nachdem wir die notwendigen Daten zwischengespeichert haben, geht es nun darum, diese Daten zu verarbeiten. Dafür gibt es drei Funktionen. Die erste Funktion berechnet die jährliche Wachstumsrate in Prozent. Die zweite Funktion erweitert die Wachstumsraten für zukünftige Jahre, wobei angenommen wird, dass die Wachstumsrate jedes Jahr leicht negativ wird, da nämlich die aktuelle Sterberate die Geburtenrate übersteigt und somit für eine Verminderung der Bevölkerung aufweist. Die dritte Funktion prognostiziert die zukünftige Bevölkerungszahl basierend auf der Startbevölkerung und den erweiterten Wachstumsraten. Die Prognose wird bis zum Jahr 2100 durchgeführt.

3.4 Visualisierung der Ergebnisse

Hier werden die Ergebnisse visualisiert, um die demografische Entwicklung Chinas besser zu verstehen.

```
1
2
3  # Plot der jährlichen Wachstumsrate in Prozent
4  plot(years, growth_rates, label="Jährliche Wachstumsrate", xlabel="Jahr",
5  ↪  ylabel="Wachstumsrate (%)", title="Jährliche Wachstumsrate von China (1960-2022)",
6  ↪  lw=2, color=:blue)
7  savefig("jaehrliche_wachstumsrate_china_1960_2022.png") # Speichere den Plot der
8  ↪  jährlichen Wachstumsrate
9
10 # Plot der historischen und prognostizierten Bevölkerung
11
12 plot(years, historical_population, label="Historische Bevölkerung", xlabel="Jahr",
13 ↪  ylabel="Bevölkerung", title="Bevölkerungsprognose für China", lw=2)
14 plot!(future_years, future_population[2:end], label="Prognostizierte Bevölkerung", lw=2,
15 ↪  color=:red)
16 savefig("bevoelkerungsprognose_china.png") # Speichere den Plot der Bevölkerungsprognose
17
18 # Weitere Visualisierung: Geburtenrate vs. Sterberate
19 plot(years, birth_rates, label="Geburtenrate", xlabel="Jahr", ylabel="Rate (pro 1000
20 ↪  Personen)", title="Geburtenrate vs. Sterberate in China", lw=2)
21 plot!(years, death_rates, label="Sterberate", lw=2, color=:red)
22 savefig("geburtenrate_vs_sterberate_china.png") # Speichere den Plot der Geburten- und
23 ↪  Sterberaten
```

Auflistung 5: Visualisierung der Ergebnisse

Es werden drei Plots erstellt: einer für die jährliche Wachstumsrate, einer für die historische und prognostizierte Bevölkerung und einer für die Geburten- und Sterberaten. Es ist wichtig zu beachten, dass diese Plots nur eine grobe Vorstellung der demografischen Entwicklung Chinas geben und nicht als genaue Prognosen betrachtet werden sollten. Dafür geben wir in der Plot-funktion die errechneten Daten an und die jeweiligen Achsenbeschriftungen, sowie noch einige

andere Attribute, die den Plot verschönern sollen. Die Plots werden im Anschluss als PNG-Dateien gespeichert.

4 Ergebnisse

In diesem Kapitel werden die Ergebnisse der Arbeit präsentiert.

4.1 Ergebnisse der Analyse

Die aus der API extrahierten Daten wurden verwendet, um 3 Diagramme zu erstellen. Diese sind notwendig um die demografische Entwicklung Chinas besser zu verstehen.

4.1.1 Geburten- und Sterberaten

Um einen Überblick über Chinas aktuellen demografische Entwicklung zu bekommen, muss man zunächst einmal die Geburten- und Sterberaten betrachten. Das folgende Diagramm zeigt die Geburten- und Sterberaten in China von 1960 bis 2022.

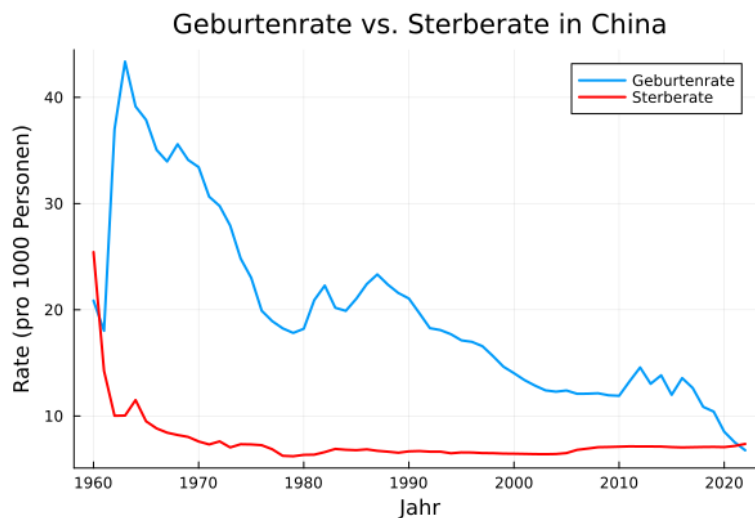


Abbildung 1: Geburtenrate vs. Sterberate in China

Man erkennt deutlich, dass die Geburtenrate in China in den letzten Jahrzehnten deutlich gesunken ist, während die Sterberate relativ stabil geblieben ist. Dies hat zu einem Rückgang des natürlichen Bevölkerungswachstums geführt. Viele Politische sowohl als auch wirtschaftliche Faktoren haben dazu beigetragen, dass die Geburtenrate in China gesunken ist. Die Ein-Kind-Politik, die seit 2015 aufgehoben wurde, zeigt hier keinen direkten Einfluss auf die Geburtenrate.

4.1.2 Jährliche Wachstumsrate

Mit den Geburten- und Sterberaten können wir die jährliche Wachstumsrate berechnen. Die jährliche Wachstumsrate gibt an, wie schnell die Bevölkerung wächst oder schrumpft. Das folgende Diagramm zeigt die jährliche Wachstumsrate in China von 1960 bis 2022.

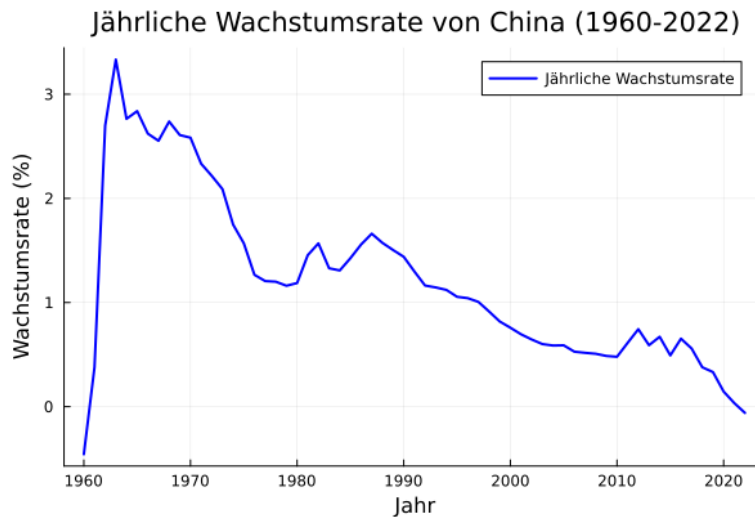


Abbildung 2: Jährliche Wachstumsrate von China (1960-2022)

Deutlich zu erkennen ist, dass die jährliche Wachstumsrate in China in den letzten Jahrzehnten gesunken ist. Die sinkende Geburtenrate und die stagnierende Sterberate haben zu einem negativen Bevölkerungswachstum geführt. Im Jahr 2022 scheint die Wachstumsrate negativen Raum fortzulaufen, was auf eine schrumpfende Bevölkerung hindeutet. Mit diesen Daten können wir nun die zukünftige Bevölkerung Chinas prognostizieren.

4.1.3 Bevölkerungsprognose

Basierend auf den historischen Daten und den prognostizierten Wachstumsraten können wir die zukünftige Bevölkerung Chinas prognostizieren. Das folgende Diagramm zeigt die historische und prognostizierte Bevölkerung Chinas von 1960 bis 2100.

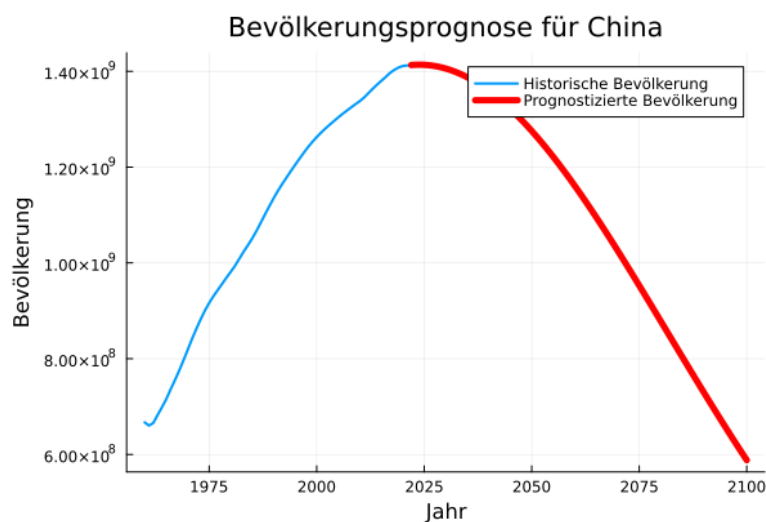


Abbildung 3: Bevölkerungsprognose für China (1960-2100)

China ein Land mit knapp 1,4 Milliarden Einwohnern, wird in den nächsten Jahrzehnten wahrscheinlich eine schrumpfende Bevölkerung haben. Die Prognose zeigt, dass die Bevölkerung

Chinas bis zum Jahr 2100 auf unter 1 Milliarde sinken wird. Dies hat weitreichende Auswirkungen auf die Wirtschaft, das Gesundheitswesen und die Gesellschaft Chinas. Die sinkende Bevölkerungszahl wird zu einem Rückgang der Arbeitskräfte und einem Anstieg der Altersbevölkerung führen. Dies wird die sozialen Sicherungssysteme und die Wirtschaft des Landes belasten. Die Ergebnisse der Analyse zeigen klar, dass China vor großen demografischen Herausforderungen steht, die in den kommenden Jahren bewältigt werden müssen.

4.2 Fazit

Ungeachtet der politischen und wirtschaftlichen Faktoren, die die demografische Entwicklung Chinas beeinflussen, ist es wichtig, die Daten und Fakten zu verstehen, um fundierte Entscheidungen zu treffen. Die Migration und andere Faktoren wurden bei dieser Analyse nicht berücksichtigt, da sie einen geringen Einfluss auf die demografische Entwicklung Chinas haben und diese Analyse nur eine grobe Vorstellung der demografischen Entwicklung Chinas gibt. In einem Land wie China, das eine lange Geschichte und eine komplexe Gesellschaft hat, sind die demografischen Herausforderungen vielfältig und erfordern eine umfassende Strategie zur Bewältigung. Die Ergebnisse dieser Analyse können als Ausgangspunkt für weitere Untersuchungen und Diskussionen dienen, um die demografische Entwicklung Chinas besser zu verstehen und angemessene Maßnahmen zu ergreifen.

5 Anhang

```
1  using HTTP
2  using JSON
3  using Plots
4
5
6  # Funktion zum Abrufen der Daten für ein bestimmtes Jahr und Indikator
7  function get_indicator_data(country_code::String, indicator::String, start_year::Int,
8  ↪   end_year::Int)
9      # URL für den API-Aufruf zur Abrufung der Daten von der Weltbank
10     url = "http://api.worldbank.org/v2/country/$country_code/indicator/"
11     url *= "$indicator?date=$start_year:$end_year&format=json&per_page=1000"
12
13     # HTTP-GET-Anfrage an die URL
14     response = HTTP.get(url)
15     # Überprüfen, ob die Anfrage erfolgreich war
16     if response.status == 200
17         # JSON-Daten aus der Antwort parsen
18         data = JSON.parse(String(response.body))
19         return data
20     else
21         # Fehlermeldung ausgeben, wenn die Anfrage nicht erfolgreich war
```

```
21     println("Fehler beim Abrufen der Daten für den Indikator $indicator:
    ↳ HTTP-Statuscode ", response.status)
22     return nothing
23 end
24 end
25
26 # Abrufen der Geburten-, Sterberaten und Bevölkerung von China für die Jahre 1960 bis 2022
27 country_code = "CHN" # Ländercode für China
28 start_year = 1960 # Startjahr
29 end_year = 2022 # Endjahr
30
31 # Indikatoren für Geburten-, Sterberaten und Bevölkerung
32 birth_rate_indicator = "SP.DYN.CBRT.IN" # Indikator für die Geburtenrate
33 death_rate_indicator = "SP.DYN.CDRT.IN" # Indikator für die Sterberate
34 population_indicator = "SP.POP.TOTL" # Indikator für die Gesamtbevölkerung
35
36 # Abrufen der Daten für die jeweiligen Indikatoren
37 birth_data = get_indicator_data(country_code, birth_rate_indicator, start_year, end_year)
38 death_data = get_indicator_data(country_code, death_rate_indicator, start_year, end_year)
39 population_data = get_indicator_data(country_code, population_indicator, start_year,
    ↳ end_year)
40
41 # Extrahieren der Daten
42 years = start_year:end_year # Jahre von 1960 bis 2022
43 birth_rates = [NaN for _ in years] # Initialisiere Geburtenraten mit NaN
44 death_rates = [NaN for _ in years] # Initialisiere Sterberaten mit NaN
45 historical_population = [NaN for _ in years] # Initialisiere historische Bevölkerung mit
    ↳ NaN
46 println(birth_data)
47 # Extrahieren der Geburtenraten
48 for record in birth_data[2]
49     year = parse{Int, record["date"]} # Jahr aus dem Datensatz parsen
50     if year in years
51         birth_rates[year - start_year + 1] = record["value"] # Geburtenrate für das
            ↳ entsprechende Jahr speichern
52     end
53 end
54
55 # Extrahieren der Sterberaten
56 for record in death_data[2]
57     year = parse{Int, record["date"]} # Jahr aus dem Datensatz parsen
58     if year in years
59         death_rates[year - start_year + 1] = record["value"] # Sterberate für das
            ↳ entsprechende Jahr speichern
60     end
61 end
62
63 # Extrahieren der Bevölkerungsdaten
64 for record in population_data[2]
65     year = parse{Int, record["date"]} # Jahr aus dem Datensatz parsen
```

```

66     if year in years
67         historical_population[year - start_year + 1] = record["value"] # Bevölkerungszahl
        ↪ für das entsprechende Jahr speichern
68     end
69 end
70
71 # Berechnung der jährlichen Wachstumsrate in Prozent
72 function calculate_growth_rate(birth_rates::Vector{Float64}, death_rates::Vector{Float64})
73     growth_rates = [NaN for _ in birth_rates] # Initialisiere Wachstumsraten mit NaN
74     for i in 1:length(birth_rates)
75         if !isnan(birth_rates[i]) && !isnan(death_rates[i]) # Überprüfen, ob Geburten- und
        ↪ Sterberate gültige Werte haben
76             growth_rates[i] = (birth_rates[i] - death_rates[i]) / 10 # Berechnung der
            ↪ Wachstumsrate in Prozent
77         end
78     end
79     return growth_rates
80 end
81
82 growth_rates = calculate_growth_rate(birth_rates, death_rates) # Berechne die jährlichen
    ↪ Wachstumsraten
83
84 # Prognose der zukünftigen Wachstumsraten mit Annahme einer sinkenden Bevölkerung
85 function extend_growth_rates(growth_rates::Vector{Float64}, future_years::Int)
86     # Nehmen wir an, dass die Wachstumsrate jedes Jahr leicht negativ wird, um eine
    ↪ moderate Abnahme der Bevölkerung zu erreichen
87     extended_growth_rates = vcat(growth_rates, [0.1 - i*0.03 for i in 1:future_years])
88     return extended_growth_rates
89 end
90
91 # Funktion zur Bevölkerungsprognose
92 function forecast_population(start_population::Float64, growth_rates::Vector{Float64},
    ↪ future_years::Int)
93     future_population = [start_population] # Initialisiere zukünftige Bevölkerung mit der
    ↪ Startpopulation
94     for i in 1:future_years
95         # Berechnung der neuen Bevölkerungszahl basierend auf der Wachstumsrate
96         new_population = future_population[end] * (1 + growth_rates[length(growth_rates) -
            ↪ future_years + i] / 100)
97         push!(future_population, new_population) # Neue Bevölkerungszahl zur Liste
            ↪ hinzufügen
98     end
99     return future_population
100 end
101
102
103 start_population = historical_population[end] # Startpopulation ist die letzte bekannte
    ↪ Bevölkerungszahl
104 future_years = 2022:2100 # Zukunftsjahre von 2022 bis 2100
105 extended_growth_rates = extend_growth_rates(growth_rates, length(future_years)) #
    ↪ Erweitere die Wachstumsraten um die Zukunftsjahre

```



```
106 future_population = forecast_population(start_population, extended_growth_rates,  
    ↪ length(future_years)) # Prognostiziere die zukünftige Bevölkerung  
107  
108 # Plot der jährlichen Wachstumsrate in Prozent  
109 plot(years, growth_rates, label="Jährliche Wachstumsrate", xlabel="Jahr",  
    ↪ ylabel="Wachstumsrate (%)", title="Jährliche Wachstumsrate von China (1960-2022)",  
    ↪ lw=2, color=:blue)  
110 savefig("jaehrliche_wachstumsrate_china_1960_2022.png") # Speichere den Plot der  
    ↪ jährlichen Wachstumsrate  
111  
112 # Plot der historischen und prognostizierten Bevölkerung  
113 plot(years, historical_population, label="Historische Bevölkerung", xlabel="Jahr",  
    ↪ ylabel="Bevölkerung", title="Bevölkerungsprognose für China", lw=2)  
114 plot!(future_years, future_population[2:end], label="Prognostizierte Bevölkerung", lw=2,  
    ↪ color=:red)  
115 savefig("bevoelkerungsprognose_china.png") # Speichere den Plot der Bevölkerungsprognose  
116  
117 # Weitere Visualisierung: Geburtenrate vs. Sterberate  
118 plot(years, birth_rates, label="Geburtenrate", xlabel="Jahr", ylabel="Rate (pro 1000  
    ↪ Personen)", title="Geburtenrate vs. Sterberate in China", lw=2)  
119 plot!(years, death_rates, label="Sterberate", lw=2, color=:red)  
120 savefig("geburtenrate_vs_sterberate_china.png") # Speichere den Plot der Geburten- und  
    ↪ Sterberaten
```