



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: 2. Juni 2024

Inhaltsverzeichnis

Abbildungsverzeichnis	i
1 Einleitung	1
1.1 Problemstellung	1
1.2 Methodik und Strategie	2
2 Technische Grundlagen	2
3 Implementierung in Julia	2
4 Ergebnisse	5

Abbildungsverzeichnis

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 modellieren und zu analysieren. 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 verschiedenen Quellen, darunter:

- Bevölkerungsstatistiken der Vereinten Nationen (UN)
- National Bureau of Statistics of China (NBS)
- WorldPop Datenbank

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

- Vorhersage des Bevölkerungsrückgangs
- Veränderung der Altersstruktur
- Auswirkungen auf die wirtschaftliche und soziale Struktur

Die Verwendung von Julia zur Modellierung und Analyse der demografischen Entwicklung Chinas bis 2100 bietet eine leistungsstarke und effiziente Methode, um komplexe Daten zu verarbeiten und aussagekräftige Ergebnisse zu erzielen.

2 Technische Grundlagen

3 Implementierung in Julia

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

- HTTP.jl
- JSON.jl
- Plots.jl

```
1 using HTTP
2 using JSON
3 using Plots
4 using Statistics
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, end_year::Int)
8     # URL für den API-Aufruf zur Abrufung der Daten von der Weltbank
9     url = "http://api.worldbank.org/v2/country/$country_code/indicator/$indicator?date=$start_year:$end_year"
10    # HTTP-GET-Anfrage an die URL
```

```

11 response = HTTP.get(url)
12 # Überprüfen, ob die Anfrage erfolgreich war
13 if response.status == 200
14     # JSON-Daten aus der Antwort parsen
15     data = JSON.parse(String(response.body))
16     return data
17 else
18     # Fehlermeldung ausgeben, wenn die Anfrage nicht erfolgreich war
19     println("Fehler beim Abrufen der Daten für den Indikator $indicator: HTTP-Statuscode ", response.status)
20     return nothing
21 end
22 end
23
24 # Abrufen der Geburten-, Sterberaten und Bevölkerung von China für die Jahre 1960 bis 2022
25 country_code = "CHN" # Ländercode für China
26 start_year = 1960 # Startjahr
27 end_year = 2022 # Endjahr
28
29 # Indikatoren für Geburten-, Sterberaten und Bevölkerung
30 birth_rate_indicator = "SP.DYN.CBRT.IN" # Indikator für die Geburtenrate
31 death_rate_indicator = "SP.DYN.CDRT.IN" # Indikator für die Sterberate
32 population_indicator = "SP.POP.TOTL" # Indikator für die Gesamtbevölkerung
33
34 # Abrufen der Daten für die jeweiligen Indikatoren
35 birth_data = get_indicator_data(country_code, birth_rate_indicator, start_year, end_year)
36 death_data = get_indicator_data(country_code, death_rate_indicator, start_year, end_year)
37 population_data = get_indicator_data(country_code, population_indicator, start_year, end_year)
38
39 # Extrahieren der Daten
40 years = start_year:end_year # Jahre von 1960 bis 2022
41 birth_rates = [NaN for _ in years] # Initialisiere Geburtenraten mit NaN
42 death_rates = [NaN for _ in years] # Initialisiere Sterberaten mit NaN
43 historical_population = [NaN for _ in years] # Initialisiere historische Bevölkerung mit NaN
44
45 # Extrahieren der Geburtenraten
46 for record in birth_data[2]
47     year = parse(Int, record["date"]) # Jahr aus dem Datensatz parsen
48     if year in years
49         birth_rates[year - start_year + 1] = record["value"] # Geburtenrate für das entsprechende Jahr
50     end
51 end
52
53 # Extrahieren der Sterberaten
54 for record in death_data[2]
55     year = parse(Int, record["date"]) # Jahr aus dem Datensatz parsen
56     if year in years
57         death_rates[year - start_year + 1] = record["value"] # Sterberate für das entsprechende Jahr
58     end
59 end
60

```

```

61 # Extrahieren der Bevölkerungsdaten
62 for record in population_data[2]
63     year = parse(Int, record["date"]) # Jahr aus dem Datensatz parsen
64     if year in years
65         historical_population[year - start_year + 1] = record["value"] # Bevölkerungszahl für das entsp
66     end
67 end
68
69 # Berechnung der jährlichen Wachstumsrate in Prozent
70 function calculate_growth_rate(birth_rates::Vector{Float64}, death_rates::Vector{Float64})
71     growth_rates = [NaN for _ in birth_rates] # Initialisiere Wachstumsraten mit NaN
72     for i in 1:length(birth_rates)
73         if !isnan(birth_rates[i]) && !isnan(death_rates[i]) # Überprüfen, ob Geburten- und Sterberate g
74             growth_rates[i] = (birth_rates[i] - death_rates[i]) / 10 # Berechnung der Wachstumsrate
75         end
76     end
77     return growth_rates
78 end
79
80 growth_rates = calculate_growth_rate(birth_rates, death_rates) # Berechne die jährlichen Wachstumsraten
81
82 # Prognose der zukünftigen Wachstumsraten mit Annahme einer sinkenden Bevölkerung
83 function extend_growth_rates(growth_rates::Vector{Float64}, future_years::Int)
84     avg_growth_rate = mean(skipmissing(growth_rates)) # Durchschnittliche Wachstumsrate berechnen, unt
85     # Nehmen wir an, dass die Wachstumsrate jedes Jahr leicht negativ wird, um eine moderate Abnahme d
86     extended_growth_rates = vcat(growth_rates, [0.1 - i*0.03 for i in 1:future_years])
87     return extended_growth_rates
88 end
89
90 # Funktion zur Bevölkerungsprognose
91 function forecast_population(start_population::Float64, growth_rates::Vector{Float64}, future_years::Int)
92     future_population = [start_population] # Initialisiere zukünftige Bevölkerung mit der Startpopulat
93     for i in 1:future_years
94         # Berechnung der neuen Bevölkerungszahl basierend auf der Wachstumsrate
95         new_population = future_population[end] * (1 + growth_rates[length(growth_rates) - future_years])
96         push!(future_population, new_population) # Neue Bevölkerungszahl zur Liste hinzufügen
97     end
98     return future_population
99 end
100
101 # Beispielhafte Startpopulation und Prognose bis 2040
102 start_population = historical_population[end] # Startpopulation ist die letzte bekannte Bevölkerungszahl
103 future_years = 2022:2100 # Zukunftsjahre von 2022 bis 2100
104 extended_growth_rates = extend_growth_rates(growth_rates, length(future_years)) # Erweitere die Wachst
105 future_population = forecast_population(start_population, extended_growth_rates, length(future_years))
106
107 # Plot der jährlichen Wachstumsrate in Prozent
108 plot(years, growth_rates, label="Jährliche Wachstumsrate", xlabel="Jahr", ylabel="Wachstumsrate (%)", t
109 savefig("jaehrliche_wachstumsrate_china_1960_2022.png") # Speichere den Plot der jährlichen Wachstumsr
110

```

```

111 # Plot der historischen und prognostizierten Bevölkerung
112 plot(years, historical_population, label="Historische Bevölkerung", xlabel="Jahr", ylabel="Bevölkerung",
113 plot!(future_years, future_population[2:end], label="Prognostizierte Bevölkerung", lw=2, color=:red)
114 savefig("bevoelkerungsprognose_china.png") # Speichere den Plot der Bevölkerungsprognose
115
116 # Weitere Visualisierung: Geburtenrate vs. Sterberate
117 plot(years, birth_rates, label="Geburtenrate", xlabel="Jahr", ylabel="Rate (pro 1000 Personen)", title="Geburten- und Sterberaten",
118 plot!(years, death_rates, label="Sterberate", lw=2, color=:red)
119 savefig("geburtenrate_vs_sterberate_china.png") # Speichere den Plot der Geburten- und Sterberaten

```

4 Ergebnisse