

Visualizing the Uninsured: A Data Science Perspective on U.S. Health Coverage

Eduardo Sáenz-Messía Laborda
Catherina Mikhail (2847118)
Charlotte Craenen (2865572)
Hugo van As
Indy Pleijter
Julian Zeguers
Wies Polderman
Tutorial group 5, group 4 – J.F. Fitzgerald

2025-06-24

Contents

0.1	1.1 Describe the Social Problem	2
0.2	1.2 Provide Background on the Problem	2
1	Part 2 – Describe and Acquire Data	2
1.1	2.1 Describe the Dataset	2
1.2	2.2 Import and Prepare the Dataset	2
1.3	Data cleaning: Datasets of Uninsured Population and The Creation of The New Variable: The Uninsured Share	8
1.4	datacleaning Annual GDP Table and The Creation of The New Variable: Annual Real GDP Growth per year and state	69
2	Part 3 – Visualize and Analyze the Data	70
2.1	3.1 Create Initial Visualizations	70
2.2	3.2 Identify Trends and Patterns	75
3	Part 4 – Communicate Findings	75
3.1	4.1 Summarize Key Insights	75
3.2	4.2 Propose Solutions or Policy Recommendations	75
4	Appendix	75
4.1	A.1 References	75
4.2	A.2 Session Info	75
Part 1 – Identify a Social Problem		

0.1 1.1 Describe the Social Problem

The high number of uninsured Americans is a persistent issue with substantial social and economic consequences. According to the U.S. Census Bureau, over 27 million Americans lacked health insurance in 2022. This lack of coverage can lead to delayed medical care, poor health outcomes, and financial instability.

0.2 1.2 Provide Background on the Problem

Since the implementation of the Affordable Care Act (ACA) in 2010, the number of uninsured has decreased, but disparities remain. People with low income, people of color, and those living in states that did not expand Medicaid are disproportionately affected. Health insurance in the U.S. is often tied to employment, further complicating access for those in part-time, temporary, or informal jobs.

1 Part 2 – Describe and Acquire Data

1.1 2.1 Describe the Dataset

The dataset used is derived from the United States Census Bureau and includes individual-level data on health insurance coverage, demographics, and employment status.

1.2 2.2 Import and Prepare the Dataset

```
library(tidyverse) library(tidyr) library(readr) library(ggplot2) library(rnaturalearth) library(maps)
library(dplyr) library(purrr) library(stringr)
```

```
# The first dataset is from the dataset "selected characteristics of the uninsured in the US from 2010-
```

```
#2010
```

```
ACSST1Y2010_S2702_2025_06_03T123607 <- read_csv("data insured population 2010-2023/ACSST1Y2010.S2702-2025-06-03T123607.csv")
```

```
## Rows: 120 Columns: 209
## -- Column specification -----
## Delimiter: ","
## chr (193): Label (Grouping), Alabama!!Total Civilian Noninstitutionalized Po...
## lgl (16): District of Columbia!!Total Civilian Noninstitutionalized Populat...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
dataset_uninsured_population_2010 = read_csv("data insured population 2010-2023/ACSST1Y2010.S2702-2025-06-03T123607.csv")
```

```
## Rows: 120 Columns: 209
## -- Column specification -----
## Delimiter: ","
## chr (193): Label (Grouping), Alabama!!Total Civilian Noninstitutionalized Po...
## lgl (16): District of Columbia!!Total Civilian Noninstitutionalized Populat...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

#2011

```
ACSST1Y2011_S2702_2025_06_03T123539 <- read_csv("data insured population 2010-2023/ACSST1Y2011.S2702-2025-06-03T123539.csv")
```

```
## Rows: 120 Columns: 209
## -- Column specification -----
## Delimiter: ","
## chr (197): Label (Grouping), Alabama!!Total Civilian Noninstitutionalized Po...
## lgl (12): District of Columbia!!Total Civilian Noninstitutionalized Populat...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
dataset_uninsured_population_2011 = read_csv("data insured population 2010-2023/ACSST1Y2011.S2702-2025-06-03T123539.csv")
```

```
## Rows: 120 Columns: 209
## -- Column specification -----
## Delimiter: ","
## chr (197): Label (Grouping), Alabama!!Total Civilian Noninstitutionalized Po...
## lgl (12): District of Columbia!!Total Civilian Noninstitutionalized Populat...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

#2012

```
ACSST1Y2012_S2702_2025_06_03T123441 <- read_csv("data insured population 2010-2023/ACSST1Y2012.S2702-2025-06-03T123441.csv")
```

```
## Rows: 120 Columns: 209
## -- Column specification -----
## Delimiter: ","
## chr (197): Label (Grouping), Alabama!!Total Civilian Noninstitutionalized Po...
## lgl (12): District of Columbia!!Total Civilian Noninstitutionalized Populat...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
dataset_uninsured_population_2012 = read_csv("data insured population 2010-2023/ACSST1Y2012.S2702-2025-06-03T123441.csv")
```

```
## Rows: 120 Columns: 209
## -- Column specification -----
## Delimiter: ","
## chr (197): Label (Grouping), Alabama!!Total Civilian Noninstitutionalized Po...
## lgl (12): District of Columbia!!Total Civilian Noninstitutionalized Populat...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

#2013

```
ACSST1Y2013_S2702_2025_06_03T123404 <- read_csv("data insured population 2010-2023/ACSST1Y2013.S2702-2025-06-03T123404.csv")
```

```
## Rows: 120 Columns: 209
## -- Column specification -----
## Delimiter: ","
## chr (197): Label (Grouping), Alabama!!Total Civilian Noninstitutionalized Po...
## lgl (12): District of Columbia!!Total Civilian Noninstitutionalized Populat...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
dataset_uninsured_population_2013 = read_csv("data insured population 2010-2023/ACSST1Y2013.S2702-2025-")
```

```
## Rows: 120 Columns: 209
## -- Column specification -----
## Delimiter: ","
## chr (197): Label (Grouping), Alabama!!Total Civilian Noninstitutionalized Po...
## lgl (12): District of Columbia!!Total Civilian Noninstitutionalized Populat...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

#2014

```
ACSST1Y2014_S2702_2025_06_03T123336 <- read_csv("data insured population 2010-2023/ACSST1Y2014.S2702-2025-")
```

```
## Rows: 120 Columns: 209
## -- Column specification -----
## Delimiter: ","
## chr (193): Label (Grouping), Alabama!!Total Civilian Noninstitutionalized Po...
## lgl (16): District of Columbia!!Total Civilian Noninstitutionalized Populat...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
dataset_uninsured_population_2014 = read_csv("data insured population 2010-2023/ACSST1Y2014.S2702-2025-")
```

```
## Rows: 120 Columns: 209
## -- Column specification -----
## Delimiter: ","
## chr (193): Label (Grouping), Alabama!!Total Civilian Noninstitutionalized Po...
## lgl (16): District of Columbia!!Total Civilian Noninstitutionalized Populat...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

#2015

```
ACSST1Y2015_S2702_2025_06_03T123312 <- read_csv("data insured population 2010-2023/ACSST1Y2015.S2702-2025-")
```

```
## Rows: 120 Columns: 209
## -- Column specification -----
## Delimiter: ","
## chr (181): Label (Grouping), Alabama!!Total Civilian Noninstitutionalized Po...
## lgl (28): Delaware!!Total Civilian Noninstitutionalized Population!!Estimat...
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.

dataset_uninsured_population_2015 = read_csv("data insured population 2010-2023/ACSST1Y2015.S2702-2025-06_03T123232")

## Rows: 120 Columns: 209
## -- Column specification -----
## Delimiter: ","
## chr (181): Label (Grouping), Alabama!!Total Civilian Noninstitutionalized Po...
## lgl (28): Delaware!!Total Civilian Noninstitutionalized Population!!Estimat...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.

#2016
ACSST1Y2016_S2702_2025_06_03T123232 <- read_csv("data insured population 2010-2023/ACSST1Y2016.S2702-2025-06_03T123232")

## Rows: 120 Columns: 209
## -- Column specification -----
## Delimiter: ","
## chr (181): Label (Grouping), Alabama!!Total Civilian Noninstitutionalized Po...
## lgl (28): Delaware!!Total Civilian Noninstitutionalized Population!!Estimat...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.

dataset_uninsured_population_2016 = read_csv("data insured population 2010-2023/ACSST1Y2016.S2702-2025-06_03T123232")

## Rows: 120 Columns: 209
## -- Column specification -----
## Delimiter: ","
## chr (181): Label (Grouping), Alabama!!Total Civilian Noninstitutionalized Po...
## lgl (28): Delaware!!Total Civilian Noninstitutionalized Population!!Estimat...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.

#2017
ACSST1Y2017_S2702_2025_06_03T123200 <- read_csv("data insured population 2010-2023/ACSST1Y2017.S2702-2025-06_03T123200")

## Rows: 120 Columns: 209
## -- Column specification -----
## Delimiter: ","
## chr (181): Label (Grouping), Alabama!!Total!!Estimate, Alabama!!Total!!Margi...
## lgl (28): Delaware!!Total!!Estimate, Delaware!!Total!!Margin of Error, Dela...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
dataset_uninsured_population_2017 = read_csv("data insured population 2010-2023/ACSST1Y2017.S2702-2025-06_03T123128")
```

```
## Rows: 120 Columns: 209
## -- Column specification -----
## Delimiter: ","
## chr (181): Label (Grouping), Alabama!!Total!!Estimate, Alabama!!Total!!Margi...
## lgl (28): Delaware!!Total!!Estimate, Delaware!!Total!!Margin of Error, Dela...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

#2018

```
ACSST1Y2018_S2702_2025_06_03T123128 <- read_csv("data insured population 2010-2023/ACSST1Y2018.S2702-2025-06_03T123128")
```

```
## Rows: 120 Columns: 209
## -- Column specification -----
## Delimiter: ","
## chr (177): Label (Grouping), Alabama!!Total!!Estimate, Alabama!!Total!!Margi...
## lgl (32): Delaware!!Total!!Estimate, Delaware!!Total!!Margin of Error, Dela...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
dataset_uninsured_population_2018 = read_csv("data insured population 2010-2023/ACSST1Y2018.S2702-2025-06_03T123128")
```

```
## Rows: 120 Columns: 209
## -- Column specification -----
## Delimiter: ","
## chr (177): Label (Grouping), Alabama!!Total!!Estimate, Alabama!!Total!!Margi...
## lgl (32): Delaware!!Total!!Estimate, Delaware!!Total!!Margin of Error, Dela...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

#2019

```
ACSST1Y2019_S2702_2025_06_03T123018 <- read_csv("data insured population 2010-2023/ACSST1Y2019.S2702-2025-06_03T123018")
```

```
## Rows: 120 Columns: 209
## -- Column specification -----
## Delimiter: ","
## chr (181): Label (Grouping), Alabama!!Total!!Estimate, Alabama!!Total!!Margi...
## lgl (28): Delaware!!Total!!Estimate, Delaware!!Total!!Margin of Error, Dela...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
dataset_uninsured_population_2019 = read_csv("data insured population 2010-2023/ACSST1Y2019.S2702-2025-06_03T123018")
```

```
## Rows: 120 Columns: 209
## -- Column specification -----
```

```
## Delimiter: ","
## chr (181): Label (Grouping), Alabama!!Total!!Estimate, Alabama!!Total!!Margi...
## lgl (28): Delaware!!Total!!Estimate, Delaware!!Total!!Margin of Error, Dela...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

#2021

```
ACSST1Y2021_S2702_2025_06_03T122952 <- read_csv("data insured population 2010-2023/ACSST1Y2021.S2702-2025-06-03T122952.csv")
```

```
## Rows: 120 Columns: 209
## -- Column specification -----
## Delimiter: ","
## chr (181): Label (Grouping), Alabama!!Total!!Estimate, Alabama!!Total!!Margi...
## lgl (28): Delaware!!Total!!Estimate, Delaware!!Total!!Margin of Error, Dela...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
dataset_uninsured_population_2021 = read_csv("data insured population 2010-2023/ACSST1Y2021.S2702-2025-06-03T122952.csv")
```

```
## Rows: 120 Columns: 209
## -- Column specification -----
## Delimiter: ","
## chr (181): Label (Grouping), Alabama!!Total!!Estimate, Alabama!!Total!!Margi...
## lgl (28): Delaware!!Total!!Estimate, Delaware!!Total!!Margin of Error, Dela...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

#2022

```
ACSST1Y2022_S2702_2025_06_03T122917 <- read_csv("data insured population 2010-2023/ACSST1Y2022.S2702-2025-06-03T122917.csv")
```

```
## Rows: 120 Columns: 209
## -- Column specification -----
## Delimiter: ","
## chr (181): Label (Grouping), Alabama!!Total!!Estimate, Alabama!!Total!!Margi...
## lgl (28): Delaware!!Total!!Estimate, Delaware!!Total!!Margin of Error, Dela...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
dataset_uninsured_population_2022 = read_csv("data insured population 2010-2023/ACSST1Y2022.S2702-2025-06-03T122917.csv")
```

```
## Rows: 120 Columns: 209
## -- Column specification -----
## Delimiter: ","
## chr (181): Label (Grouping), Alabama!!Total!!Estimate, Alabama!!Total!!Margi...
## lgl (28): Delaware!!Total!!Estimate, Delaware!!Total!!Margin of Error, Dela...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
#2023
```

```
ACSST1Y2023_S2702_2025_06_04T165512 <- read_csv("data insured population 2010-2023/ACSST1Y2023.S2702-2025-06-04T165512.csv")
```

```
## Rows: 120 Columns: 209
## -- Column specification -----
## Delimiter: ","
## chr (181): Label (Grouping), Alabama!!Total!!Estimate, Alabama!!Total!!Margi...
## lgl (28): District of Columbia!!Total!!Estimate, District of Columbia!!Tota...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
dataset_uninsured_population_2023 = read_csv("data insured population 2010-2023/ACSST1Y2023.S2702-2025-06-04T165512.csv")
```

```
## Rows: 120 Columns: 209
## -- Column specification -----
## Delimiter: ","
## chr (181): Label (Grouping), Alabama!!Total!!Estimate, Alabama!!Total!!Margi...
## lgl (28): District of Columbia!!Total!!Estimate, District of Columbia!!Tota...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
#####
```

```
#The second dataset is from the Bureau of Economic Analysis and is the annual real GDP of the 44 states
```

```
Table_GDP_absolute_values_Table_3_ <- read_csv("data GDP/Table GDP absolute values - Table (3).csv")
```

```
## Rows: 45 Columns: 15
## -- Column specification -----
## Delimiter: ","
## chr (1): GeoName
## dbl (14): 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019, ...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
#renaming the table
```

```
table_GDP = Table_GDP_absolute_values_Table_3_
```

1.3 Data cleaning: Datasets of Uninsured Population and The Creation of The New Variable: The Uninsured Share

- describe here which states we first took out the margin of errors, then we took out the 6 small states and tell which states they are (deleware, district of columbia, hawaii, puerto rico, north dakota, rhode island, vermont and wyoming check de spelling!!), then that we took out the metadata and tell which metadata that is: age, sex, race and hispanic or latino origin, nativity and U.S. citizenship status, disability status, residence 1 year ago, educational attainment, employment status, work experience, civilian noninstitutionalized workers 16 years and over, earnings in the past 12 months and ratio of income to poverty level in the past 12 months. and that we only kept the total civilian noninstitutionalized

population and household income (inflation adjusted). After that we coerced the character values in the yearly datasets to numeric and created our new variable of uninsured share. After that we combined all the yearly datasets

```
# This is me cleaning out the data insured population from 2010-2016 getting out the margin of error
dataset_uninsured_population_2010 = dataset_uninsured_population_2010[, which(!grepl("Margin of Error",
View(dataset_uninsured_population_2010)

dataset_uninsured_population_2011 = dataset_uninsured_population_2011[, which(!grepl("Margin of Error",
View(dataset_uninsured_population_2011)

dataset_uninsured_population_2012 = dataset_uninsured_population_2012[, which(!grepl("Margin of Error",
View(dataset_uninsured_population_2012)

dataset_uninsured_population_2013 = dataset_uninsured_population_2013[, which(!grepl("Margin of Error",
View(dataset_uninsured_population_2013)

dataset_uninsured_population_2014 = dataset_uninsured_population_2014[, which(!grepl("Margin of Error",
View(dataset_uninsured_population_2014)

dataset_uninsured_population_2015 = dataset_uninsured_population_2015[, which(!grepl("Margin of Error",
View(dataset_uninsured_population_2015)

dataset_uninsured_population_2016 = dataset_uninsured_population_2016[, which(!grepl("Margin of Error",
View(dataset_uninsured_population_2016)

dataset_uninsured_population_2017 = dataset_uninsured_population_2017[, which(!grepl("Margin of Error",
View(dataset_uninsured_population_2017)

dataset_uninsured_population_2018 = dataset_uninsured_population_2018[, which(!grepl("Margin of Error",
View(dataset_uninsured_population_2018)

dataset_uninsured_population_2019 = dataset_uninsured_population_2019[, which(!grepl("Margin of Error",
View(dataset_uninsured_population_2019)

dataset_uninsured_population_2021 = dataset_uninsured_population_2021[, which(!grepl("Margin of Error",
View(dataset_uninsured_population_2021)

dataset_uninsured_population_2022 = dataset_uninsured_population_2022[, which(!grepl("Margin of Error",
View(dataset_uninsured_population_2022)

dataset_uninsured_population_2023 = dataset_uninsured_population_2023[, which(!grepl("Margin of Error",
View(dataset_uninsured_population_2023)

#####

#This is me cleaning out the columns of Delaware
dataset_uninsured_population_2010 = dataset_uninsured_population_2010[, which(!grepl("Delaware", colnam
View(dataset_uninsured_population_2010)

dataset_uninsured_population_2011 = dataset_uninsured_population_2011[, which(!grepl("Delaware", colnam
View(dataset_uninsured_population_2011)

dataset_uninsured_population_2012 = dataset_uninsured_population_2012[, which(!grepl("Delaware", colnam
```

```

View(dataset_uninsured_population_2012)

dataset_uninsured_population_2013 = dataset_uninsured_population_2013[, which(!grepl("Delaware", colnames(dataset_uninsured_population_2013)))]
View(dataset_uninsured_population_2013)

dataset_uninsured_population_2014 = dataset_uninsured_population_2014[, which(!grepl("Delaware", colnames(dataset_uninsured_population_2014)))]
View(dataset_uninsured_population_2014)

dataset_uninsured_population_2015 = dataset_uninsured_population_2015[, which(!grepl("Delaware", colnames(dataset_uninsured_population_2015)))]
View(dataset_uninsured_population_2015)

dataset_uninsured_population_2016 = dataset_uninsured_population_2016[, which(!grepl("Delaware", colnames(dataset_uninsured_population_2016)))]
View(dataset_uninsured_population_2016)

dataset_uninsured_population_2017 = dataset_uninsured_population_2017[, which(!grepl("Delaware", colnames(dataset_uninsured_population_2017)))]
View(dataset_uninsured_population_2017)

dataset_uninsured_population_2018 = dataset_uninsured_population_2018[, which(!grepl("Delaware", colnames(dataset_uninsured_population_2018)))]
View(dataset_uninsured_population_2018)

dataset_uninsured_population_2019 = dataset_uninsured_population_2019[, which(!grepl("Delaware", colnames(dataset_uninsured_population_2019)))]
View(dataset_uninsured_population_2019)

dataset_uninsured_population_2021 = dataset_uninsured_population_2021[, which(!grepl("Delaware", colnames(dataset_uninsured_population_2021)))]
View(dataset_uninsured_population_2021)

dataset_uninsured_population_2022 = dataset_uninsured_population_2022[, which(!grepl("Delaware", colnames(dataset_uninsured_population_2022)))]
View(dataset_uninsured_population_2022)

dataset_uninsured_population_2023 = dataset_uninsured_population_2023[, which(!grepl("Delaware", colnames(dataset_uninsured_population_2023)))]
View(dataset_uninsured_population_2023)

#This is me cleaning out the columns of District of Columbia
dataset_uninsured_population_2011 = dataset_uninsured_population_2011[, which(!grepl("District of Columbia", colnames(dataset_uninsured_population_2011)))]
View(dataset_uninsured_population_2011)

dataset_uninsured_population_2012 = dataset_uninsured_population_2012[, which(!grepl("District of Columbia", colnames(dataset_uninsured_population_2012)))]
View(dataset_uninsured_population_2012)

dataset_uninsured_population_2013 = dataset_uninsured_population_2013[, which(!grepl("District of Columbia", colnames(dataset_uninsured_population_2013)))]
View(dataset_uninsured_population_2013)

dataset_uninsured_population_2014 = dataset_uninsured_population_2014[, which(!grepl("District of Columbia", colnames(dataset_uninsured_population_2014)))]
View(dataset_uninsured_population_2014)

dataset_uninsured_population_2015 = dataset_uninsured_population_2015[, which(!grepl("District of Columbia", colnames(dataset_uninsured_population_2015)))]
View(dataset_uninsured_population_2015)

dataset_uninsured_population_2016 = dataset_uninsured_population_2016[, which(!grepl("District of Columbia", colnames(dataset_uninsured_population_2016)))]
View(dataset_uninsured_population_2016)

dataset_uninsured_population_2017 = dataset_uninsured_population_2017[, which(!grepl("District of Columbia", colnames(dataset_uninsured_population_2017)))]
View(dataset_uninsured_population_2017)

```

```

dataset_uninsured_population_2018 = dataset_uninsured_population_2018[, which(!grepl("District of Colum
View(dataset_uninsured_population_2018)

dataset_uninsured_population_2019 = dataset_uninsured_population_2019[, which(!grepl("District of Colum
View(dataset_uninsured_population_2019)

dataset_uninsured_population_2021 = dataset_uninsured_population_2021[, which(!grepl("District of Colum
View(dataset_uninsured_population_2021)

dataset_uninsured_population_2022 = dataset_uninsured_population_2022[, which(!grepl("District of Colum
View(dataset_uninsured_population_2022)

dataset_uninsured_population_2023 = dataset_uninsured_population_2023[, which(!grepl("District of Colum
View(dataset_uninsured_population_2023)

#This is me cleaning out the columns of Hawaii
dataset_uninsured_population_2010 = dataset_uninsured_population_2010[, which(!grepl("Hawaii", colnames
View(dataset_uninsured_population_2010)

dataset_uninsured_population_2011 = dataset_uninsured_population_2011[, which(!grepl("Hawaii", colnames
View(dataset_uninsured_population_2011)

dataset_uninsured_population_2012 = dataset_uninsured_population_2012[, which(!grepl("Hawaii", colnames
View(dataset_uninsured_population_2012)

dataset_uninsured_population_2013 = dataset_uninsured_population_2013[, which(!grepl("Hawaii", colnames
View(dataset_uninsured_population_2013)

dataset_uninsured_population_2014 = dataset_uninsured_population_2014[, which(!grepl("Hawaii", colnames
View(dataset_uninsured_population_2014)

dataset_uninsured_population_2015 = dataset_uninsured_population_2015[, which(!grepl("Hawaii", colnames
View(dataset_uninsured_population_2015)

dataset_uninsured_population_2016 = dataset_uninsured_population_2016[, which(!grepl("Hawaii", colnames
View(dataset_uninsured_population_2016)

dataset_uninsured_population_2017 = dataset_uninsured_population_2017[, which(!grepl("Hawaii", colnames
View(dataset_uninsured_population_2017)

dataset_uninsured_population_2018 = dataset_uninsured_population_2018[, which(!grepl("Hawaii", colnames
View(dataset_uninsured_population_2018)

dataset_uninsured_population_2019 = dataset_uninsured_population_2019[, which(!grepl("Hawaii", colnames
View(dataset_uninsured_population_2019)

dataset_uninsured_population_2021 = dataset_uninsured_population_2021[, which(!grepl("Hawaii", colnames
View(dataset_uninsured_population_2021)

dataset_uninsured_population_2022 = dataset_uninsured_population_2022[, which(!grepl("Hawaii", colnames
View(dataset_uninsured_population_2022)

dataset_uninsured_population_2023 = dataset_uninsured_population_2023[, which(!grepl("Hawaii", colnames

```

```

View(dataset_uninsured_population_2023)
#This is me cleaning out the columns of North Dakota
dataset_uninsured_population_2010 = dataset_uninsured_population_2010[, which(!grepl("North Dakota", col
View(dataset_uninsured_population_2010)

dataset_uninsured_population_2011 = dataset_uninsured_population_2011[, which(!grepl("North Dakota", col
View(dataset_uninsured_population_2011)

dataset_uninsured_population_2012 = dataset_uninsured_population_2012[, which(!grepl("North Dakota", col
View(dataset_uninsured_population_2012)

dataset_uninsured_population_2013 = dataset_uninsured_population_2013[, which(!grepl("North Dakota", col
View(dataset_uninsured_population_2013)

dataset_uninsured_population_2014 = dataset_uninsured_population_2014[, which(!grepl("North Dakota", col
View(dataset_uninsured_population_2014)

dataset_uninsured_population_2015 = dataset_uninsured_population_2015[, which(!grepl("North Dakota", col
View(dataset_uninsured_population_2015)

dataset_uninsured_population_2016 = dataset_uninsured_population_2016[, which(!grepl("North Dakota", col
View(dataset_uninsured_population_2016)

dataset_uninsured_population_2017 = dataset_uninsured_population_2017[, which(!grepl("North Dakota", col
View(dataset_uninsured_population_2017)

dataset_uninsured_population_2018 = dataset_uninsured_population_2018[, which(!grepl("North Dakota", col
View(dataset_uninsured_population_2018)

dataset_uninsured_population_2019 = dataset_uninsured_population_2019[, which(!grepl("North Dakota", col
View(dataset_uninsured_population_2019)

dataset_uninsured_population_2021 = dataset_uninsured_population_2021[, which(!grepl("North Dakota", col
View(dataset_uninsured_population_2021)

dataset_uninsured_population_2022 = dataset_uninsured_population_2022[, which(!grepl("North Dakota", col
View(dataset_uninsured_population_2022)

dataset_uninsured_population_2023 = dataset_uninsured_population_2023[, which(!grepl("North Dakota", col
View(dataset_uninsured_population_2023)
#This is me cleaning out the columns of Puerto Rico
dataset_uninsured_population_2010 = dataset_uninsured_population_2010[, which(!grepl("Puerto Rico", col
View(dataset_uninsured_population_2010)

dataset_uninsured_population_2011 = dataset_uninsured_population_2011[, which(!grepl("Puerto Rico", col
View(dataset_uninsured_population_2011)

dataset_uninsured_population_2012 = dataset_uninsured_population_2012[, which(!grepl("Puerto Rico", col
View(dataset_uninsured_population_2012)

dataset_uninsured_population_2013 = dataset_uninsured_population_2013[, which(!grepl("Puerto Rico", col
View(dataset_uninsured_population_2013)

```

```

dataset_uninsured_population_2014 = dataset_uninsured_population_2014[, which(!grepl("Puerto Rico", col
View(dataset_uninsured_population_2014)

dataset_uninsured_population_2015 = dataset_uninsured_population_2015[, which(!grepl("Puerto Rico", col
View(dataset_uninsured_population_2015)

dataset_uninsured_population_2016 = dataset_uninsured_population_2016[, which(!grepl("Puerto Rico", col
View(dataset_uninsured_population_2016)

dataset_uninsured_population_2017 = dataset_uninsured_population_2017[, which(!grepl("Puerto Rico", col
View(dataset_uninsured_population_2017)

dataset_uninsured_population_2018 = dataset_uninsured_population_2018[, which(!grepl("Puerto Rico", col
View(dataset_uninsured_population_2018)

dataset_uninsured_population_2019 = dataset_uninsured_population_2019[, which(!grepl("Puerto Rico", col
View(dataset_uninsured_population_2019)

dataset_uninsured_population_2021 = dataset_uninsured_population_2021[, which(!grepl("Puerto Rico", col
View(dataset_uninsured_population_2021)

dataset_uninsured_population_2022 = dataset_uninsured_population_2022[, which(!grepl("Puerto Rico", col
View(dataset_uninsured_population_2022)

dataset_uninsured_population_2023 = dataset_uninsured_population_2023[, which(!grepl("Puerto Rico", col
View(dataset_uninsured_population_2023)
#This is me cleaning out the columns of Rhode Island
dataset_uninsured_population_2010 = dataset_uninsured_population_2010[, which(!grepl("Rhode Island", col
View(dataset_uninsured_population_2010)

dataset_uninsured_population_2011 = dataset_uninsured_population_2011[, which(!grepl("Rhode Island", col
View(dataset_uninsured_population_2011)

dataset_uninsured_population_2012 = dataset_uninsured_population_2012[, which(!grepl("Rhode Island", col
View(dataset_uninsured_population_2012)

dataset_uninsured_population_2013 = dataset_uninsured_population_2013[, which(!grepl("Rhode Island", col
View(dataset_uninsured_population_2013)

dataset_uninsured_population_2014 = dataset_uninsured_population_2014[, which(!grepl("Rhode Island", col
View(dataset_uninsured_population_2014)

dataset_uninsured_population_2015 = dataset_uninsured_population_2015[, which(!grepl("Rhode Island", col
View(dataset_uninsured_population_2015)

dataset_uninsured_population_2016 = dataset_uninsured_population_2016[, which(!grepl("Rhode Island", col
View(dataset_uninsured_population_2016)

dataset_uninsured_population_2017 = dataset_uninsured_population_2017[, which(!grepl("Rhode Island", col
View(dataset_uninsured_population_2017)

dataset_uninsured_population_2018 = dataset_uninsured_population_2018[, which(!grepl("Rhode Island", col
View(dataset_uninsured_population_2018)

```

```

dataset_uninsured_population_2019 = dataset_uninsured_population_2019[, which(!grepl("Rhode Island", colnames(dataset_uninsured_population_2019)))]
View(dataset_uninsured_population_2019)

dataset_uninsured_population_2021 = dataset_uninsured_population_2021[, which(!grepl("Rhode Island", colnames(dataset_uninsured_population_2021)))]
View(dataset_uninsured_population_2021)

dataset_uninsured_population_2022 = dataset_uninsured_population_2022[, which(!grepl("Rhode Island", colnames(dataset_uninsured_population_2022)))]
View(dataset_uninsured_population_2022)

dataset_uninsured_population_2023 = dataset_uninsured_population_2023[, which(!grepl("Rhode Island", colnames(dataset_uninsured_population_2023)))]
View(dataset_uninsured_population_2023)
#This is me cleaning out the columns of Vermont
dataset_uninsured_population_2010 = dataset_uninsured_population_2010[, which(!grepl("Vermont", colnames(dataset_uninsured_population_2010)))]
View(dataset_uninsured_population_2010)

dataset_uninsured_population_2011 = dataset_uninsured_population_2011[, which(!grepl("Vermont", colnames(dataset_uninsured_population_2011)))]
View(dataset_uninsured_population_2011)

dataset_uninsured_population_2012 = dataset_uninsured_population_2012[, which(!grepl("Vermont", colnames(dataset_uninsured_population_2012)))]
View(dataset_uninsured_population_2012)

dataset_uninsured_population_2013 = dataset_uninsured_population_2013[, which(!grepl("Vermont", colnames(dataset_uninsured_population_2013)))]
View(dataset_uninsured_population_2013)

dataset_uninsured_population_2014 = dataset_uninsured_population_2014[, which(!grepl("Vermont", colnames(dataset_uninsured_population_2014)))]
View(dataset_uninsured_population_2014)

dataset_uninsured_population_2015 = dataset_uninsured_population_2015[, which(!grepl("Vermont", colnames(dataset_uninsured_population_2015)))]
View(dataset_uninsured_population_2015)

dataset_uninsured_population_2016 = dataset_uninsured_population_2016[, which(!grepl("Vermont", colnames(dataset_uninsured_population_2016)))]
View(dataset_uninsured_population_2016)

dataset_uninsured_population_2017 = dataset_uninsured_population_2017[, which(!grepl("Vermont", colnames(dataset_uninsured_population_2017)))]
View(dataset_uninsured_population_2017)

dataset_uninsured_population_2018 = dataset_uninsured_population_2018[, which(!grepl("Vermont", colnames(dataset_uninsured_population_2018)))]
View(dataset_uninsured_population_2018)

dataset_uninsured_population_2019 = dataset_uninsured_population_2019[, which(!grepl("Vermont", colnames(dataset_uninsured_population_2019)))]
View(dataset_uninsured_population_2019)

dataset_uninsured_population_2021 = dataset_uninsured_population_2021[, which(!grepl("Vermont", colnames(dataset_uninsured_population_2021)))]
View(dataset_uninsured_population_2021)

dataset_uninsured_population_2022 = dataset_uninsured_population_2022[, which(!grepl("Vermont", colnames(dataset_uninsured_population_2022)))]
View(dataset_uninsured_population_2022)

dataset_uninsured_population_2023 = dataset_uninsured_population_2023[, which(!grepl("Vermont", colnames(dataset_uninsured_population_2023)))]
View(dataset_uninsured_population_2023)
#This is me cleaning out the columns of Wyoming
dataset_uninsured_population_2010 = dataset_uninsured_population_2010[, which(!grepl("Wyoming", colnames(dataset_uninsured_population_2010)))]
View(dataset_uninsured_population_2010)

```

```

dataset_uninsured_population_2011 = dataset_uninsured_population_2011[, which(!grepl("Wyoming", colnames(dataset_uninsured_population_2011)))]
View(dataset_uninsured_population_2011)

dataset_uninsured_population_2012 = dataset_uninsured_population_2012[, which(!grepl("Wyoming", colnames(dataset_uninsured_population_2012)))]
View(dataset_uninsured_population_2012)

dataset_uninsured_population_2013 = dataset_uninsured_population_2013[, which(!grepl("Wyoming", colnames(dataset_uninsured_population_2013)))]
View(dataset_uninsured_population_2013)

dataset_uninsured_population_2014 = dataset_uninsured_population_2014[, which(!grepl("Wyoming", colnames(dataset_uninsured_population_2014)))]
View(dataset_uninsured_population_2014)

dataset_uninsured_population_2015 = dataset_uninsured_population_2015[, which(!grepl("Wyoming", colnames(dataset_uninsured_population_2015)))]
View(dataset_uninsured_population_2015)

dataset_uninsured_population_2016 = dataset_uninsured_population_2016[, which(!grepl("Wyoming", colnames(dataset_uninsured_population_2016)))]
View(dataset_uninsured_population_2016)

dataset_uninsured_population_2017 = dataset_uninsured_population_2017[, which(!grepl("Wyoming", colnames(dataset_uninsured_population_2017)))]
View(dataset_uninsured_population_2017)

dataset_uninsured_population_2018 = dataset_uninsured_population_2018[, which(!grepl("Wyoming", colnames(dataset_uninsured_population_2018)))]
View(dataset_uninsured_population_2018)

dataset_uninsured_population_2019 = dataset_uninsured_population_2019[, which(!grepl("Wyoming", colnames(dataset_uninsured_population_2019)))]
View(dataset_uninsured_population_2019)

dataset_uninsured_population_2021 = dataset_uninsured_population_2021[, which(!grepl("Wyoming", colnames(dataset_uninsured_population_2021)))]
View(dataset_uninsured_population_2021)

dataset_uninsured_population_2022 = dataset_uninsured_population_2022[, which(!grepl("Wyoming", colnames(dataset_uninsured_population_2022)))]
View(dataset_uninsured_population_2022)

dataset_uninsured_population_2023 = dataset_uninsured_population_2023[, which(!grepl("Wyoming", colnames(dataset_uninsured_population_2023)))]
View(dataset_uninsured_population_2023)

#####

#Now we are cleaning out the unnecessary metadata in the rows
#this is me cleaning out the rows with unnecessary metadata and keeping household income levels and the
dataset_uninsured_population_2010 <- dataset_uninsured_population_2010[-c(2:104),]
View(dataset_uninsured_population_2010)

dataset_uninsured_population_2011 <- dataset_uninsured_population_2011[-c(2:104),]
View(dataset_uninsured_population_2011)

dataset_uninsured_population_2012 <- dataset_uninsured_population_2012[-c(2:104),]
View(dataset_uninsured_population_2012)

dataset_uninsured_population_2013 <- dataset_uninsured_population_2013[-c(2:104),]
View(dataset_uninsured_population_2013)

dataset_uninsured_population_2014 <- dataset_uninsured_population_2014[-c(2:104),]

```



```

View(dataset_uninsured_population_2014)

dataset_uninsured_population_2015 <- dataset_uninsured_population_2015[-c(2:104),]
View(dataset_uninsured_population_2015)

dataset_uninsured_population_2016 <- dataset_uninsured_population_2016[-c(2:104),]
View(dataset_uninsured_population_2016)

dataset_uninsured_population_2017 <- dataset_uninsured_population_2017[-c(2:104),]
View(dataset_uninsured_population_2017)

dataset_uninsured_population_2018 <- dataset_uninsured_population_2018[-c(2:104),]
View(dataset_uninsured_population_2018)

dataset_uninsured_population_2019 <- dataset_uninsured_population_2019[-c(2:104),]
View(dataset_uninsured_population_2019)

dataset_uninsured_population_2021 <- dataset_uninsured_population_2021[-c(2:104),]
View(dataset_uninsured_population_2021)

dataset_uninsured_population_2022 <- dataset_uninsured_population_2022[-c(2:104),]
View(dataset_uninsured_population_2022)

dataset_uninsured_population_2023 <- dataset_uninsured_population_2023[-c(2:104),]
View(dataset_uninsured_population_2023)

dataset_uninsured_population_2010 <- dataset_uninsured_population_2010[-c(10:17),]
View(dataset_uninsured_population_2010)

dataset_uninsured_population_2011 <- dataset_uninsured_population_2011[-c(10:17),]
View(dataset_uninsured_population_2011)

dataset_uninsured_population_2012 <- dataset_uninsured_population_2012[-c(10:17),]
View(dataset_uninsured_population_2012)

dataset_uninsured_population_2013 <- dataset_uninsured_population_2013[-c(10:17),]
View(dataset_uninsured_population_2013)

dataset_uninsured_population_2014 <- dataset_uninsured_population_2014[-c(10:17),]
View(dataset_uninsured_population_2014)

dataset_uninsured_population_2015 <- dataset_uninsured_population_2015[-c(10:17),]
View(dataset_uninsured_population_2015)

dataset_uninsured_population_2016 <- dataset_uninsured_population_2016[-c(10:17),]
View(dataset_uninsured_population_2016)

dataset_uninsured_population_2017 <- dataset_uninsured_population_2017[-c(10:17),]
View(dataset_uninsured_population_2017)

dataset_uninsured_population_2018 <- dataset_uninsured_population_2018[-c(10:17),]
View(dataset_uninsured_population_2018)

```



```
dataset_uninsured_population_2019 <- dataset_uninsured_population_2019[-c(10:17),]
View(dataset_uninsured_population_2019)
```

```
dataset_uninsured_population_2021 <- dataset_uninsured_population_2021[-c(10:17),]
View(dataset_uninsured_population_2021)
```

```
dataset_uninsured_population_2022 <- dataset_uninsured_population_2022[-c(10:17),]
View(dataset_uninsured_population_2022)
```

```
dataset_uninsured_population_2023 <- dataset_uninsured_population_2023[-c(10:17),]
View(dataset_uninsured_population_2023)
```

```
#####
```

```
#As the values in these yearly datasets are stored as characters, we are now going to coerce them into numeric
#The values in the datasets of the uninsured population are stored as characters. Here I am going to get rid of the
```

```
# Keep the first column as-is
first_col <- dataset_uninsured_population_2010[[1]]
```

```
# Clean and convert the rest of the columns
# Keep the first column as-is
first_col <- dataset_uninsured_population_2010[[1]]
```

```
# Clean and convert the rest of the columns
dataset_uninsured_population_2010_numeric <- as.data.frame(
  lapply(dataset_uninsured_population_2010[, -1], function(col) {
    cleaned <- gsub("[\\%,]", "", col)
    as.numeric(cleaned)
  }),
  stringsAsFactors = FALSE
)
```

```
#Combine back with the first column
dataset_uninsured_population_2010_cleaned <- cbind(first_col, dataset_uninsured_population_2010_numeric)
```

```
#rename the first column
colnames(dataset_uninsured_population_2010_cleaned)[1] <- colnames(dataset_uninsured_population_2010)[1]
```

```
View(dataset_uninsured_population_2010_cleaned)
class(dataset_uninsured_population_2010_cleaned)
```

```
## [1] "data.frame"
```

```
class(dataset_uninsured_population_2010_cleaned[[2]])
```

```
## [1] "numeric"
```

```
# Keep the first column as-is
first_col <- dataset_uninsured_population_2011[[1]]
```

```
# Clean and convert the rest of the columns
```

```

dataset_uninsured_population_2011_numeric <- as.data.frame(
  lapply(dataset_uninsured_population_2011[ , -1], function(col) {
    cleaned <- gsub("[\\%,]", "", col)
    as.numeric(cleaned)
  }),
  stringsAsFactors = FALSE
)

#Combine back with the first column
dataset_uninsured_population_2011_cleaned <- cbind(first_col, dataset_uninsured_population_2011_numeric)

#rename the first column
colnames(dataset_uninsured_population_2011_cleaned)[1] <- colnames(dataset_uninsured_population_2011)[1]

View(dataset_uninsured_population_2011_cleaned)
class(dataset_uninsured_population_2011_cleaned)

```

```
## [1] "data.frame"
```

```
class(dataset_uninsured_population_2011_cleaned[[2]])
```

```
## [1] "numeric"
```

```

# Keep the first column as-is
first_col <- dataset_uninsured_population_2012[[1]]

# Clean and convert the rest of the columns
dataset_uninsured_population_2012_numeric <- as.data.frame(
  lapply(dataset_uninsured_population_2012[ , -1], function(col) {
    cleaned <- gsub("[\\%,]", "", col)
    as.numeric(cleaned)
  }),
  stringsAsFactors = FALSE
)

#Combine back with the first column
dataset_uninsured_population_2012_cleaned <- cbind(first_col, dataset_uninsured_population_2012_numeric)

#rename the first column
colnames(dataset_uninsured_population_2012_cleaned)[1] <- colnames(dataset_uninsured_population_2012)[1]

View(dataset_uninsured_population_2012_cleaned)
class(dataset_uninsured_population_2012_cleaned)

```

```
## [1] "data.frame"
```

```
class(dataset_uninsured_population_2012_cleaned[[2]])
```

```
## [1] "numeric"
```

```

# Keep the first column as-is
first_col <- dataset_uninsured_population_2013[[1]]

# Clean and convert the rest of the columns
dataset_uninsured_population_2013_numeric <- as.data.frame(
  lapply(dataset_uninsured_population_2013[, -1], function(col) {
    cleaned <- gsub("[\\%,]", "", col)
    as.numeric(cleaned)
  }),
  stringsAsFactors = FALSE
)

#Combine back with the first column
dataset_uninsured_population_2013_cleaned <- cbind(first_col, dataset_uninsured_population_2013_numeric)

#rename the first column
colnames(dataset_uninsured_population_2013_cleaned)[1] <- colnames(dataset_uninsured_population_2013)[1]

View(dataset_uninsured_population_2013_cleaned)
class(dataset_uninsured_population_2013_cleaned)

## [1] "data.frame"

```

```

class(dataset_uninsured_population_2013_cleaned[[2]])

```

```

## [1] "numeric"

```

```

# Keep the first column as-is
first_col <- dataset_uninsured_population_2014[[1]]

# Clean and convert the rest of the columns
dataset_uninsured_population_2014_numeric <- as.data.frame(
  lapply(dataset_uninsured_population_2014[, -1], function(col) {
    cleaned <- gsub("[\\%,]", "", col)
    as.numeric(cleaned)
  }),
  stringsAsFactors = FALSE
)

#Combine back with the first column
dataset_uninsured_population_2014_cleaned <- cbind(first_col, dataset_uninsured_population_2014_numeric)

#rename the first column
colnames(dataset_uninsured_population_2014_cleaned)[1] <- colnames(dataset_uninsured_population_2014)[1]

View(dataset_uninsured_population_2014_cleaned)
class(dataset_uninsured_population_2014_cleaned)

## [1] "data.frame"

```

```
class(dataset_uninsured_population_2014_cleaned[[2]])
```

```
## [1] "numeric"
```

```
# Keep the first column as-is
```

```
first_col <- dataset_uninsured_population_2015[[1]]
```

```
# Clean and convert the rest of the columns
```

```
dataset_uninsured_population_2015_numeric <- as.data.frame(  
  lapply(dataset_uninsured_population_2015[, -1], function(col) {  
    cleaned <- gsub("\\%", "", col)  
    as.numeric(cleaned)  
  }),  
  stringsAsFactors = FALSE  
)
```

```
#Combine back with the first column
```

```
dataset_uninsured_population_2015_cleaned <- cbind(first_col, dataset_uninsured_population_2015_numeric)
```

```
#rename the first column
```

```
colnames(dataset_uninsured_population_2015_cleaned)[1] <- colnames(dataset_uninsured_population_2015)[1]
```

```
View(dataset_uninsured_population_2015_cleaned)
```

```
class(dataset_uninsured_population_2015_cleaned)
```

```
## [1] "data.frame"
```

```
class(dataset_uninsured_population_2015_cleaned[[2]])
```

```
## [1] "numeric"
```

```
# Keep the first column as-is
```

```
first_col <- dataset_uninsured_population_2016[[1]]
```

```
# Clean and convert the rest of the columns
```

```
dataset_uninsured_population_2016_numeric <- as.data.frame(  
  lapply(dataset_uninsured_population_2016[, -1], function(col) {  
    cleaned <- gsub("\\%", "", col)  
    as.numeric(cleaned)  
  }),  
  stringsAsFactors = FALSE  
)
```

```
#Combine back with the first column
```

```
dataset_uninsured_population_2016_cleaned <- cbind(first_col, dataset_uninsured_population_2016_numeric)
```

```
#rename the first column
```

```
colnames(dataset_uninsured_population_2016_cleaned)[1] <- colnames(dataset_uninsured_population_2016)[1]
```

```
View(dataset_uninsured_population_2016_cleaned)
```

```
class(dataset_uninsured_population_2016_cleaned)
```

```
## [1] "data.frame"
```

```
class(dataset_uninsured_population_2016_cleaned[[2]])
```

```
## [1] "numeric"
```

```
# Keep the first column as-is
```

```
first_col <- dataset_uninsured_population_2017[[1]]
```

```
# Clean and convert the rest of the columns
```

```
dataset_uninsured_population_2017_numeric <- as.data.frame(  
  lapply(dataset_uninsured_population_2017[, -1], function(col) {  
    cleaned <- gsub("\\%", "", col)  
    as.numeric(cleaned)  
  }),  
  stringsAsFactors = FALSE  
)
```

```
#Combine back with the first column
```

```
dataset_uninsured_population_2017_cleaned <- cbind(first_col, dataset_uninsured_population_2017_numeric)
```

```
#rename the first column
```

```
colnames(dataset_uninsured_population_2017_cleaned)[1] <- colnames(dataset_uninsured_population_2017)[1]
```

```
View(dataset_uninsured_population_2017_cleaned)
```

```
class(dataset_uninsured_population_2017_cleaned)
```

```
## [1] "data.frame"
```

```
class(dataset_uninsured_population_2017_cleaned[[2]])
```

```
## [1] "numeric"
```

```
# Keep the first column as-is
```

```
first_col <- dataset_uninsured_population_2018[[1]]
```

```
# Clean and convert the rest of the columns
```

```
dataset_uninsured_population_2018_numeric <- as.data.frame(  
  lapply(dataset_uninsured_population_2018[, -1], function(col) {  
    cleaned <- gsub("\\%", "", col)  
    as.numeric(cleaned)  
  }),  
  stringsAsFactors = FALSE  
)
```

```
#Combine back with the first column
```

```
dataset_uninsured_population_2018_cleaned <- cbind(first_col, dataset_uninsured_population_2018_numeric)
```

```
#rename the first column
```

```
colnames(dataset_uninsured_population_2018_cleaned)[1] <- colnames(dataset_uninsured_population_2018)[1]
```

```
View(dataset_uninsured_population_2018_cleaned)
```

```
class(dataset_uninsured_population_2018_cleaned)
```

```
## [1] "data.frame"
```

```
class(dataset_uninsured_population_2018_cleaned[[2]])
```

```
## [1] "numeric"
```

```
# Keep the first column as-is
```

```
first_col <- dataset_uninsured_population_2019[[1]]
```

```
# Clean and convert the rest of the columns
```

```
dataset_uninsured_population_2019_numeric <- as.data.frame(  
  lapply(dataset_uninsured_population_2019[, -1], function(col) {  
    cleaned <- gsub("\\%", "", col)  
    as.numeric(cleaned)  
  }),  
  stringsAsFactors = FALSE  
)
```

```
#Combine back with the first column
```

```
dataset_uninsured_population_2019_cleaned <- cbind(first_col, dataset_uninsured_population_2019_numeric)
```

```
#rename the first column
```

```
colnames(dataset_uninsured_population_2019_cleaned)[1] <- colnames(dataset_uninsured_population_2019)[1]
```

```
View(dataset_uninsured_population_2019_cleaned)
```

```
class(dataset_uninsured_population_2019_cleaned)
```

```
## [1] "data.frame"
```

```
class(dataset_uninsured_population_2019_cleaned[[2]])
```

```
## [1] "numeric"
```

```
# Keep the first column as-is
```

```
first_col <- dataset_uninsured_population_2021[[1]]
```

```
# Clean and convert the rest of the columns
```

```
dataset_uninsured_population_2021_numeric <- as.data.frame(  
  lapply(dataset_uninsured_population_2021[, -1], function(col) {  
    cleaned <- gsub("\\%", "", col)  
    as.numeric(cleaned)  
  }),  
  stringsAsFactors = FALSE  
)
```

```
#Combine back with the first column
```

```
dataset_uninsured_population_2021_cleaned <- cbind(first_col, dataset_uninsured_population_2021_numeric)
```

```
#rename the first column
```

```
colnames(dataset_uninsured_population_2021_cleaned)[1] <- colnames(dataset_uninsured_population_2021)[1]
```

```
View(dataset_uninsured_population_2021_cleaned)
```

```
class(dataset_uninsured_population_2021_cleaned)
```

```
## [1] "data.frame"
```

```
class(dataset_uninsured_population_2021_cleaned[[2]])
```

```
## [1] "numeric"
```

```
# Keep the first column as-is
```

```
first_col <- dataset_uninsured_population_2022[[1]]
```

```
# Clean and convert the rest of the columns
```

```
dataset_uninsured_population_2022_numeric <- as.data.frame(  
  lapply(dataset_uninsured_population_2022[, -1], function(col) {  
    cleaned <- gsub("\\%", "", col)  
    as.numeric(cleaned)  
  }),  
  stringsAsFactors = FALSE  
)
```

```
#Combine back with the first column
```

```
dataset_uninsured_population_2022_cleaned <- cbind(first_col, dataset_uninsured_population_2022_numeric)
```

```
#rename the first column
```

```
colnames(dataset_uninsured_population_2022_cleaned)[1] <- colnames(dataset_uninsured_population_2022)[1]
```

```
View(dataset_uninsured_population_2022_cleaned)
```

```
class(dataset_uninsured_population_2022_cleaned)
```

```
## [1] "data.frame"
```

```
class(dataset_uninsured_population_2022_cleaned[[2]])
```

```
## [1] "numeric"
```

```
# Keep the first column as-is
```

```
first_col <- dataset_uninsured_population_2023[[1]]
```

```
# Clean and convert the rest of the columns
```

```
dataset_uninsured_population_2023_numeric <- as.data.frame(  
  lapply(dataset_uninsured_population_2023[, -1], function(col) {  
    cleaned <- gsub("\\%", "", col)  
    as.numeric(cleaned)  
  }),  
  stringsAsFactors = FALSE  
)
```

```
#Combine back with the first column
```

```
dataset_uninsured_population_2023_cleaned <- cbind(first_col, dataset_uninsured_population_2023_numeric)
```

```
#rename the first column
```

```
colnames(dataset_uninsured_population_2023_cleaned)[1] <- colnames(dataset_uninsured_population_2023)[1]
```

```
View(dataset_uninsured_population_2023_cleaned)
```

```
class(dataset_uninsured_population_2023_cleaned)
```

```
## [1] "data.frame"
```

```
class(dataset_uninsured_population_2023_cleaned[[2]])
```

```
## [1] "numeric"
```

```
#####
```

```
#Now that the values in the yearly datasets are stored as numeric values but the first column as the fi
```

```
#Here I am trying to compute and create a new variable, which is the share of uninsured population per
```

```
# Copy original data 2010
```

```
df_pop_2010 <- dataset_uninsured_population_2010_cleaned
```

```
col_names <- colnames(df_pop_2010)
```

```
new_col_order <- col_names # to control column ordering
```

```
# Loop through all columns ending in "..Uninsured.Population..Estimate"
```

```
uninsured_cols <- grep("\\\\.\\.Uninsured\\.\\.Population\\.\\.\\.Estimate$", col_names, value = TRUE)
```

```
for (uninsured_col in uninsured_cols) {
```

```
  # Extract the state name - everything before the first ".."
```

```
  state <- strsplit(uninsured_col, "\\\\.\\.\\.")[[1]][1]
```

```
  # Define the total population column and the share column
```

```
  total_col <- paste0(state, "..Total.Civilian.Noninstitutionalized.Population..Estimate")
```

```
  share_col <- paste0(state, "..Uninsured.Share")
```

```
  cat("Processing:", state, "\n")
```

```
  if (total_col %in% col_names) {
```

```
    # Initialize new share column with NA
```

```
    df_pop_2010[[share_col]] <- NA
```

```
    # Calculate only for first row
```

```
    if (!is.na(df_pop_2010[[total_col]][1]) && df_pop_2010[[total_col]][1] != 0) {
```

```
      df_pop_2010[[share_col]][1] <- df_pop_2010[[uninsured_col]][1] / df_pop_2010[[total_col]][1]
```

```
    } else {
```

```
      df_pop_2010[[share_col]][1] <- NA
```

```
    }
```

```
    # Insert new column right after the uninsured column
```

```
    pos <- match(uninsured_col, new_col_order)
```

```
    new_col_order <- append(new_col_order, share_col, after = pos)
```

```
  } else {
```

```
    cat(" Total population column not found for:", state, "\n")
```

```
  }
```

```
}
```

```
## Processing: Alabama
```

```
## Processing: Alaska
```

```
## Processing: Arizona
```



```
## Processing: Arkansas
## Processing: California
## Processing: Colorado
## Processing: Connecticut
## Processing: District.of.Columbia
## Processing: Florida
## Processing: Georgia
## Processing: Idaho
## Processing: Illinois
## Processing: Indiana
## Processing: Iowa
## Processing: Kansas
## Processing: Kentucky
## Processing: Louisiana
## Processing: Maine
## Processing: Maryland
## Processing: Massachusetts
## Processing: Michigan
## Processing: Minnesota
## Processing: Mississippi
## Processing: Missouri
## Processing: Montana
## Processing: Nebraska
## Processing: Nevada
## Processing: New.Hampshire
## Processing: New.Jersey
## Processing: New.Mexico
## Processing: New.York
## Processing: North.Carolina
## Processing: Ohio
## Processing: Oklahoma
## Processing: Oregon
## Processing: Pennsylvania
## Processing: South.Carolina
## Processing: South.Dakota
## Processing: Tennessee
## Processing: Texas
## Processing: Utah
## Processing: Virginia
## Processing: Washington
## Processing: West.Virginia
## Processing: Wisconsin
```

```
# Reorder the columns
df_pop_2010 <- df_pop_2010[, new_col_order]

# Preview: Show share columns for the first row
share_cols <- grep("\\.\\.Uninsured.Share$", colnames(df_pop_2010), value = TRUE)
cat("\n Created share columns:\n")
```

```
##
## Created share columns:
```

```
print(share_cols)
```

```
## [1] "Alabama..Uninsured.Share"
## [2] "Alaska..Uninsured.Share"
## [3] "Arizona..Uninsured.Share"
## [4] "Arkansas..Uninsured.Share"
## [5] "California..Uninsured.Share"
## [6] "Colorado..Uninsured.Share"
## [7] "Connecticut..Uninsured.Share"
## [8] "District.of.Columbia..Uninsured.Share"
## [9] "Florida..Uninsured.Share"
## [10] "Georgia..Uninsured.Share"
## [11] "Idaho..Uninsured.Share"
## [12] "Illinois..Uninsured.Share"
## [13] "Indiana..Uninsured.Share"
## [14] "Iowa..Uninsured.Share"
## [15] "Kansas..Uninsured.Share"
## [16] "Kentucky..Uninsured.Share"
## [17] "Louisiana..Uninsured.Share"
## [18] "Maine..Uninsured.Share"
## [19] "Maryland..Uninsured.Share"
## [20] "Massachusetts..Uninsured.Share"
## [21] "Michigan..Uninsured.Share"
## [22] "Minnesota..Uninsured.Share"
## [23] "Mississippi..Uninsured.Share"
## [24] "Missouri..Uninsured.Share"
## [25] "Montana..Uninsured.Share"
## [26] "Nebraska..Uninsured.Share"
## [27] "Nevada..Uninsured.Share"
## [28] "New.Hampshire..Uninsured.Share"
## [29] "New.Jersey..Uninsured.Share"
## [30] "New.Mexico..Uninsured.Share"
## [31] "New.York..Uninsured.Share"
## [32] "North.Carolina..Uninsured.Share"
## [33] "Ohio..Uninsured.Share"
## [34] "Oklahoma..Uninsured.Share"
## [35] "Oregon..Uninsured.Share"
## [36] "Pennsylvania..Uninsured.Share"
## [37] "South.Carolina..Uninsured.Share"
## [38] "South.Dakota..Uninsured.Share"
## [39] "Tennessee..Uninsured.Share"
## [40] "Texas..Uninsured.Share"
## [41] "Utah..Uninsured.Share"
## [42] "Virginia..Uninsured.Share"
## [43] "Washington..Uninsured.Share"
## [44] "West.Virginia..Uninsured.Share"
## [45] "Wisconsin..Uninsured.Share"
```

```
print(df_pop_2010[1, share_cols])
```

```
## Alabama..Uninsured.Share Alaska..Uninsured.Share Arizona..Uninsured.Share
## 1 0.1459974 0.1985349 0.168796
```

```
## Arkansas..Uninsured.Share California..Uninsured.Share
## 1 0.1745749 0.1853812
## Colorado..Uninsured.Share Connecticut..Uninsured.Share
## 1 0.1591949 0.09094912
## District.of.Columbia..Uninsured.Share Florida..Uninsured.Share
## 1 NA 0.2126385
## Georgia..Uninsured.Share Idaho..Uninsured.Share Illinois..Uninsured.Share
## 1 0.1970166 0.1774928 0.1377355
## Indiana..Uninsured.Share Iowa..Uninsured.Share Kansas..Uninsured.Share
## 1 0.1483102 0.09311871 0.1388666
## Kentucky..Uninsured.Share Louisiana..Uninsured.Share Maine..Uninsured.Share
## 1 0.1525064 0.1781376 0.1013414
## Maryland..Uninsured.Share Massachusetts..Uninsured.Share
## 1 0.112669 0.04410529
## Michigan..Uninsured.Share Minnesota..Uninsured.Share
## 1 0.1236607 0.09067987
## Mississippi..Uninsured.Share Missouri..Uninsured.Share
## 1 0.1819194 0.1316177
## Montana..Uninsured.Share Nebraska..Uninsured.Share Nevada..Uninsured.Share
## 1 0.173214 0.1152316 0.2261925
## New.Hampshire..Uninsured.Share New.Jersey..Uninsured.Share
## 1 0.1113225 0.1324153
## New.Mexico..Uninsured.Share New.York..Uninsured.Share
## 1 0.1960733 0.119021
## North.Carolina..Uninsured.Share Ohio..Uninsured.Share
## 1 0.1677058 0.1231611
## Oklahoma..Uninsured.Share Oregon..Uninsured.Share
## 1 0.1886081 0.1714887
## Pennsylvania..Uninsured.Share South.Carolina..Uninsured.Share
## 1 0.1016456 0.1752929
## South.Dakota..Uninsured.Share Tennessee..Uninsured.Share
## 1 0.1238954 0.1435505
## Texas..Uninsured.Share Utah..Uninsured.Share Virginia..Uninsured.Share
## 1 0.2371108 0.1534072 0.1306617
## Washington..Uninsured.Share West.Virginia..Uninsured.Share
## 1 0.1424492 0.1459719
## Wisconsin..Uninsured.Share
## 1 0.09427438
```

```
View(df_pop_2010)
```

```
#####
```

```
# Copy original data 2011
```

```
df_pop_2011 <- dataset_uninsured_population_2011_cleaned
```

```
col_names <- colnames(df_pop_2011)
```

```
new_col_order <- col_names # to control column ordering
```

```
# Loop through all columns ending in "..Uninsured.Population..Estimate"
```

```
uninsured_cols <- grep("\\\\.\\.Uninsured\\.\\.Population\\.\\.\\.Estimate$", col_names, value = TRUE)
```

```
for (uninsured_col in uninsured_cols) {
```

```
  # Extract the state name - everything before the first ".."
```

```
  state <- strsplit(uninsured_col, "\\\\.\\.")[[1]][1]
```

```

# Define the total population column and the share column
total_col <- paste0(state, "..Total.Civilian.Noninstitutionalized.Population..Estimate")
share_col <- paste0(state, "..Uninsured.Share")

cat("Processing:", state, "\n")

if (total_col %in% col_names) {
  # Initialize new share column with NA
  df_pop_2011[[share_col]] <- NA

  # Calculate only for first row
  if (!is.na(df_pop_2011[[total_col]][1]) && df_pop_2011[[total_col]][1] != 0) {
    df_pop_2011[[share_col]][1] <- df_pop_2011[[uninsured_col]][1] / df_pop_2011[[total_col]][1]
  } else {
    df_pop_2011[[share_col]][1] <- NA
  }

  # Insert new column right after the uninsured column
  pos <- match(uninsured_col, new_col_order)
  new_col_order <- append(new_col_order, share_col, after = pos)

} else {
  cat(" Total population column not found for:", state, "\n")
}
}

```

```

## Processing: Alabama
## Processing: Alaska
## Processing: Arizona
## Processing: Arkansas
## Processing: California
## Processing: Colorado
## Processing: Connecticut
## Processing: Florida
## Processing: Georgia
## Processing: Idaho
## Processing: Illinois
## Processing: Indiana
## Processing: Iowa
## Processing: Kansas
## Processing: Kentucky
## Processing: Louisiana
## Processing: Maine
## Processing: Maryland
## Processing: Massachusetts
## Processing: Michigan
## Processing: Minnesota
## Processing: Mississippi
## Processing: Missouri
## Processing: Montana
## Processing: Nebraska
## Processing: Nevada
## Processing: New.Hampshire

```

```
## Processing: New.Jersey
## Processing: New.Mexico
## Processing: New.York
## Processing: North.Carolina
## Processing: Ohio
## Processing: Oklahoma
## Processing: Oregon
## Processing: Pennsylvania
## Processing: South.Carolina
## Processing: South.Dakota
## Processing: Tennessee
## Processing: Texas
## Processing: Utah
## Processing: Virginia
## Processing: Washington
## Processing: West.Virginia
## Processing: Wisconsin
```

```
# Reorder the columns
df_pop_2011 <- df_pop_2011[, new_col_order]

# Preview: Show share columns for the first row
share_cols <- grep("\\.\\.Uninsured.Share$", colnames(df_pop_2011), value = TRUE)
cat("\n Created share columns:\n")
```

```
##
## Created share columns:
```

```
print(share_cols)
```

```
## [1] "Alabama..Uninsured.Share" "Alaska..Uninsured.Share"
## [3] "Arizona..Uninsured.Share" "Arkansas..Uninsured.Share"
## [5] "California..Uninsured.Share" "Colorado..Uninsured.Share"
## [7] "Connecticut..Uninsured.Share" "Florida..Uninsured.Share"
## [9] "Georgia..Uninsured.Share" "Idaho..Uninsured.Share"
## [11] "Illinois..Uninsured.Share" "Indiana..Uninsured.Share"
## [13] "Iowa..Uninsured.Share" "Kansas..Uninsured.Share"
## [15] "Kentucky..Uninsured.Share" "Louisiana..Uninsured.Share"
## [17] "Maine..Uninsured.Share" "Maryland..Uninsured.Share"
## [19] "Massachusetts..Uninsured.Share" "Michigan..Uninsured.Share"
## [21] "Minnesota..Uninsured.Share" "Mississippi..Uninsured.Share"
## [23] "Missouri..Uninsured.Share" "Montana..Uninsured.Share"
## [25] "Nebraska..Uninsured.Share" "Nevada..Uninsured.Share"
## [27] "New.Hampshire..Uninsured.Share" "New.Jersey..Uninsured.Share"
## [29] "New.Mexico..Uninsured.Share" "New.York..Uninsured.Share"
## [31] "North.Carolina..Uninsured.Share" "Ohio..Uninsured.Share"
## [33] "Oklahoma..Uninsured.Share" "Oregon..Uninsured.Share"
## [35] "Pennsylvania..Uninsured.Share" "South.Carolina..Uninsured.Share"
## [37] "South.Dakota..Uninsured.Share" "Tennessee..Uninsured.Share"
## [39] "Texas..Uninsured.Share" "Utah..Uninsured.Share"
## [41] "Virginia..Uninsured.Share" "Washington..Uninsured.Share"
## [43] "West.Virginia..Uninsured.Share" "Wisconsin..Uninsured.Share"
```

```
print(df_pop_2011[1, share_cols])
```

```
## Alabama..Uninsured.Share Alaska..Uninsured.Share Arizona..Uninsured.Share
## 1 0.142787 0.2011521 0.1716779
## Arkansas..Uninsured.Share California..Uninsured.Share
## 1 0.1706322 0.1809817
## Colorado..Uninsured.Share Connecticut..Uninsured.Share
## 1 0.1511714 0.08770097
## Florida..Uninsured.Share Georgia..Uninsured.Share Idaho..Uninsured.Share
## 1 0.2085815 0.1957182 0.1654871
## Illinois..Uninsured.Share Indiana..Uninsured.Share Iowa..Uninsured.Share
## 1 0.1307751 0.1450422 0.08931007
## Kansas..Uninsured.Share Kentucky..Uninsured.Share Louisiana..Uninsured.Share
## 1 0.1260445 0.1443526 0.1752746
## Maine..Uninsured.Share Maryland..Uninsured.Share
## 1 0.1066606 0.1043205
## Massachusetts..Uninsured.Share Michigan..Uninsured.Share
## 1 0.04255642 0.1177909
## Minnesota..Uninsured.Share Mississippi..Uninsured.Share
## 1 0.08841631 0.1773511
## Missouri..Uninsured.Share Montana..Uninsured.Share Nebraska..Uninsured.Share
## 1 0.1368738 0.1826408 0.1141399
## Nevada..Uninsured.Share New.Hampshire..Uninsured.Share
## 1 0.2191278 0.1050122
## New.Jersey..Uninsured.Share New.Mexico..Uninsured.Share
## 1 0.1307364 0.1984055
## New.York..Uninsured.Share North.Carolina..Uninsured.Share
## 1 0.114291 0.1632918
## Ohio..Uninsured.Share Oklahoma..Uninsured.Share Oregon..Uninsured.Share
## 1 0.1191326 0.1869208 0.1571703
## Pennsylvania..Uninsured.Share South.Carolina..Uninsured.Share
## 1 0.1008431 0.1673434
## South.Dakota..Uninsured.Share Tennessee..Uninsured.Share
## 1 0.1190657 0.1463977
## Texas..Uninsured.Share Utah..Uninsured.Share Virginia..Uninsured.Share
## 1 0.229779 0.1527304 0.1246153
## Washington..Uninsured.Share West.Virginia..Uninsured.Share
## 1 0.1418275 0.1490771
## Wisconsin..Uninsured.Share
## 1 0.08993048
```

```
View(df_pop_2011)
```

```
#####
```

```
# Copy original data 2012
```

```
df_pop_2012 <- dataset_uninsured_population_2012_cleaned
```

```
col_names <- colnames(df_pop_2012)
```

```
new_col_order <- col_names # to control column ordering
```

```
# Loop through all columns ending in "..Uninsured.Population..Estimate"
```

```

uninsured_cols <- grep("\\.\\.Uninsured\\.Population\\.\\.\\.Estimate$", col_names, value = TRUE)

for (uninsured_col in uninsured_cols) {
  # Extract the state name - everything before the first "."
  state <- strsplit(uninsured_col, "\\\\.\\.\\.")[[1]][1]

  # Define the total population column and the share column
  total_col <- paste0(state, "..Total.Civilian.Noninstitutionalized.Population..Estimate")
  share_col <- paste0(state, "..Uninsured.Share")

  cat("Processing:", state, "\n")

  if (total_col %in% col_names) {
    # Initialize new share column with NA
    df_pop_2012[[share_col]] <- NA

    # Calculate only for first row
    if (!is.na(df_pop_2012[[total_col]][1]) && df_pop_2012[[total_col]][1] != 0) {
      df_pop_2012[[share_col]][1] <- df_pop_2012[[uninsured_col]][1] / df_pop_2012[[total_col]][1]
    } else {
      df_pop_2012[[share_col]][1] <- NA
    }

    # Insert new column right after the uninsured column
    pos <- match(uninsured_col, new_col_order)
    new_col_order <- append(new_col_order, share_col, after = pos)

  } else {
    cat(" Total population column not found for:", state, "\n")
  }
}

```

```

## Processing: Alabama
## Processing: Alaska
## Processing: Arizona
## Processing: Arkansas
## Processing: California
## Processing: Colorado
## Processing: Connecticut
## Processing: Florida
## Processing: Georgia
## Processing: Idaho
## Processing: Illinois
## Processing: Indiana
## Processing: Iowa
## Processing: Kansas
## Processing: Kentucky
## Processing: Louisiana
## Processing: Maine
## Processing: Maryland
## Processing: Massachusetts
## Processing: Michigan
## Processing: Minnesota

```

```
## Processing: Mississippi
## Processing: Missouri
## Processing: Montana
## Processing: Nebraska
## Processing: Nevada
## Processing: New.Hampshire
## Processing: New.Jersey
## Processing: New.Mexico
## Processing: New.York
## Processing: North.Carolina
## Processing: Ohio
## Processing: Oklahoma
## Processing: Oregon
## Processing: Pennsylvania
## Processing: South.Carolina
## Processing: South.Dakota
## Processing: Tennessee
## Processing: Texas
## Processing: Utah
## Processing: Virginia
## Processing: Washington
## Processing: West.Virginia
## Processing: Wisconsin
```

```
# Reorder the columns
```

```
df_pop_2012 <- df_pop_2012[, new_col_order]
```

```
# Preview: Show share columns for the first row
```

```
share_cols <- grep("\\\\.\\.Uninsured.Share$", colnames(df_pop_2012), value = TRUE)
cat("\n Created share columns:\n")
```

```
##
## Created share columns:
```

```
print(share_cols)
```

```
## [1] "Alabama..Uninsured.Share"      "Alaska..Uninsured.Share"
## [3] "Arizona..Uninsured.Share"      "Arkansas..Uninsured.Share"
## [5] "California..Uninsured.Share"    "Colorado..Uninsured.Share"
## [7] "Connecticut..Uninsured.Share"   "Florida..Uninsured.Share"
## [9] "Georgia..Uninsured.Share"       "Idaho..Uninsured.Share"
## [11] "Illinois..Uninsured.Share"      "Indiana..Uninsured.Share"
## [13] "Iowa..Uninsured.Share"          "Kansas..Uninsured.Share"
## [15] "Kentucky..Uninsured.Share"      "Louisiana..Uninsured.Share"
## [17] "Maine..Uninsured.Share"         "Maryland..Uninsured.Share"
## [19] "Massachusetts..Uninsured.Share" "Michigan..Uninsured.Share"
## [21] "Minnesota..Uninsured.Share"     "Mississippi..Uninsured.Share"
## [23] "Missouri..Uninsured.Share"      "Montana..Uninsured.Share"
## [25] "Nebraska..Uninsured.Share"      "Nevada..Uninsured.Share"
## [27] "New.Hampshire..Uninsured.Share" "New.Jersey..Uninsured.Share"
## [29] "New.Mexico..Uninsured.Share"    "New.York..Uninsured.Share"
## [31] "North.Carolina..Uninsured.Share" "Ohio..Uninsured.Share"
## [33] "Oklahoma..Uninsured.Share"      "Oregon..Uninsured.Share"
```



```
## [35] "Pennsylvania..Uninsured.Share" "South.Carolina..Uninsured.Share"
## [37] "South.Dakota..Uninsured.Share" "Tennessee..Uninsured.Share"
## [39] "Texas..Uninsured.Share" "Utah..Uninsured.Share"
## [41] "Virginia..Uninsured.Share" "Washington..Uninsured.Share"
## [43] "West.Virginia..Uninsured.Share" "Wisconsin..Uninsured.Share"
```

```
print(df_pop_2012[1, share_cols])
```

```
## Alabama..Uninsured.Share Alaska..Uninsured.Share Arizona..Uninsured.Share
## 1 0.1333908 0.2049352 0.1755152
## Arkansas..Uninsured.Share California..Uninsured.Share
## 1 0.1644548 0.1788066
## Colorado..Uninsured.Share Connecticut..Uninsured.Share
## 1 0.1474541 0.09109101
## Florida..Uninsured.Share Georgia..Uninsured.Share Idaho..Uninsured.Share
## 1 0.2007167 0.1844061 0.1616716
## Illinois..Uninsured.Share Indiana..Uninsured.Share Iowa..Uninsured.Share
## 1 0.1277695 0.1428998 0.08395062
## Kansas..Uninsured.Share Kentucky..Uninsured.Share Louisiana..Uninsured.Share
## 1 0.1258448 0.138714 0.1688479
## Maine..Uninsured.Share Maryland..Uninsured.Share
## 1 0.1023856 0.1032367
## Massachusetts..Uninsured.Share Michigan..Uninsured.Share
## 1 0.03872404 0.1140256
## Minnesota..Uninsured.Share Mississippi..Uninsured.Share
## 1 0.07982694 0.1704733
## Missouri..Uninsured.Share Montana..Uninsured.Share Nebraska..Uninsured.Share
## 1 0.1356255 0.1802464 0.1125484
## Nevada..Uninsured.Share New.Hampshire..Uninsured.Share
## 1 0.2215245 0.1063088
## New.Jersey..Uninsured.Share New.Mexico..Uninsured.Share
## 1 0.1270832 0.1842212
## New.York..Uninsured.Share North.Carolina..Uninsured.Share
## 1 0.1088647 0.1656357
## Ohio..Uninsured.Share Oklahoma..Uninsured.Share Oregon..Uninsured.Share
## 1 0.1146917 0.183516 0.1492077
## Pennsylvania..Uninsured.Share South.Carolina..Uninsured.Share
## 1 0.09750922 0.1683219
## South.Dakota..Uninsured.Share Tennessee..Uninsured.Share
## 1 0.115299 0.1388153
## Texas..Uninsured.Share Utah..Uninsured.Share Virginia..Uninsured.Share
## 1 0.2252285 0.1446344 0.1253827
## Washington..Uninsured.Share West.Virginia..Uninsured.Share
## 1 0.1390785 0.1444332
## Wisconsin..Uninsured.Share
## 1 0.08952413
```

```
#####
```

```
# Copy original data 2013
```

```
df_pop_2013 <- dataset_uninsured_population_2013_cleaned
col_names <- colnames(df_pop_2013)
```

```

new_col_order <- col_names # to control column ordering

# Loop through all columns ending in "..Uninsured.Population..Estimate"
uninsured_cols <- grep("\\\\.\\.Uninsured\\.\\.Population\\.\\.\\.Estimate$", col_names, value = TRUE)

for (uninsured_col in uninsured_cols) {
  # Extract the state name - everything before the first "."
  state <- strsplit(uninsured_col, "\\\\.\\.\\.")[1][1]

  # Define the total population column and the share column
  total_col <- paste0(state, "..Total.Civilian.Noninstitutionalized.Population..Estimate")
  share_col <- paste0(state, "..Uninsured.Share")

  cat("Processing:", state, "\n")

  if (total_col %in% col_names) {
    # Initialize new share column with NA
    df_pop_2013[[share_col]] <- NA

    # Calculate only for first row
    if (!is.na(df_pop_2013[[total_col]][1]) && df_pop_2013[[total_col]][1] != 0) {
      df_pop_2013[[share_col]][1] <- df_pop_2013[[uninsured_col]][1] / df_pop_2013[[total_col]][1]
    } else {
      df_pop_2013[[share_col]][1] <- NA
    }

    # Insert new column right after the uninsured column
    pos <- match(uninsured_col, new_col_order)
    new_col_order <- append(new_col_order, share_col, after = pos)

  } else {
    cat(" Total population column not found for:", state, "\n")
  }
}

```

```

## Processing: Alabama
## Processing: Alaska
## Processing: Arizona
## Processing: Arkansas
## Processing: California
## Processing: Colorado
## Processing: Connecticut
## Processing: Florida
## Processing: Georgia
## Processing: Idaho
## Processing: Illinois
## Processing: Indiana
## Processing: Iowa
## Processing: Kansas
## Processing: Kentucky
## Processing: Louisiana
## Processing: Maine
## Processing: Maryland

```

```
## Processing: Massachusetts
## Processing: Michigan
## Processing: Minnesota
## Processing: Mississippi
## Processing: Missouri
## Processing: Montana
## Processing: Nebraska
## Processing: Nevada
## Processing: New.Hampshire
## Processing: New.Jersey
## Processing: New.Mexico
## Processing: New.York
## Processing: North.Carolina
## Processing: Ohio
## Processing: Oklahoma
## Processing: Oregon
## Processing: Pennsylvania
## Processing: South.Carolina
## Processing: South.Dakota
## Processing: Tennessee
## Processing: Texas
## Processing: Utah
## Processing: Virginia
## Processing: Washington
## Processing: West.Virginia
## Processing: Wisconsin
```

```
# Reorder the columns
df_pop_2013 <- df_pop_2013[, new_col_order]

# Preview: Show share columns for the first row
share_cols <- grep("\\.\\.Uninsured.Share$", colnames(df_pop_2013), value = TRUE)
cat("\n Created share columns:\n")
```

```
##
## Created share columns:
```

```
print(share_cols)
```

```
## [1] "Alabama..Uninsured.Share"      "Alaska..Uninsured.Share"
## [3] "Arizona..Uninsured.Share"      "Arkansas..Uninsured.Share"
## [5] "California..Uninsured.Share"    "Colorado..Uninsured.Share"
## [7] "Connecticut..Uninsured.Share"  "Florida..Uninsured.Share"
## [9] "Georgia..Uninsured.Share"      "Idaho..Uninsured.Share"
## [11] "Illinois..Uninsured.Share"     "Indiana..Uninsured.Share"
## [13] "Iowa..Uninsured.Share"         "Kansas..Uninsured.Share"
## [15] "Kentucky..Uninsured.Share"     "Louisiana..Uninsured.Share"
## [17] "Maine..Uninsured.Share"        "Maryland..Uninsured.Share"
## [19] "Massachusetts..Uninsured.Share" "Michigan..Uninsured.Share"
## [21] "Minnesota..Uninsured.Share"    "Mississippi..Uninsured.Share"
## [23] "Missouri..Uninsured.Share"     "Montana..Uninsured.Share"
## [25] "Nebraska..Uninsured.Share"     "Nevada..Uninsured.Share"
## [27] "New.Hampshire..Uninsured.Share" "New.Jersey..Uninsured.Share"
```

```
## [29] "New.Mexico..Uninsured.Share" "New.York..Uninsured.Share"
## [31] "North.Carolina..Uninsured.Share" "Ohio..Uninsured.Share"
## [33] "Oklahoma..Uninsured.Share" "Oregon..Uninsured.Share"
## [35] "Pennsylvania..Uninsured.Share" "South.Carolina..Uninsured.Share"
## [37] "South.Dakota..Uninsured.Share" "Tennessee..Uninsured.Share"
## [39] "Texas..Uninsured.Share" "Utah..Uninsured.Share"
## [41] "Virginia..Uninsured.Share" "Washington..Uninsured.Share"
## [43] "West.Virginia..Uninsured.Share" "Wisconsin..Uninsured.Share"
```

```
print(df_pop_2013[1, share_cols])
```

```
## Alabama..Uninsured.Share Alaska..Uninsured.Share Arizona..Uninsured.Share
## 1 0.1357273 0.185309 0.1714687
## Arkansas..Uninsured.Share California..Uninsured.Share
## 1 0.1599408 0.171819
## Colorado..Uninsured.Share Connecticut..Uninsured.Share
## 1 0.140953 0.09389003
## Florida..Uninsured.Share Georgia..Uninsured.Share Idaho..Uninsured.Share
## 1 0.2002046 0.1883814 0.1617131
## Illinois..Uninsured.Share Indiana..Uninsured.Share Iowa..Uninsured.Share
## 1 0.1273713 0.13952 0.08134447
## Kansas..Uninsured.Share Kentucky..Uninsured.Share Louisiana..Uninsured.Share
## 1 0.1227105 0.1429577 0.166115
## Maine..Uninsured.Share Maryland..Uninsured.Share
## 1 0.1118673 0.1016754
## Massachusetts..Uninsured.Share Michigan..Uninsured.Share
## 1 0.03729648 0.1095126
## Minnesota..Uninsured.Share Mississippi..Uninsured.Share
## 1 0.0820073 0.1708762
## Missouri..Uninsured.Share Montana..Uninsured.Share Nebraska..Uninsured.Share
## 1 0.1303111 0.1646484 0.113317
## Nevada..Uninsured.Share New.Hampshire..Uninsured.Share
## 1 0.2067596 0.1071724
## New.Jersey..Uninsured.Share New.Mexico..Uninsured.Share
## 1 0.1319768 0.1863834
## New.York..Uninsured.Share North.Carolina..Uninsured.Share
## 1 0.106676 0.1564487
## Ohio..Uninsured.Share Oklahoma..Uninsured.Share Oregon..Uninsured.Share
## 1 0.1103284 0.1765857 0.1466397
## Pennsylvania..Uninsured.Share South.Carolina..Uninsured.Share
## 1 0.09725265 0.1579377
## South.Dakota..Uninsured.Share Tennessee..Uninsured.Share
## 1 0.1128435 0.138657
## Texas..Uninsured.Share Utah..Uninsured.Share Virginia..Uninsured.Share
## 1 0.2212828 0.1399075 0.1229951
## Washington..Uninsured.Share West.Virginia..Uninsured.Share
## 1 0.1398558 0.1397775
## Wisconsin..Uninsured.Share
## 1 0.09133387
```

```
#####
```

```

# Copy original data 2014
df_pop_2014 <- dataset_uninsured_population_2014_cleaned
col_names <- colnames(df_pop_2014)
new_col_order <- col_names # to control column ordering

# Loop through all columns ending in "..Uninsured.Population..Estimate"
uninsured_cols <- grep("\\\\.\\.Uninsured\\.\\.Population\\.\\.\\.Estimate$", col_names, value = TRUE)

for (uninsured_col in uninsured_cols) {
  # Extract the state name - everything before the first ".."
  state <- strsplit(uninsured_col, "\\\\.\\.")[[1]][1]

  # Define the total population column and the share column
  total_col <- paste0(state, "..Total.Civilian.Noninstitutionalized.Population..Estimate")
  share_col <- paste0(state, "..Uninsured.Share")

  cat("Processing:", state, "\n")

  if (total_col %in% col_names) {
    # Initialize new share column with NA
    df_pop_2014[[share_col]] <- NA

    # Calculate only for first row
    if (!is.na(df_pop_2014[[total_col]][1]) && df_pop_2014[[total_col]][1] != 0) {
      df_pop_2014[[share_col]][1] <- df_pop_2014[[uninsured_col]][1] / df_pop_2014[[total_col]][1]
    } else {
      df_pop_2014[[share_col]][1] <- NA
    }

    # Insert new column right after the uninsured column
    pos <- match(uninsured_col, new_col_order)
    new_col_order <- append(new_col_order, share_col, after = pos)

  } else {
    cat(" Total population column not found for:", state, "\n")
  }
}

```

```

## Processing: Alabama
## Processing: Alaska
## Processing: Arizona
## Processing: Arkansas
## Processing: California
## Processing: Colorado
## Processing: Connecticut
## Processing: Florida
## Processing: Georgia
## Processing: Idaho
## Processing: Illinois
## Processing: Indiana
## Processing: Iowa
## Processing: Kansas
## Processing: Kentucky

```

```
## Processing: Louisiana
## Processing: Maine
## Processing: Maryland
## Processing: Massachusetts
## Processing: Michigan
## Processing: Minnesota
## Processing: Mississippi
## Processing: Missouri
## Processing: Montana
## Processing: Nebraska
## Processing: Nevada
## Processing: New.Hampshire
## Processing: New.Jersey
## Processing: New.Mexico
## Processing: New.York
## Processing: North.Carolina
## Processing: Ohio
## Processing: Oklahoma
## Processing: Oregon
## Processing: Pennsylvania
## Processing: South.Carolina
## Processing: South.Dakota
## Processing: Tennessee
## Processing: Texas
## Processing: Utah
## Processing: Virginia
## Processing: Washington
## Processing: West.Virginia
## Processing: Wisconsin
```

```
# Reorder the columns
df_pop_2014 <- df_pop_2014[, new_col_order]

# Preview: Show share columns for the first row
share_cols <- grep("\\.\\.Uninsured.Share$", colnames(df_pop_2014), value = TRUE)
cat("\n Created share columns:\n")
```

```
##
## Created share columns:
```

```
print(share_cols)
```

```
## [1] "Alabama..Uninsured.Share"      "Alaska..Uninsured.Share"
## [3] "Arizona..Uninsured.Share"      "Arkansas..Uninsured.Share"
## [5] "California..Uninsured.Share"    "Colorado..Uninsured.Share"
## [7] "Connecticut..Uninsured.Share"   "Florida..Uninsured.Share"
## [9] "Georgia..Uninsured.Share"       "Idaho..Uninsured.Share"
## [11] "Illinois..Uninsured.Share"      "Indiana..Uninsured.Share"
## [13] "Iowa..Uninsured.Share"          "Kansas..Uninsured.Share"
## [15] "Kentucky..Uninsured.Share"      "Louisiana..Uninsured.Share"
## [17] "Maine..Uninsured.Share"         "Maryland..Uninsured.Share"
## [19] "Massachusetts..Uninsured.Share" "Michigan..Uninsured.Share"
## [21] "Minnesota..Uninsured.Share"     "Mississippi..Uninsured.Share"
```

```
## [23] "Missouri..Uninsured.Share"      "Montana..Uninsured.Share"
## [25] "Nebraska..Uninsured.Share"      "Nevada..Uninsured.Share"
## [27] "New.Hampshire..Uninsured.Share"  "New.Jersey..Uninsured.Share"
## [29] "New.Mexico..Uninsured.Share"     "New.York..Uninsured.Share"
## [31] "North.Carolina..Uninsured.Share" "Ohio..Uninsured.Share"
## [33] "Oklahoma..Uninsured.Share"       "Oregon..Uninsured.Share"
## [35] "Pennsylvania..Uninsured.Share"   "South.Carolina..Uninsured.Share"
## [37] "South.Dakota..Uninsured.Share"   "Tennessee..Uninsured.Share"
## [39] "Texas..Uninsured.Share"          "Utah..Uninsured.Share"
## [41] "Virginia..Uninsured.Share"       "Washington..Uninsured.Share"
## [43] "West.Virginia..Uninsured.Share"  "Wisconsin..Uninsured.Share"
```

```
print(df_pop_2014[1, share_cols])
```

```
## Alabama..Uninsured.Share Alaska..Uninsured.Share Arizona..Uninsured.Share
## 1 0.1214857 0.1720716 0.136391
## Arkansas..Uninsured.Share California..Uninsured.Share
## 1 0.1177511 0.1244655
## Colorado..Uninsured.Share Connecticut..Uninsured.Share
## 1 0.10304 0.06929666
## Florida..Uninsured.Share Georgia..Uninsured.Share Idaho..Uninsured.Share
## 1 0.1657101 0.1583123 0.1355995
## Illinois..Uninsured.Share Indiana..Uninsured.Share Iowa..Uninsured.Share
## 1 0.09745612 0.119403 0.06166868
## Kansas..Uninsured.Share Kentucky..Uninsured.Share Louisiana..Uninsured.Share
## 1 0.1024364 0.08456971 0.1477168
## Maine..Uninsured.Share Maryland..Uninsured.Share
## 1 0.1014676 0.07870881
## Massachusetts..Uninsured.Share Michigan..Uninsured.Share
## 1 0.03280003 0.08539242
## Minnesota..Uninsured.Share Mississippi..Uninsured.Share
## 1 0.05877077 0.1448983
## Missouri..Uninsured.Share Montana..Uninsured.Share Nebraska..Uninsured.Share
## 1 0.1166063 0.1421672 0.09667559
## Nevada..Uninsured.Share New.Hampshire..Uninsured.Share
## 1 0.1520756 0.0918165
## New.Jersey..Uninsured.Share New.Mexico..Uninsured.Share
## 1 0.1093211 0.1452173
## New.York..Uninsured.Share North.Carolina..Uninsured.Share
## 1 0.08700504 0.1308485
## Ohio..Uninsured.Share Oklahoma..Uninsured.Share Oregon..Uninsured.Share
## 1 0.08363724 0.153638 0.09733686
## Pennsylvania..Uninsured.Share South.Carolina..Uninsured.Share
## 1 0.08460563 0.1355285
## South.Dakota..Uninsured.Share Tennessee..Uninsured.Share
## 1 0.09807362 0.1203831
## Texas..Uninsured.Share Utah..Uninsured.Share Virginia..Uninsured.Share
## 1 0.1905727 0.1253378 0.1088908
## Washington..Uninsured.Share West.Virginia..Uninsured.Share
## 1 0.0924043 0.08572436
## Wisconsin..Uninsured.Share
## 1 0.07348165
```

```
#####

# Copy original data 2015
df_pop_2015 <- dataset_uninsured_population_2015_cleaned
col_names <- colnames(df_pop_2015)
new_col_order <- col_names # to control column ordering

# Loop through all columns ending in "..Uninsured.Population..Estimate"
uninsured_cols <- grep("\\.\\.Uninsured\\.\\.Population\\.\\.Estimate$", col_names, value = TRUE)

for (uninsured_col in uninsured_cols) {
  # Extract the state name - everything before the first ".."
  state <- strsplit(uninsured_col, "\\..\\.")[[1]][1]

  # Define the total population column and the share column
  total_col <- paste0(state, "..Total.Civilian.Noninstitutionalized.Population..Estimate")
  share_col <- paste0(state, "..Uninsured.Share")

  cat("Processing:", state, "\n")

  if (total_col %in% col_names) {
    # Initialize new share column with NA
    df_pop_2015[[share_col]] <- NA

    # Calculate only for first row
    if (!is.na(df_pop_2015[[total_col]][1]) && df_pop_2015[[total_col]][1] != 0) {
      df_pop_2015[[share_col]][1] <- df_pop_2015[[uninsured_col]][1] / df_pop_2015[[total_col]][1]
    } else {
      df_pop_2015[[share_col]][1] <- NA
    }

    # Insert new column right after the uninsured column
    pos <- match(uninsured_col, new_col_order)
    new_col_order <- append(new_col_order, share_col, after = pos)

  } else {
    cat(" Total population column not found for:", state, "\n")
  }
}

```

```
## Processing: Alabama
## Processing: Alaska
## Processing: Arizona
## Processing: Arkansas
## Processing: California
## Processing: Colorado
## Processing: Connecticut
## Processing: Florida
## Processing: Georgia
## Processing: Idaho
## Processing: Illinois
## Processing: Indiana

```



```
## Processing: Iowa
## Processing: Kansas
## Processing: Kentucky
## Processing: Louisiana
## Processing: Maine
## Processing: Maryland
## Processing: Massachusetts
## Processing: Michigan
## Processing: Minnesota
## Processing: Mississippi
## Processing: Missouri
## Processing: Montana
## Processing: Nebraska
## Processing: Nevada
## Processing: New.Hampshire
## Processing: New.Jersey
## Processing: New.Mexico
## Processing: New.York
## Processing: North.Carolina
## Processing: Ohio
## Processing: Oklahoma
## Processing: Oregon
## Processing: Pennsylvania
## Processing: South.Carolina
## Processing: South.Dakota
## Processing: Tennessee
## Processing: Texas
## Processing: Utah
## Processing: Virginia
## Processing: Washington
## Processing: West.Virginia
## Processing: Wisconsin
```

```
# Reorder the columns
df_pop_2015 <- df_pop_2015[, new_col_order]

# Preview: Show share columns for the first row
share_cols <- grep("\\.\\..Uninsured.Share$", colnames(df_pop_2015), value = TRUE)
cat("\n Created share columns:\n")
```

```
##
## Created share columns:
```

```
print(share_cols)
```

```
## [1] "Alabama..Uninsured.Share" "Alaska..Uninsured.Share"
## [3] "Arizona..Uninsured.Share" "Arkansas..Uninsured.Share"
## [5] "California..Uninsured.Share" "Colorado..Uninsured.Share"
## [7] "Connecticut..Uninsured.Share" "Florida..Uninsured.Share"
## [9] "Georgia..Uninsured.Share" "Idaho..Uninsured.Share"
## [11] "Illinois..Uninsured.Share" "Indiana..Uninsured.Share"
## [13] "Iowa..Uninsured.Share" "Kansas..Uninsured.Share"
## [15] "Kentucky..Uninsured.Share" "Louisiana..Uninsured.Share"
```

```
## [17] "Maine..Uninsured.Share"      "Maryland..Uninsured.Share"
## [19] "Massachusetts..Uninsured.Share" "Michigan..Uninsured.Share"
## [21] "Minnesota..Uninsured.Share"   "Mississippi..Uninsured.Share"
## [23] "Missouri..Uninsured.Share"    "Montana..Uninsured.Share"
## [25] "Nebraska..Uninsured.Share"    "Nevada..Uninsured.Share"
## [27] "New.Hampshire..Uninsured.Share" "New.Jersey..Uninsured.Share"
## [29] "New.Mexico..Uninsured.Share"   "New.York..Uninsured.Share"
## [31] "North.Carolina..Uninsured.Share" "Ohio..Uninsured.Share"
## [33] "Oklahoma..Uninsured.Share"     "Oregon..Uninsured.Share"
## [35] "Pennsylvania..Uninsured.Share" "South.Carolina..Uninsured.Share"
## [37] "South.Dakota..Uninsured.Share" "Tennessee..Uninsured.Share"
## [39] "Texas..Uninsured.Share"        "Utah..Uninsured.Share"
## [41] "Virginia..Uninsured.Share"     "Washington..Uninsured.Share"
## [43] "West.Virginia..Uninsured.Share" "Wisconsin..Uninsured.Share"
```

```
print(df_pop_2015[1, share_cols])
```

```
## Alabama..Uninsured.Share Alaska..Uninsured.Share Arizona..Uninsured.Share
## 1          0.1012893          0.1486028          0.1083402
## Arkansas..Uninsured.Share California..Uninsured.Share
## 1          0.09491035          0.08582917
## Colorado..Uninsured.Share Connecticut..Uninsured.Share
## 1          0.08060276          0.0595034
## Florida..Uninsured.Share Georgia..Uninsured.Share Idaho..Uninsured.Share
## 1          0.133343          0.1385068          0.1104888
## Illinois..Uninsured.Share Indiana..Uninsured.Share Iowa..Uninsured.Share
## 1          0.07100149          0.09627842          0.05030822
## Kansas..Uninsured.Share Kentucky..Uninsured.Share Louisiana..Uninsured.Share
## 1          0.09140122          0.06020847          0.1194612
## Maine..Uninsured.Share Maryland..Uninsured.Share
## 1          0.08404615          0.06587079
## Massachusetts..Uninsured.Share Michigan..Uninsured.Share
## 1          0.02817363          0.06081482
## Minnesota..Uninsured.Share Mississippi..Uninsured.Share
## 1          0.04512386          0.1269262
## Missouri..Uninsured.Share Montana..Uninsured.Share Nebraska..Uninsured.Share
## 1          0.09768298          0.1164756          0.08239485
## Nevada..Uninsured.Share New.Hampshire..Uninsured.Share
## 1          0.1227896          0.06344916
## New.Jersey..Uninsured.Share New.Mexico..Uninsured.Share
## 1          0.0871367          0.1092739
## New.York..Uninsured.Share North.Carolina..Uninsured.Share
## 1          0.07059297          0.1118943
## Ohio..Uninsured.Share Oklahoma..Uninsured.Share Oregon..Uninsured.Share
## 1          0.06522235          0.1390244          0.07022141
## Pennsylvania..Uninsured.Share South.Carolina..Uninsured.Share
## 1          0.06364794          0.1089763
## South.Dakota..Uninsured.Share Tennessee..Uninsured.Share
## 1          0.1021141          0.102738
## Texas..Uninsured.Share Utah..Uninsured.Share Virginia..Uninsured.Share
## 1          0.1709948          0.1047932          0.09142927
## Washington..Uninsured.Share West.Virginia..Uninsured.Share
## 1          0.06622038          0.05953275
## Wisconsin..Uninsured.Share
```

```
## 1 0.05660928
```

```
#####

# Copy original data 2016
df_pop_2016 <- dataset_uninsured_population_2016_cleaned
col_names <- colnames(df_pop_2016)
new_col_order <- col_names # to control column ordering

# Loop through all columns ending in "..Uninsured.Population..Estimate"
uninsured_cols <- grep("\\\\.\\.Uninsured\\.\\.Population\\.\\.\\.Estimate$", col_names, value = TRUE)

for (uninsured_col in uninsured_cols) {
  # Extract the state name - everything before the first ".."
  state <- strsplit(uninsured_col, "\\\\.\\.")[[1]][1]

  # Define the total population column and the share column
  total_col <- paste0(state, "..Total.Civilian.Noninstitutionalized.Population..Estimate")
  share_col <- paste0(state, "..Uninsured.Share")

  cat("Processing:", state, "\n")

  if (total_col %in% col_names) {
    # Initialize new share column with NA
    df_pop_2016[[share_col]] <- NA

    # Calculate only for first row
    if (!is.na(df_pop_2016[[total_col]][1]) && df_pop_2016[[total_col]][1] != 0) {
      df_pop_2016[[share_col]][1] <- df_pop_2016[[uninsured_col]][1] / df_pop_2016[[total_col]][1]
    } else {
      df_pop_2016[[share_col]][1] <- NA
    }

    # Insert new column right after the uninsured column
    pos <- match(uninsured_col, new_col_order)
    new_col_order <- append(new_col_order, share_col, after = pos)

  } else {
    cat(" Total population column not found for:", state, "\n")
  }
}
}
```

```
## Processing: Alabama
## Processing: Alaska
## Processing: Arizona
## Processing: Arkansas
## Processing: California
## Processing: Colorado
## Processing: Connecticut
## Processing: Florida
## Processing: Georgia
## Processing: Idaho
## Processing: Illinois
```

```
## Processing: Indiana
## Processing: Iowa
## Processing: Kansas
## Processing: Kentucky
## Processing: Louisiana
## Processing: Maine
## Processing: Maryland
## Processing: Massachusetts
## Processing: Michigan
## Processing: Minnesota
## Processing: Mississippi
## Processing: Missouri
## Processing: Montana
## Processing: Nebraska
## Processing: Nevada
## Processing: New.Hampshire
## Processing: New.Jersey
## Processing: New.Mexico
## Processing: New.York
## Processing: North.Carolina
## Processing: Ohio
## Processing: Oklahoma
## Processing: Oregon
## Processing: Pennsylvania
## Processing: South.Carolina
## Processing: South.Dakota
## Processing: Tennessee
## Processing: Texas
## Processing: Utah
## Processing: Virginia
## Processing: Washington
## Processing: West.Virginia
## Processing: Wisconsin
```

```
# Reorder the columns
df_pop_2016 <- df_pop_2016[, new_col_order]

# Preview: Show share columns for the first row
share_cols <- grep("\\\\.\\.Uninsured.Share$", colnames(df_pop_2016), value = TRUE)
cat("\n Created share columns:\n")
```

```
##
## Created share columns:
```

```
print(share_cols)
```

```
## [1] "Alabama..Uninsured.Share" "Alaska..Uninsured.Share"
## [3] "Arizona..Uninsured.Share" "Arkansas..Uninsured.Share"
## [5] "California..Uninsured.Share" "Colorado..Uninsured.Share"
## [7] "Connecticut..Uninsured.Share" "Florida..Uninsured.Share"
## [9] "Georgia..Uninsured.Share" "Idaho..Uninsured.Share"
## [11] "Illinois..Uninsured.Share" "Indiana..Uninsured.Share"
## [13] "Iowa..Uninsured.Share" "Kansas..Uninsured.Share"
```

```
## [15] "Kentucky..Uninsured.Share"      "Louisiana..Uninsured.Share"
## [17] "Maine..Uninsured.Share"          "Maryland..Uninsured.Share"
## [19] "Massachusetts..Uninsured.Share"  "Michigan..Uninsured.Share"
## [21] "Minnesota..Uninsured.Share"      "Mississippi..Uninsured.Share"
## [23] "Missouri..Uninsured.Share"       "Montana..Uninsured.Share"
## [25] "Nebraska..Uninsured.Share"       "Nevada..Uninsured.Share"
## [27] "New.Hampshire..Uninsured.Share"  "New.Jersey..Uninsured.Share"
## [29] "New.Mexico..Uninsured.Share"     "New.York..Uninsured.Share"
## [31] "North.Carolina..Uninsured.Share" "Ohio..Uninsured.Share"
## [33] "Oklahoma..Uninsured.Share"       "Oregon..Uninsured.Share"
## [35] "Pennsylvania..Uninsured.Share"   "South.Carolina..Uninsured.Share"
## [37] "South.Dakota..Uninsured.Share"   "Tennessee..Uninsured.Share"
## [39] "Texas..Uninsured.Share"          "Utah..Uninsured.Share"
## [41] "Virginia..Uninsured.Share"       "Washington..Uninsured.Share"
## [43] "West.Virginia..Uninsured.Share"  "Wisconsin..Uninsured.Share"
```

```
print(df_pop_2016[1, share_cols])
```

```
## Alabama..Uninsured.Share Alaska..Uninsured.Share Arizona..Uninsured.Share
## 1 0.09095042 0.1400687 0.09982028
## Arkansas..Uninsured.Share California..Uninsured.Share
## 1 0.07898893 0.07336489
## Colorado..Uninsured.Share Connecticut..Uninsured.Share
## 1 0.07521312 0.0488529
## Florida..Uninsured.Share Georgia..Uninsured.Share Idaho..Uninsured.Share
## 1 0.1253665 0.1293944 0.1010965
## Illinois..Uninsured.Share Indiana..Uninsured.Share Iowa..Uninsured.Share
## 1 0.06470498 0.08115837 0.04257308
## Kansas..Uninsured.Share Kentucky..Uninsured.Share Louisiana..Uninsured.Share
## 1 0.08733288 0.05129318 0.1026466
## Maine..Uninsured.Share Maryland..Uninsured.Share
## 1 0.08045641 0.06136299
## Massachusetts..Uninsured.Share Michigan..Uninsured.Share
## 1 0.02539468 0.05367536
## Minnesota..Uninsured.Share Mississippi..Uninsured.Share
## 1 0.04118051 0.1183862
## Missouri..Uninsured.Share Montana..Uninsured.Share Nebraska..Uninsured.Share
## 1 0.08899202 0.08118266 0.08551256
## Nevada..Uninsured.Share New.Hampshire..Uninsured.Share
## 1 0.1137111 0.05900414
## New.Jersey..Uninsured.Share New.Mexico..Uninsured.Share
## 1 0.07976959 0.09173456
## New.York..Uninsured.Share North.Carolina..Uninsured.Share
## 1 0.06063357 0.1042796
## Ohio..Uninsured.Share Oklahoma..Uninsured.Share Oregon..Uninsured.Share
## 1 0.05627068 0.1377274 0.06234435
## Pennsylvania..Uninsured.Share South.Carolina..Uninsured.Share
## 1 0.05632031 0.09995622
## South.Dakota..Uninsured.Share Tennessee..Uninsured.Share
## 1 0.08744142 0.09047163
## Texas..Uninsured.Share Utah..Uninsured.Share Virginia..Uninsured.Share
## 1 0.1659744 0.08754625 0.08718548
## Washington..Uninsured.Share West.Virginia..Uninsured.Share
## 1 0.05958873 0.05319177
```

```
## Wisconsin..Uninsured.Share
## 1 0.05260606
```

```
#####

#something went wrong here with 2017 and 2018, so the function that worked is much more below as the co

# Copy original data 2017
df_pop_2017 <- dataset_uninsured_population_2017_cleaned
col_names <- colnames(df_pop_2017)
new_col_order <- col_names # to control column ordering

# Loop through all columns ending in "..Uninsured.Population..Estimate"
uninsured_cols <- grep("\\.\\.Uninsured\\.\\.Population\\.\\.Estimate$", col_names, value = TRUE)

for (uninsured_col in uninsured_cols) {
  # Extract the state name - everything before the first ".."
  state <- strsplit(uninsured_col, "\\..\\.")[1][1]

  # Define the total population column and the share column
  total_col <- paste0(state, "..Total.Civilian.Noninstitutionalized.Population..Estimate")
  share_col <- paste0(state, "..Uninsured.Share")

  cat("Processing:", state, "\n")

  if (total_col %in% col_names) {
    # Initialize new share column with NA
    df_pop_2017[[share_col]] <- NA

    # Calculate only for first row
    if (!is.na(df_pop_2017[[total_col]][1]) && df_pop_2017[[total_col]][1] != 0) {
      df_pop_2017[[share_col]][1] <- df_pop_2017[[uninsured_col]][1] / df_pop_2017[[total_col]][1]
    } else {
      df_pop_2017[[share_col]][1] <- NA
    }

    # Insert new column right after the uninsured column
    pos <- match(uninsured_col, new_col_order)
    new_col_order <- append(new_col_order, share_col, after = pos)

  } else {
    cat(" Total population column not found for:", state, "\n")
  }
}

# Reorder the columns
df_pop_2017 <- df_pop_2017[, new_col_order]

# Preview: Show share columns for the first row
share_cols <- grep("\\.\\.Uninsured.Share$", colnames(df_pop_2017), value = TRUE)
cat("\n Created share columns:\n")
```

```
##
## Created share columns:
```

```

print(share_cols)

## character(0)

print(df_pop_2017[1, share_cols])

## data frame with 0 columns and 1 row

# Copy original data 2018
df_pop_2018<- dataset_uninsured_population_2018_cleaned
col_names <- colnames(df_pop_2018)
new_col_order <- col_names # to control column ordering

# Loop through all columns ending in "..Uninsured.Population..Estimate"
uninsured_cols <- grep("\\\\.\\.Uninsured\\.\\.Population\\.\\.\\.Estimate$", col_names, value = TRUE)

for (uninsured_col in uninsured_cols) {
  # Extract the state name - everything before the first ".."
  state <- strsplit(uninsured_col, "\\\\.\\.")[[1]][1]

  # Define the total population column and the share column
  total_col <- paste0(state, "..Total.Civilian.Noninstitutionalized.Population..Estimate")
  share_col <- paste0(state, "..Uninsured.Share")

  cat("Processing:", state, "\n")

  if (total_col %in% col_names) {
    # Initialize new share column with NA
    df_pop_2018[[share_col]] <- NA

    # Calculate only for first row
    if (!is.na(df_pop_2018[[total_col]][1]) && df_pop_2018[[total_col]][1] != 0) {
      df_pop_2018[[share_col]][1] <- df_pop_2018[[uninsured_col]][1] / df_pop_2018[[total_col]][1]
    } else {
      df_pop_2018[[share_col]][1] <- NA
    }

    # Insert new column right after the uninsured column
    pos <- match(uninsured_col, new_col_order)
    new_col_order <- append(new_col_order, share_col, after = pos)

  } else {
    cat(" Total population column not found for:", state, "\n")
  }
}

# Reorder the columns
df_pop_2018 <- df_pop_2018[, new_col_order]

# Preview: Show share columns for the first row
share_cols <- grep("\\\\.\\.Uninsured.Share$", colnames(df_pop_2018), value = TRUE)
cat("\n Created share columns:\n")

```

```
##  
## Created share columns:
```

```
print(share_cols)
```

```
## character(0)
```

```
print(df_pop_2018[1, share_cols])
```

```
## data frame with 0 columns and 1 row
```

```
View(df_pop_2017)  
View(df_pop_2016)  
View(df_pop_2018)
```

```
#I have seen that it did not work for the years of 2017 and 2018 so I have to see what the names of the  
grep("Alabama", colnames(dataset_uninsured_population_2017_cleaned), value = TRUE)
```

```
## [1] "Alabama..Total..Estimate"          "Alabama..Total.Uninsured..Estimate"
```

```
grep("Alabama", colnames(dataset_uninsured_population_2018_cleaned), value = TRUE)
```

```
## [1] "Alabama..Total..Estimate"          "Alabama..Total.Uninsured..Estimate"
```

```
#Here I am starting again from 2017
```

```
# Make a working copy of your data 2017
```

```
df_pop_2017 <- dataset_uninsured_population_2017_cleaned
```

```
col_names <- colnames(df_pop_2017)
```

```
new_col_order <- col_names
```

```
# Loop through all columns ending with "..Total.Uninsured..Estimate"
```

```
uninsured_cols <- grep("\\.\\.Total\\.\\.Uninsured\\.\\.\\.Estimate$", col_names, value = TRUE)
```

```
for (uninsured_col in uninsured_cols) {
```

```
  # Extract state (everything before first "..")
```

```
  state <- strsplit(uninsured_col, "\\..\\.\\.")[[1]][1]
```

```
  # Construct matching total population column and new share column
```

```
  total_col <- paste0(state, "..Total..Estimate")
```

```
  share_col <- paste0(state, "..Uninsured.Share")
```

```
  cat("Processing:", state, "\n")
```

```
  # Check if the total population column exists
```

```
  if (total_col %in% col_names) {
```

```
    # Create new share column, default NA
```

```
    df_pop_2017[[share_col]] <- NA
```

```
    # Calculate share only for first row
```



```

total_val <- df_pop_2017[[total_col]][1]
uninsured_val <- df_pop_2017[[uninsured_col]][1]

if (!is.na(total_val) && total_val != 0) {
  df_pop_2017[[share_col]][1] <- uninsured_val / total_val
}

# Insert share column after the uninsured column
pos <- match(uninsured_col, new_col_order)
new_col_order <- append(new_col_order, share_col, after = pos)
} else {
  cat(" Total column not found for", state, "\n")
}
}

```

```

## Processing: Alabama
## Processing: Alaska
## Processing: Arizona
## Processing: Arkansas
## Processing: California
## Processing: Colorado
## Processing: Connecticut
## Processing: Florida
## Processing: Georgia
## Processing: Idaho
## Processing: Illinois
## Processing: Indiana
## Processing: Iowa
## Processing: Kansas
## Processing: Kentucky
## Processing: Louisiana
## Processing: Maine
## Processing: Maryland
## Processing: Massachusetts
## Processing: Michigan
## Processing: Minnesota
## Processing: Mississippi
## Processing: Missouri
## Processing: Montana
## Processing: Nebraska
## Processing: Nevada
## Processing: New.Hampshire
## Processing: New.Jersey
## Processing: New.Mexico
## Processing: New.York
## Processing: North.Carolina
## Processing: Ohio
## Processing: Oklahoma
## Processing: Oregon
## Processing: Pennsylvania
## Processing: South.Carolina
## Processing: South.Dakota
## Processing: Tennessee

```

```
## Processing: Texas
## Processing: Utah
## Processing: Virginia
## Processing: Washington
## Processing: West.Virginia
## Processing: Wisconsin
```

```
# Reorder columns
df_pop_2017 <- df_pop_2017[, new_col_order]

# Show all created share columns
share_cols <- grep("\\.\\..Uninsured\\.Share$", names(df_pop_2017), value = TRUE)
cat("\n Created columns:\n")
```

```
##
## Created columns:
```

```
print(share_cols)
```

```
## [1] "Alabama..Uninsured.Share" "Alaska..Uninsured.Share"
## [3] "Arizona..Uninsured.Share" "Arkansas..Uninsured.Share"
## [5] "California..Uninsured.Share" "Colorado..Uninsured.Share"
## [7] "Connecticut..Uninsured.Share" "Florida..Uninsured.Share"
## [9] "Georgia..Uninsured.Share" "Idaho..Uninsured.Share"
## [11] "Illinois..Uninsured.Share" "Indiana..Uninsured.Share"
## [13] "Iowa..Uninsured.Share" "Kansas..Uninsured.Share"
## [15] "Kentucky..Uninsured.Share" "Louisiana..Uninsured.Share"
## [17] "Maine..Uninsured.Share" "Maryland..Uninsured.Share"
## [19] "Massachusetts..Uninsured.Share" "Michigan..Uninsured.Share"
## [21] "Minnesota..Uninsured.Share" "Mississippi..Uninsured.Share"
## [23] "Missouri..Uninsured.Share" "Montana..Uninsured.Share"
## [25] "Nebraska..Uninsured.Share" "Nevada..Uninsured.Share"
## [27] "New.Hampshire..Uninsured.Share" "New.Jersey..Uninsured.Share"
## [29] "New.Mexico..Uninsured.Share" "New.York..Uninsured.Share"
## [31] "North.Carolina..Uninsured.Share" "Ohio..Uninsured.Share"
## [33] "Oklahoma..Uninsured.Share" "Oregon..Uninsured.Share"
## [35] "Pennsylvania..Uninsured.Share" "South.Carolina..Uninsured.Share"
## [37] "South.Dakota..Uninsured.Share" "Tennessee..Uninsured.Share"
## [39] "Texas..Uninsured.Share" "Utah..Uninsured.Share"
## [41] "Virginia..Uninsured.Share" "Washington..Uninsured.Share"
## [43] "West.Virginia..Uninsured.Share" "Wisconsin..Uninsured.Share"
```

```
print(df_pop_2017[1, share_cols])
```

```
## Alabama..Uninsured.Share Alaska..Uninsured.Share Arizona..Uninsured.Share
## 1 0.09365284 0.1365526 0.1005758
## Arkansas..Uninsured.Share California..Uninsured.Share
## 1 0.07859547 0.0716214
## Colorado..Uninsured.Share Connecticut..Uninsured.Share
## 1 0.07509462 0.05479847
## Florida..Uninsured.Share Georgia..Uninsured.Share Idaho..Uninsured.Share
## 1 0.1294156 0.1342736 0.1014663
```

```
## Illinois..Uninsured.Share Indiana..Uninsured.Share Iowa..Uninsured.Share
## 1 0.06808625 0.08166315 0.04706917
## Kansas..Uninsured.Share Kentucky..Uninsured.Share Louisiana..Uninsured.Share
## 1 0.08733883 0.05381611 0.08354311
## Maine..Uninsured.Share Maryland..Uninsured.Share
## 1 0.08101016 0.0613637
## Massachusetts..Uninsured.Share Michigan..Uninsured.Share
## 1 0.02797105 0.05171328
## Minnesota..Uninsured.Share Mississippi..Uninsured.Share
## 1 0.04393917 0.1204304
## Missouri..Uninsured.Share Montana..Uninsured.Share Nebraska..Uninsured.Share
## 1 0.09137354 0.08464118 0.08289077
## Nevada..Uninsured.Share New.Hampshire..Uninsured.Share
## 1 0.1122881 0.05815703
## New.Jersey..Uninsured.Share New.Mexico..Uninsured.Share
## 1 0.07728989 0.09102658
## New.York..Uninsured.Share North.Carolina..Uninsured.Share
## 1 0.0567594 0.1068421
## Ohio..Uninsured.Share Oklahoma..Uninsured.Share Oregon..Uninsured.Share
## 1 0.0597309 0.141636 0.06844328
## Pennsylvania..Uninsured.Share South.Carolina..Uninsured.Share
## 1 0.05492597 0.1098832
## South.Dakota..Uninsured.Share Tennessee..Uninsured.Share
## 1 0.09079084 0.0951436
## Texas..Uninsured.Share Utah..Uninsured.Share Virginia..Uninsured.Share
## 1 0.1730424 0.09158774 0.08831244
## Washington..Uninsured.Share West.Virginia..Uninsured.Share
## 1 0.06111198 0.06092967
## Wisconsin..Uninsured.Share
## 1 0.05405416
```

#now it worked and I can work further on 2018-2023 except for 2020 as we do not have the data of that

```
# Make a working copy of your data 2018
df_pop_2018 <- dataset_uninsured_population_2018_cleaned
col_names <- colnames(df_pop_2018)
new_col_order <- col_names

# Loop through all columns ending with "..Total.Uninsured..Estimate"
uninsured_cols <- grep("\\.\\.Total\\.\\.Uninsured\\.\\.Estimate$", col_names, value = TRUE)

for (uninsured_col in uninsured_cols) {
  # Extract state (everything before first "..")
  state <- strsplit(uninsured_col, "\\..\\.\\.")[1][1]

  # Construct matching total population column and new share column
  total_col <- paste0(state, "..Total..Estimate")
  share_col <- paste0(state, "..Uninsured.Share")

  cat("Processing:", state, "\n")

  # Check if the total population column exists
  if (total_col %in% col_names) {
    # Create new share column, default NA
```

```

df_pop_2018[[share_col]] <- NA

# Calculate share only for first row
total_val <- df_pop_2018[[total_col]][1]
uninsured_val <- df_pop_2018[[uninsured_col]][1]

if (!is.na(total_val) && total_val != 0) {
  df_pop_2018[[share_col]][1] <- uninsured_val / total_val
}

# Insert share column after the uninsured column
pos <- match(uninsured_col, new_col_order)
new_col_order <- append(new_col_order, share_col, after = pos)
} else {
  cat(" Total column not found for", state, "\n")
}
}

```

```

## Processing: Alabama
## Processing: Alaska
## Processing: Arizona
## Processing: Arkansas
## Processing: California
## Processing: Colorado
## Processing: Connecticut
## Processing: Florida
## Processing: Georgia
## Processing: Idaho
## Processing: Illinois
## Processing: Indiana
## Processing: Iowa
## Processing: Kansas
## Processing: Kentucky
## Processing: Louisiana
## Processing: Maine
## Processing: Maryland
## Processing: Massachusetts
## Processing: Michigan
## Processing: Minnesota
## Processing: Mississippi
## Processing: Missouri
## Processing: Montana
## Processing: Nebraska
## Processing: Nevada
## Processing: New Hampshire
## Processing: New Jersey
## Processing: New Mexico
## Processing: New York
## Processing: North Carolina
## Processing: Ohio
## Processing: Oklahoma
## Processing: Oregon
## Processing: Pennsylvania

```

```
## Processing: South.Carolina
## Processing: South.Dakota
## Processing: Tennessee
## Processing: Texas
## Processing: Utah
## Processing: Virginia
## Processing: Washington
## Processing: West.Virginia
## Processing: Wisconsin
```

```
# Reorder columns
```

```
df_pop_2018 <- df_pop_2018[, new_col_order]
```

```
# Show all created share columns
```

```
share_cols <- grep("\\\\.\\.Uninsured\\.\\.Share$", names(df_pop_2018), value = TRUE)
cat("\n Created columns:\n")
```

```
##
## Created columns:
```

```
print(share_cols)
```

```
## [1] "Alabama..Uninsured.Share"      "Alaska..Uninsured.Share"
## [3] "Arizona..Uninsured.Share"      "Arkansas..Uninsured.Share"
## [5] "California..Uninsured.Share"    "Colorado..Uninsured.Share"
## [7] "Connecticut..Uninsured.Share"   "Florida..Uninsured.Share"
## [9] "Georgia..Uninsured.Share"      "Idaho..Uninsured.Share"
## [11] "Illinois..Uninsured.Share"     "Indiana..Uninsured.Share"
## [13] "Iowa..Uninsured.Share"         "Kansas..Uninsured.Share"
## [15] "Kentucky..Uninsured.Share"     "Louisiana..Uninsured.Share"
## [17] "Maine..Uninsured.Share"        "Maryland..Uninsured.Share"
## [19] "Massachusetts..Uninsured.Share" "Michigan..Uninsured.Share"
## [21] "Minnesota..Uninsured.Share"    "Mississippi..Uninsured.Share"
## [23] "Missouri..Uninsured.Share"     "Montana..Uninsured.Share"
## [25] "Nebraska..Uninsured.Share"     "Nevada..Uninsured.Share"
## [27] "New.Hampshire..Uninsured.Share" "New.Jersey..Uninsured.Share"
## [29] "New.Mexico..Uninsured.Share"   "New.York..Uninsured.Share"
## [31] "North.Carolina..Uninsured.Share" "Ohio..Uninsured.Share"
## [33] "Oklahoma..Uninsured.Share"     "Oregon..Uninsured.Share"
## [35] "Pennsylvania..Uninsured.Share" "South.Carolina..Uninsured.Share"
## [37] "South.Dakota..Uninsured.Share" "Tennessee..Uninsured.Share"
## [39] "Texas..Uninsured.Share"        "Utah..Uninsured.Share"
## [41] "Virginia..Uninsured.Share"     "Washington..Uninsured.Share"
## [43] "West.Virginia..Uninsured.Share" "Wisconsin..Uninsured.Share"
```

```
print(df_pop_2018[1, share_cols])
```

```
## Alabama..Uninsured.Share Alaska..Uninsured.Share Arizona..Uninsured.Share
## 1 0.1000519 0.1264009 0.1061569
## Arkansas..Uninsured.Share California..Uninsured.Share
## 1 0.08227825 0.07233594
## Colorado..Uninsured.Share Connecticut..Uninsured.Share
```

```
## 1          0.07523842          0.05304523
##  Florida..Uninsured.Share Georgia..Uninsured.Share Idaho..Uninsured.Share
## 1          0.1299526          0.1365446          0.1111173
##  Illinois..Uninsured.Share Indiana..Uninsured.Share Iowa..Uninsured.Share
## 1          0.06961274          0.08268831          0.04723342
##  Kansas..Uninsured.Share Kentucky..Uninsured.Share Louisiana..Uninsured.Share
## 1          0.08769696          0.05643334          0.07975324
##  Maine..Uninsured.Share Maryland..Uninsured.Share
## 1          0.08024475          0.06002089
##  Massachusetts..Uninsured.Share Michigan..Uninsured.Share
## 1          0.0276552          0.05405769
##  Minnesota..Uninsured.Share Mississippi..Uninsured.Share
## 1          0.04400399          0.1212982
##  Missouri..Uninsured.Share Montana..Uninsured.Share Nebraska..Uninsured.Share
## 1          0.09415649          0.08216126          0.08294648
##  Nevada..Uninsured.Share New.Hampshire..Uninsured.Share
## 1          0.1121165          0.05749024
##  New.Jersey..Uninsured.Share New.Mexico..Uninsured.Share
## 1          0.0743981          0.0952338
##  New.York..Uninsured.Share North.Carolina..Uninsured.Share
## 1          0.05394253          0.1072345
##  Ohio..Uninsured.Share Oklahoma..Uninsured.Share Oregon..Uninsured.Share
## 1          0.06459064          0.1419926          0.07060989
##  Pennsylvania..Uninsured.Share South.Carolina..Uninsured.Share
## 1          0.05548705          0.1046962
##  South.Dakota..Uninsured.Share Tennessee..Uninsured.Share
## 1          0.0979727          0.1012128
##  Texas..Uninsured.Share Utah..Uninsured.Share Virginia..Uninsured.Share
## 1          0.1771363          0.09419778          0.08811778
##  Washington..Uninsured.Share West.Virginia..Uninsured.Share
## 1          0.06425818          0.06421398
##  Wisconsin..Uninsured.Share
## 1          0.05455078
```

```
#####
```

```
# Make a working copy of your data 2019
df_pop_2019 <- dataset_uninsured_population_2019_cleaned
col_names <- colnames(df_pop_2019)
new_col_order <- col_names

# Loop through all columns ending with "..Total.Uninsured..Estimate"
uninsured_cols <- grep("\\\\.\\.Total\\.\\.Uninsured\\.\\.\\.Estimate$", col_names, value = TRUE)

for (uninsured_col in uninsured_cols) {
  # Extract state (everything before first "..")
  state <- strsplit(uninsured_col, "\\\\.\\.\\.")[1][1]

  # Construct matching total population column and new share column
  total_col <- paste0(state, "..Total..Estimate")
  share_col <- paste0(state, "..Uninsured.Share")

  cat("Processing:", state, "\n")
}
```

```

# Check if the total population column exists
if (total_col %in% col_names) {
  # Create new share column, default NA
  df_pop_2019[[share_col]] <- NA

  # Calculate share only for first row
  total_val <- df_pop_2019[[total_col]][1]
  uninsured_val <- df_pop_2019[[uninsured_col]][1]

  if (!is.na(total_val) && total_val != 0) {
    df_pop_2019[[share_col]][1] <- uninsured_val / total_val
  }

  # Insert share column after the uninsured column
  pos <- match(uninsured_col, new_col_order)
  new_col_order <- append(new_col_order, share_col, after = pos)
} else {
  cat(" Total column not found for", state, "\n")
}
}

```

```

## Processing: Alabama
## Processing: Alaska
## Processing: Arizona
## Processing: Arkansas
## Processing: California
## Processing: Colorado
## Processing: Connecticut
## Processing: Florida
## Processing: Georgia
## Processing: Idaho
## Processing: Illinois
## Processing: Indiana
## Processing: Iowa
## Processing: Kansas
## Processing: Kentucky
## Processing: Louisiana
## Processing: Maine
## Processing: Maryland
## Processing: Massachusetts
## Processing: Michigan
## Processing: Minnesota
## Processing: Mississippi
## Processing: Missouri
## Processing: Montana
## Processing: Nebraska
## Processing: Nevada
## Processing: New.Hampshire
## Processing: New.Jersey
## Processing: New.Mexico
## Processing: New.York
## Processing: North.Carolina
## Processing: Ohio

```

```
## Processing: Oklahoma
## Processing: Oregon
## Processing: Pennsylvania
## Processing: South.Carolina
## Processing: South.Dakota
## Processing: Tennessee
## Processing: Texas
## Processing: Utah
## Processing: Virginia
## Processing: Washington
## Processing: West.Virginia
## Processing: Wisconsin
```

```
# Reorder columns
```

```
df_pop_2019 <- df_pop_2019[, new_col_order]
```

```
# Show all created share columns
```

```
share_cols <- grep("\\\\.\\.Uninsured\\.\\.Share$", names(df_pop_2019), value = TRUE)
cat("\n Created columns:\n")
```

```
##
```

```
## Created columns:
```

```
print(share_cols)
```

```
## [1] "Alabama..Uninsured.Share"      "Alaska..Uninsured.Share"
## [3] "Arizona..Uninsured.Share"      "Arkansas..Uninsured.Share"
## [5] "California..Uninsured.Share"   "Colorado..Uninsured.Share"
## [7] "Connecticut..Uninsured.Share"  "Florida..Uninsured.Share"
## [9] "Georgia..Uninsured.Share"      "Idaho..Uninsured.Share"
## [11] "Illinois..Uninsured.Share"     "Indiana..Uninsured.Share"
## [13] "Iowa..Uninsured.Share"         "Kansas..Uninsured.Share"
## [15] "Kentucky..Uninsured.Share"     "Louisiana..Uninsured.Share"
## [17] "Maine..Uninsured.Share"        "Maryland..Uninsured.Share"
## [19] "Massachusetts..Uninsured.Share" "Michigan..Uninsured.Share"
## [21] "Minnesota..Uninsured.Share"    "Mississippi..Uninsured.Share"
## [23] "Missouri..Uninsured.Share"     "Montana..Uninsured.Share"
## [25] "Nebraska..Uninsured.Share"     "Nevada..Uninsured.Share"
## [27] "New.Hampshire..Uninsured.Share" "New.Jersey..Uninsured.Share"
## [29] "New.Mexico..Uninsured.Share"   "New.York..Uninsured.Share"
## [31] "North.Carolina..Uninsured.Share" "Ohio..Uninsured.Share"
## [33] "Oklahoma..Uninsured.Share"     "Oregon..Uninsured.Share"
## [35] "Pennsylvania..Uninsured.Share" "South.Carolina..Uninsured.Share"
## [37] "South.Dakota..Uninsured.Share" "Tennessee..Uninsured.Share"
## [39] "Texas..Uninsured.Share"        "Utah..Uninsured.Share"
## [41] "Virginia..Uninsured.Share"     "Washington..Uninsured.Share"
## [43] "West.Virginia..Uninsured.Share" "Wisconsin..Uninsured.Share"
```

```
print(df_pop_2019[1, share_cols])
```

```
## Alabama..Uninsured.Share Alaska..Uninsured.Share Arizona..Uninsured.Share
## 1 0.09723165 0.1221173 0.1128459
```



```
## Arkansas..Uninsured.Share California..Uninsured.Share
## 1 0.0914637 0.07697906
## Colorado..Uninsured.Share Connecticut..Uninsured.Share
## 1 0.08005986 0.05895016
## Florida..Uninsured.Share Georgia..Uninsured.Share Idaho..Uninsured.Share
## 1 0.1316055 0.1341428 0.1083312
## Illinois..Uninsured.Share Indiana..Uninsured.Share Iowa..Uninsured.Share
## 1 0.07393202 0.08722513 0.05014984
## Kansas..Uninsured.Share Kentucky..Uninsured.Share Louisiana..Uninsured.Share
## 1 0.09200826 0.06444007 0.08904859
## Maine..Uninsured.Share Maryland..Uninsured.Share
## 1 0.08024643 0.0600135
## Massachusetts..Uninsured.Share Michigan..Uninsured.Share
## 1 0.02991936 0.0578107
## Minnesota..Uninsured.Share Mississippi..Uninsured.Share
## 1 0.04884795 0.1297496
## Missouri..Uninsured.Share Montana..Uninsured.Share Nebraska..Uninsured.Share
## 1 0.1002716 0.08270425 0.08293356
## Nevada..Uninsured.Share New.Hampshire..Uninsured.Share
## 1 0.1142238 0.06254871
## New.Jersey..Uninsured.Share New.Mexico..Uninsured.Share
## 1 0.07889048 0.09964263
## New.York..Uninsured.Share North.Carolina..Uninsured.Share
## 1 0.0523884 0.1125642
## Ohio..Uninsured.Share Oklahoma..Uninsured.Share Oregon..Uninsured.Share
## 1 0.06580679 0.1427902 0.07171733
## Pennsylvania..Uninsured.Share South.Carolina..Uninsured.Share
## 1 0.05768603 0.1084807
## South.Dakota..Uninsured.Share Tennessee..Uninsured.Share
## 1 0.1018523 0.1014459
## Texas..Uninsured.Share Utah..Uninsured.Share Virginia..Uninsured.Share
## 1 0.1835548 0.09654436 0.0792765
## Washington..Uninsured.Share West.Virginia..Uninsured.Share
## 1 0.06616207 0.06676534
## Wisconsin..Uninsured.Share
## 1 0.05714326
```

```
#####
```

```
# Make a working copy of your data 2021
df_pop_2021 <- dataset_uninsured_population_2021_cleaned
col_names <- colnames(df_pop_2021)
new_col_order <- col_names

# Loop through all columns ending with "..Total.Uninsured..Estimate"
uninsured_cols <- grep("\\\\.\\.Total\\.\\.Uninsured\\.\\.Estimate$", col_names, value = TRUE)

for (uninsured_col in uninsured_cols) {
  # Extract state (everything before first "..")
  state <- strsplit(uninsured_col, "\\\\.\\.")[[1]][1]

  # Construct matching total population column and new share column
  total_col <- paste0(state, "..Total..Estimate")
}
```

```

share_col <- paste0(state, "..Uninsured.Share")

cat("Processing:", state, "\n")

# Check if the total population column exists
if (total_col %in% col_names) {
  # Create new share column, default NA
  df_pop_2021[[share_col]] <- NA

  # Calculate share only for first row
  total_val <- df_pop_2021[[total_col]][1]
  uninsured_val <- df_pop_2021[[uninsured_col]][1]

  if (!is.na(total_val) && total_val != 0) {
    df_pop_2021[[share_col]][1] <- uninsured_val / total_val
  }

  # Insert share column after the uninsured column
  pos <- match(uninsured_col, new_col_order)
  new_col_order <- append(new_col_order, share_col, after = pos)
} else {
  cat(" Total column not found for", state, "\n")
}
}

```

```

## Processing: Alabama
## Processing: Alaska
## Processing: Arizona
## Processing: Arkansas
## Processing: California
## Processing: Colorado
## Processing: Connecticut
## Processing: Florida
## Processing: Georgia
## Processing: Idaho
## Processing: Illinois
## Processing: Indiana
## Processing: Iowa
## Processing: Kansas
## Processing: Kentucky
## Processing: Louisiana
## Processing: Maine
## Processing: Maryland
## Processing: Massachusetts
## Processing: Michigan
## Processing: Minnesota
## Processing: Mississippi
## Processing: Missouri
## Processing: Montana
## Processing: Nebraska
## Processing: Nevada
## Processing: New.Hampshire
## Processing: New.Jersey

```

```
## Processing: New.Mexico
## Processing: New.York
## Processing: North.Carolina
## Processing: Ohio
## Processing: Oklahoma
## Processing: Oregon
## Processing: Pennsylvania
## Processing: South.Carolina
## Processing: South.Dakota
## Processing: Tennessee
## Processing: Texas
## Processing: Utah
## Processing: Virginia
## Processing: Washington
## Processing: West.Virginia
## Processing: Wisconsin
```

```
# Reorder columns
df_pop_2021 <- df_pop_2021[, new_col_order]

# Show all created share columns
share_cols <- grep("\\\\.\\.Uninsured\\.\\.Share$", names(df_pop_2021), value = TRUE)
cat("\n Created columns:\n")
```

```
##
## Created columns:
```

```
print(share_cols)
```

```
## [1] "Alabama..Uninsured.Share" "Alaska..Uninsured.Share"
## [3] "Arizona..Uninsured.Share" "Arkansas..Uninsured.Share"
## [5] "California..Uninsured.Share" "Colorado..Uninsured.Share"
## [7] "Connecticut..Uninsured.Share" "Florida..Uninsured.Share"
## [9] "Georgia..Uninsured.Share" "Idaho..Uninsured.Share"
## [11] "Illinois..Uninsured.Share" "Indiana..Uninsured.Share"
## [13] "Iowa..Uninsured.Share" "Kansas..Uninsured.Share"
## [15] "Kentucky..Uninsured.Share" "Louisiana..Uninsured.Share"
## [17] "Maine..Uninsured.Share" "Maryland..Uninsured.Share"
## [19] "Massachusetts..Uninsured.Share" "Michigan..Uninsured.Share"
## [21] "Minnesota..Uninsured.Share" "Mississippi..Uninsured.Share"
## [23] "Missouri..Uninsured.Share" "Montana..Uninsured.Share"
## [25] "Nebraska..Uninsured.Share" "Nevada..Uninsured.Share"
## [27] "New.Hampshire..Uninsured.Share" "New.Jersey..Uninsured.Share"
## [29] "New.Mexico..Uninsured.Share" "New.York..Uninsured.Share"
## [31] "North.Carolina..Uninsured.Share" "Ohio..Uninsured.Share"
## [33] "Oklahoma..Uninsured.Share" "Oregon..Uninsured.Share"
## [35] "Pennsylvania..Uninsured.Share" "South.Carolina..Uninsured.Share"
## [37] "South.Dakota..Uninsured.Share" "Tennessee..Uninsured.Share"
## [39] "Texas..Uninsured.Share" "Utah..Uninsured.Share"
## [41] "Virginia..Uninsured.Share" "Washington..Uninsured.Share"
## [43] "West.Virginia..Uninsured.Share" "Wisconsin..Uninsured.Share"
```

```
print(df_pop_2021[1, share_cols])
```

```
## Alabama..Uninsured.Share Alaska..Uninsured.Share Arizona..Uninsured.Share
## 1 0.09854925 0.1140533 0.1067932
## Arkansas..Uninsured.Share California..Uninsured.Share
## 1 0.09178368 0.0700475
## Colorado..Uninsured.Share Connecticut..Uninsured.Share
## 1 0.07967111 0.0515895
## Florida..Uninsured.Share Georgia..Uninsured.Share Idaho..Uninsured.Share
## 1 0.121051 0.1263088 0.08829542
## Illinois..Uninsured.Share Indiana..Uninsured.Share Iowa..Uninsured.Share
## 1 0.07003409 0.0751946 0.04800897
## Kansas..Uninsured.Share Kentucky..Uninsured.Share Louisiana..Uninsured.Share
## 1 0.09161134 0.05663162 0.07646627
## Maine..Uninsured.Share Maryland..Uninsured.Share
## 1 0.05720865 0.06094254
## Massachusetts..Uninsured.Share Michigan..Uninsured.Share
## 1 0.02496419 0.04978955
## Minnesota..Uninsured.Share Mississippi..Uninsured.Share
## 1 0.04455518 0.1188529
## Missouri..Uninsured.Share Montana..Uninsured.Share Nebraska..Uninsured.Share
## 1 0.0941563 0.08216107 0.07106947
## Nevada..Uninsured.Share New.Hampshire..Uninsured.Share
## 1 0.1164494 0.05135849
## New.Jersey..Uninsured.Share New.Mexico..Uninsured.Share
## 1 0.07166672 0.09970751
## New.York..Uninsured.Share North.Carolina..Uninsured.Share
## 1 0.05200696 0.104206
## Ohio..Uninsured.Share Oklahoma..Uninsured.Share Oregon..Uninsured.Share
## 1 0.06525657 0.1377177 0.06072583
## Pennsylvania..Uninsured.Share South.Carolina..Uninsured.Share
## 1 0.0549479 0.1003831
## South.Dakota..Uninsured.Share Tennessee..Uninsured.Share
## 1 0.09491317 0.09983068
## Texas..Uninsured.Share Utah..Uninsured.Share Virginia..Uninsured.Share
## 1 0.179725 0.09027935 0.06825086
## Washington..Uninsured.Share West.Virginia..Uninsured.Share
## 1 0.06406261 0.06083364
## Wisconsin..Uninsured.Share
## 1 0.05356075
```

```
#####
```

```
# Make a working copy of your data 2022
```

```
df_pop_2022 <- dataset_uninsured_population_2022_cleaned
```

```
col_names <- colnames(df_pop_2022)
```

```
new_col_order <- col_names
```

```
# Loop through all columns ending with "..Total.Uninsured..Estimate"
```

```
uninsured_cols <- grep("\\.\\.Total\\.\\.Uninsured\\.\\.\\.Estimate$", col_names, value = TRUE)
```

```
for (uninsured_col in uninsured_cols) {
```

```

# Extract state (everything before first "..")
state <- strsplit(uninsured_col, "\\..\\.")[[1]][1]

# Construct matching total population column and new share column
total_col <- paste0(state, "..Total..Estimate")
share_col <- paste0(state, "..Uninsured.Share")

cat("Processing:", state, "\n")

# Check if the total population column exists
if (total_col %in% col_names) {
  # Create new share column, default NA
  df_pop_2022[[share_col]] <- NA

  # Calculate share only for first row
  total_val <- df_pop_2022[[total_col]][1]
  uninsured_val <- df_pop_2022[[uninsured_col]][1]

  if (!is.na(total_val) && total_val != 0) {
    df_pop_2022[[share_col]][1] <- uninsured_val / total_val
  }

  # Insert share column after the uninsured column
  pos <- match(uninsured_col, new_col_order)
  new_col_order <- append(new_col_order, share_col, after = pos)
} else {
  cat(" Total column not found for", state, "\n")
}
}

```

```

## Processing: Alabama
## Processing: Alaska
## Processing: Arizona
## Processing: Arkansas
## Processing: California
## Processing: Colorado
## Processing: Connecticut
## Processing: Florida
## Processing: Georgia
## Processing: Idaho
## Processing: Illinois
## Processing: Indiana
## Processing: Iowa
## Processing: Kansas
## Processing: Kentucky
## Processing: Louisiana
## Processing: Maine
## Processing: Maryland
## Processing: Massachusetts
## Processing: Michigan
## Processing: Minnesota
## Processing: Mississippi
## Processing: Missouri

```

```
## Processing: Montana
## Processing: Nebraska
## Processing: Nevada
## Processing: New.Hampshire
## Processing: New.Jersey
## Processing: New.Mexico
## Processing: New.York
## Processing: North.Carolina
## Processing: Ohio
## Processing: Oklahoma
## Processing: Oregon
## Processing: Pennsylvania
## Processing: South.Carolina
## Processing: South.Dakota
## Processing: Tennessee
## Processing: Texas
## Processing: Utah
## Processing: Virginia
## Processing: Washington
## Processing: West.Virginia
## Processing: Wisconsin
```

```
# Reorder columns
```

```
df_pop_2022 <- df_pop_2022[, new_col_order]
```

```
# Show all created share columns
```

```
share_cols <- grep("\\\\.\\.Uninsured\\.\\.Share$", names(df_pop_2022), value = TRUE)
cat("\n Created columns:\n")
```

```
##
## Created columns:
```

```
print(share_cols)
```

```
## [1] "Alabama..Uninsured.Share"      "Alaska..Uninsured.Share"
## [3] "Arizona..Uninsured.Share"      "Arkansas..Uninsured.Share"
## [5] "California..Uninsured.Share"    "Colorado..Uninsured.Share"
## [7] "Connecticut..Uninsured.Share"  "Florida..Uninsured.Share"
## [9] "Georgia..Uninsured.Share"      "Idaho..Uninsured.Share"
## [11] "Illinois..Uninsured.Share"     "Indiana..Uninsured.Share"
## [13] "Iowa..Uninsured.Share"         "Kansas..Uninsured.Share"
## [15] "Kentucky..Uninsured.Share"     "Louisiana..Uninsured.Share"
## [17] "Maine..Uninsured.Share"        "Maryland..Uninsured.Share"
## [19] "Massachusetts..Uninsured.Share" "Michigan..Uninsured.Share"
## [21] "Minnesota..Uninsured.Share"    "Mississippi..Uninsured.Share"
## [23] "Missouri..Uninsured.Share"     "Montana..Uninsured.Share"
## [25] "Nebraska..Uninsured.Share"     "Nevada..Uninsured.Share"
## [27] "New.Hampshire..Uninsured.Share" "New.Jersey..Uninsured.Share"
## [29] "New.Mexico..Uninsured.Share"   "New.York..Uninsured.Share"
## [31] "North.Carolina..Uninsured.Share" "Ohio..Uninsured.Share"
## [33] "Oklahoma..Uninsured.Share"     "Oregon..Uninsured.Share"
## [35] "Pennsylvania..Uninsured.Share" "South.Carolina..Uninsured.Share"
## [37] "South.Dakota..Uninsured.Share" "Tennessee..Uninsured.Share"
```

```
## [39] "Texas..Uninsured.Share"      "Utah..Uninsured.Share"
## [41] "Virginia..Uninsured.Share"    "Washington..Uninsured.Share"
## [43] "West.Virginia..Uninsured.Share" "Wisconsin..Uninsured.Share"
```

```
print(df_pop_2022[1, share_cols])
```

```
## Alabama..Uninsured.Share Alaska..Uninsured.Share Arizona..Uninsured.Share
## 1 0.08765301 0.1097246 0.1032432
## Arkansas..Uninsured.Share California..Uninsured.Share
## 1 0.08429496 0.06464148
## Colorado..Uninsured.Share Connecticut..Uninsured.Share
## 1 0.07114232 0.05162027
## Florida..Uninsured.Share Georgia..Uninsured.Share Idaho..Uninsured.Share
## 1 0.1117466 0.1167391 0.08192574
## Illinois..Uninsured.Share Indiana..Uninsured.Share Iowa..Uninsured.Share
## 1 0.06551264 0.06967824 0.04464984
## Kansas..Uninsured.Share Kentucky..Uninsured.Share Louisiana..Uninsured.Share
## 1 0.08590088 0.05576913 0.06946939
## Maine..Uninsured.Share Maryland..Uninsured.Share
## 1 0.0656296 0.0606604
## Massachusetts..Uninsured.Share Michigan..Uninsured.Share
## 1 0.02423125 0.04546727
## Minnesota..Uninsured.Share Mississippi..Uninsured.Share
## 1 0.04497038 0.1084825
## Missouri..Uninsured.Share Montana..Uninsured.Share Nebraska..Uninsured.Share
## 1 0.08577111 0.08259932 0.06679221
## Nevada..Uninsured.Share New.Hampshire..Uninsured.Share
## 1 0.1110718 0.04895806
## New.Jersey..Uninsured.Share New.Mexico..Uninsured.Share
## 1 0.06840516 0.08217944
## New.York..Uninsured.Share North.Carolina..Uninsured.Share
## 1 0.04851316 0.09284511
## Ohio..Uninsured.Share Oklahoma..Uninsured.Share Oregon..Uninsured.Share
## 1 0.05890281 0.1172645 0.06011162
## Pennsylvania..Uninsured.Share South.Carolina..Uninsured.Share
## 1 0.05324045 0.09073011
## South.Dakota..Uninsured.Share Tennessee..Uninsured.Share
## 1 0.08057311 0.09308205
## Texas..Uninsured.Share Utah..Uninsured.Share Virginia..Uninsured.Share
## 1 0.1658447 0.08128566 0.06452529
## Washington..Uninsured.Share West.Virginia..Uninsured.Share
## 1 0.06098285 0.05919066
## Wisconsin..Uninsured.Share
## 1 0.0519327
```

```
#####
```

```
# Make a working copy of your data 2023
```

```
df_pop_2023 <- dataset_uninsured_population_2023_cleaned
```

```
col_names <- colnames(df_pop_2023)
```

```
new_col_order <- col_names
```

```
# Loop through all columns ending with "..Total.Uninsured..Estimate"
```

```

uninsured_cols <- grep("\\.\\.Total\\.Uninsured\\.\\.\\.Estimate$", col_names, value = TRUE)

for (uninsured_col in uninsured_cols) {
  # Extract state (everything before first "..")
  state <- strsplit(uninsured_col, "\\\\.\\.\\.")[[1]][1]

  # Construct matching total population column and new share column
  total_col <- paste0(state, "..Total..Estimate")
  share_col <- paste0(state, "..Uninsured.Share")

  cat("Processing:", state, "\n")

  # Check if the total population column exists
  if (total_col %in% col_names) {
    # Create new share column, default NA
    df_pop_2023[[share_col]] <- NA

    # Calculate share only for first row
    total_val <- df_pop_2023[[total_col]][1]
    uninsured_val <- df_pop_2023[[uninsured_col]][1]

    if (!is.na(total_val) && total_val != 0) {
      df_pop_2023[[share_col]][1] <- uninsured_val / total_val
    }

    # Insert share column after the uninsured column
    pos <- match(uninsured_col, new_col_order)
    new_col_order <- append(new_col_order, share_col, after = pos)
  } else {
    cat(" Total column not found for", state, "\n")
  }
}

```

```

## Processing: Alabama
## Processing: Alaska
## Processing: Arizona
## Processing: Arkansas
## Processing: California
## Processing: Colorado
## Processing: Connecticut
## Processing: Florida
## Processing: Georgia
## Processing: Idaho
## Processing: Illinois
## Processing: Indiana
## Processing: Iowa
## Processing: Kansas
## Processing: Kentucky
## Processing: Louisiana
## Processing: Maine
## Processing: Maryland
## Processing: Massachusetts
## Processing: Michigan

```



```
## Processing: Minnesota
## Processing: Mississippi
## Processing: Missouri
## Processing: Montana
## Processing: Nebraska
## Processing: Nevada
## Processing: New.Hampshire
## Processing: New.Jersey
## Processing: New.Mexico
## Processing: New.York
## Processing: North.Carolina
## Processing: Ohio
## Processing: Oklahoma
## Processing: Oregon
## Processing: Pennsylvania
## Processing: South.Carolina
## Processing: South.Dakota
## Processing: Tennessee
## Processing: Texas
## Processing: Utah
## Processing: Virginia
## Processing: Washington
## Processing: West.Virginia
## Processing: Wisconsin
```

```
# Reorder columns
```

```
df_pop_2023 <- df_pop_2023[, new_col_order]
```

```
# Show all created share columns
```

```
share_cols <- grep("\\\\.\\.Uninsured\\.\\.Share$", names(df_pop_2023), value = TRUE)
cat("\n Created columns:\n")
```

```
##
## Created columns:
```

```
print(share_cols)
```

```
## [1] "Alabama..Uninsured.Share"      "Alaska..Uninsured.Share"
## [3] "Arizona..Uninsured.Share"      "Arkansas..Uninsured.Share"
## [5] "California..Uninsured.Share"    "Colorado..Uninsured.Share"
## [7] "Connecticut..Uninsured.Share"   "Florida..Uninsured.Share"
## [9] "Georgia..Uninsured.Share"       "Idaho..Uninsured.Share"
## [11] "Illinois..Uninsured.Share"      "Indiana..Uninsured.Share"
## [13] "Iowa..Uninsured.Share"          "Kansas..Uninsured.Share"
## [15] "Kentucky..Uninsured.Share"      "Louisiana..Uninsured.Share"
## [17] "Maine..Uninsured.Share"         "Maryland..Uninsured.Share"
## [19] "Massachusetts..Uninsured.Share" "Michigan..Uninsured.Share"
## [21] "Minnesota..Uninsured.Share"     "Mississippi..Uninsured.Share"
## [23] "Missouri..Uninsured.Share"      "Montana..Uninsured.Share"
## [25] "Nebraska..Uninsured.Share"      "Nevada..Uninsured.Share"
## [27] "New.Hampshire..Uninsured.Share" "New.Jersey..Uninsured.Share"
## [29] "New.Mexico..Uninsured.Share"    "New.York..Uninsured.Share"
## [31] "North.Carolina..Uninsured.Share" "Ohio..Uninsured.Share"
```

```
## [33] "Oklahoma..Uninsured.Share"      "Oregon..Uninsured.Share"
## [35] "Pennsylvania..Uninsured.Share"  "South.Carolina..Uninsured.Share"
## [37] "South.Dakota..Uninsured.Share"  "Tennessee..Uninsured.Share"
## [39] "Texas..Uninsured.Share"         "Utah..Uninsured.Share"
## [41] "Virginia..Uninsured.Share"      "Washington..Uninsured.Share"
## [43] "West.Virginia..Uninsured.Share" "Wisconsin..Uninsured.Share"
```

```
print(df_pop_2023[1, share_cols])
```

```
## Alabama..Uninsured.Share Alaska..Uninsured.Share Arizona..Uninsured.Share
## 1          0.08490298          0.1038765          0.0993637
## Arkansas..Uninsured.Share California..Uninsured.Share
## 1          0.08920865          0.06386153
## Colorado..Uninsured.Share Connecticut..Uninsured.Share
## 1          0.06703853          0.05740886
## Florida..Uninsured.Share Georgia..Uninsured.Share Idaho..Uninsured.Share
## 1          0.1069023          0.1138529          0.08854436
## Illinois..Uninsured.Share Indiana..Uninsured.Share Iowa..Uninsured.Share
## 1          0.06160363          0.06853826          0.0502048
## Kansas..Uninsured.Share Kentucky..Uninsured.Share Louisiana..Uninsured.Share
## 1          0.0835362          0.05430606          0.06918709
## Maine..Uninsured.Share Maryland..Uninsured.Share
## 1          0.05912624          0.06264785
## Massachusetts..Uninsured.Share Michigan..Uninsured.Share
## 1          0.02582464          0.0451428
## Minnesota..Uninsured.Share Mississippi..Uninsured.Share
## 1          0.04187831          0.1034669
## Missouri..Uninsured.Share Montana..Uninsured.Share Nebraska..Uninsured.Share
## 1          0.07520097          0.08433544          0.06121848
## Nevada..Uninsured.Share New.Hampshire..Uninsured.Share
## 1          0.1080003          0.04734072
## New.Jersey..Uninsured.Share New.Mexico..Uninsured.Share
## 1          0.07190929          0.09082783
## New.York..Uninsured.Share North.Carolina..Uninsured.Share
## 1          0.04843565          0.0923095
## Ohio..Uninsured.Share Oklahoma..Uninsured.Share Oregon..Uninsured.Share
## 1          0.06134679          0.1135415          0.05492359
## Pennsylvania..Uninsured.Share South.Carolina..Uninsured.Share
## 1          0.05418139          0.09100654
## South.Dakota..Uninsured.Share Tennessee..Uninsured.Share
## 1          0.0826191          0.09305311
## Texas..Uninsured.Share Utah..Uninsured.Share Virginia..Uninsured.Share
## 1          0.1637895          0.07969644          0.06401962
## Washington..Uninsured.Share West.Virginia..Uninsured.Share
## 1          0.06252503          0.05916401
## Wisconsin..Uninsured.Share
## 1          0.04856452
```

```
#####
```

```
#Now we have the new variable in the separate datasets and now we are going to merge these yearly datas
```

```
# Define years
```

```

years_early <- 2010:2016
years_late <- c(2017:2019, 2021:2023)

# Function to process one year of data
process_df <- function(df, year, is_early) {
  label_col <- names(df)[1]
  state_cols <- names(df)[!names(df) %in% c(label_col, "Unnamed: 0")]
  states <- unique(str_extract(state_cols, "^([.]+)"))

  # Set correct suffixes depending on year
  if (is_early) {
    pop_suffix <- "..Total.Civilian.Noninstitutionalized.Population..Estimate"
    unins_suffix <- "..Uninsured.Population..Estimate"
    share_suffix <- "..Uninsured.Share"
  } else {
    pop_suffix <- "..Total..Estimate"
    unins_suffix <- "..Total.Uninsured..Estimate"
    share_suffix <- "..Uninsured.Share"
  }

  # Extract for each state
  rows <- lapply(states, function(state) {
    pop_col <- paste0(state, pop_suffix)
    unins_col <- paste0(state, unins_suffix)
    share_col <- paste0(state, share_suffix)

    # Get only total population row
    total_row <- df %>%
      filter(str_trim(.data[[label_col]]) == "Total population")

    if (nrow(total_row) == 0 || !(pop_col %in% names(df))) return(NULL)

    tibble(
      state = state,
      year = year,
      total_population = as.numeric(total_row[[pop_col]]),
      total_uninsured = as.numeric(total_row[[unins_col]]),
      uninsured_share = as.numeric(total_row[[share_col]])
    )
  })

  bind_rows(rows)
}

# Process all early years
dfs_early <- mget(paste0("df_pop_", years_early))
panel_early <- map2_dfr(dfs_early, years_early, ~process_df(.x, .y, is_early = TRUE))

# Process all late years
dfs_late <- mget(paste0("df_pop_", years_late))
#panel_late <- map2_dfr(dfs_late, years_late, ~process_df(.x, .y, is_early = FALSE))

early_states <- bind_rows(

```

```
dfs_early, .id = "year") %>%
pivot_longer(
  ~c(year, "Label (Grouping)"),
  names_to = "tmp",
  values_to = "value") %>%
separate(tmp,
  into = c("state", "series", "measure"),
  sep = "\\.\\".,
  remove = TRUE)
```

```
## Warning: Expected 3 pieces. Missing pieces filled with `NA` in 2835 rows [3, 6, 9, 12,
## 15, 18, 21, 24, 27, 30, 33, 36, 39, 42, 45, 48, 51, 54, 57, 60, ...].
```

```
late_states <- bind_rows(
  dfs_late, .id = "year") %>%
pivot_longer(
  ~c(year, "Label (Grouping)"),
  names_to = "tmp",
  values_to = "value") %>%
separate(tmp,
  into = c("state", "series", "measure"),
  sep = "\\.\\".,
  remove = TRUE)
```

```
## Warning: Expected 3 pieces. Missing pieces filled with `NA` in 2376 rows [3, 6, 9, 12,
## 15, 18, 21, 24, 27, 30, 33, 36, 39, 42, 45, 48, 51, 54, 57, 60, ...].
```

```
# View
```

```
all_states <- rbind(late_states, early_states)

head(all_states)
```

```
## # A tibble: 6 x 6
##   year      `Label (Grouping)`      state series measure  value
##   <chr>      <chr>                <chr> <chr>  <chr>    <dbl>
## 1 df_pop_2017 Total civilian noninstitutionalized ~ Alab~ Total  Estima~ 4.79e+6
## 2 df_pop_2017 Total civilian noninstitutionalized ~ Alab~ Total  Estima~ 4.49e+5
## 3 df_pop_2017 Total civilian noninstitutionalized ~ Alab~ Unins~ <NA>    9.37e-2
## 4 df_pop_2017 Total civilian noninstitutionalized ~ Alas~ Total  Estima~ 7.16e+5
## 5 df_pop_2017 Total civilian noninstitutionalized ~ Alas~ Total  Estima~ 9.78e+4
## 6 df_pop_2017 Total civilian noninstitutionalized ~ Alas~ Unins~ <NA>    1.37e-1
```

```
#####
```

```
#cleaning out df_pop from the column of years to only remain the year numbers
```

```
unique(all_states$year)
```

```
## [1] "df_pop_2017" "df_pop_2018" "df_pop_2019" "df_pop_2021" "df_pop_2022"
## [6] "df_pop_2023" "df_pop_2010" "df_pop_2011" "df_pop_2012" "df_pop_2013"
## [11] "df_pop_2014" "df_pop_2015" "df_pop_2016"
```

```

all_states$year <- str_remove(all_states$year, "df_pop_")

#Now I am coercing the string into numeric so plotting becomes easier

all_states$year <- as.numeric(all_states$year)
str(all_states)

## tibble [15,633 x 6] (S3: tbl_df/tbl/data.frame)
## $ year          : num [1:15633] 2017 2017 2017 2017 2017 ...
## $ Label (Grouping): chr [1:15633] "Total civilian noninstitutionalized population" "Total civilian ...
## $ state         : chr [1:15633] "Alabama" "Alabama" "Alabama" "Alaska" ...
## $ series        : chr [1:15633] "Total" "Total.Uninsured" "Uninsured.Share" "Total" ...
## $ measure       : chr [1:15633] "Estimate" "Estimate" NA "Estimate" ...
## $ value         : num [1:15633] 4.79e+06 4.49e+05 9.37e-02 7.16e+05 9.78e+04 ...

#####
#Here I am multiplying the uninsured share by 100, such that they are representing percentages
all_states_100 = all_states %>%
  mutate(value =
    ifelse(grepl("Uninsured.Share",
      series), value*100, value))

#This is the dataset for the Uninsured Population we can use for the visualizations

```

1.4 datacleaning Annual GDP Table and The Creation of The New Variable: Annual Real GDP Growth per year and state

```

#Here I am making a new variable which is the GDP growth rate of the states

#renaming the table
table_GDP = Table_GDP_absolute_values_Table_3_

#class coercion of double to numeric of all columns but the first column
table_GDP[, -1] = lapply(table_GDP[, -1], as.numeric)

#checking the class
sapply(table_GDP, class)

```

```

##      GeoName      2009      2010      2011      2012      2013
## "character" "numeric" "numeric" "numeric" "numeric" "numeric"
##      2014      2015      2016      2017      2018      2019
## "numeric" "numeric" "numeric" "numeric" "numeric" "numeric"
##      2021      2022      2023
## "numeric" "numeric" "numeric"

```

```

#yes, all the numbers are now numeric instead of the first column with the names of the states

# Pivot longer
gdp_long <- table_GDP %>%

```

```

pivot_longer(
  cols = -GeoName,
  names_to = "year",
  values_to = "gdp"
) %>%
mutate(year = as.integer(year))

View(table_GDP)
View(gdp_long)

# Calculate GDP growth rate
gdp_growth <- gdp_long %>%
  group_by(GeoName) %>%
  arrange(year) %>%
  mutate(growth_rate = (gdp - lag(gdp)) / lag(gdp) * 100)

View(gdp_growth)

#delete rows from 2009
gdp_growth = gdp_growth[-c(1:45), ]

```

2 Part 3 – Visualize and Analyze the Data

2.1 3.1 Create Initial Visualizations

Spatial Variation Visualization, U.S. Share of The Uninsured Population in 2023

- describe here that we mapped here the spatial variation of the uninsured population of the states of which we have the ACS one year estimates of and that we first created a subset of 2023 of the big dataset all_states_100 and that we plotted Alaska separately otherwise the whole map would be too small to visualize.

```

#Here I am making a sub dataframe for the spatial visualization of 2023 of all the states with availabl

spatial_df = all_states %>%
  filter(year == 2023, series == "Uninsured.Share") %>%
  select(state, value)

#deleting unnecessary rows
spatial_df = spatial_df[-c(45:396),]

#Here I am trying to plot the spatial subset

#getting the US map
us_map = map_data("state")

#clean up states names in both datasets
us_map$region = tolower(us_map$region)
spatial_df$state = tolower(spatial_df$state)

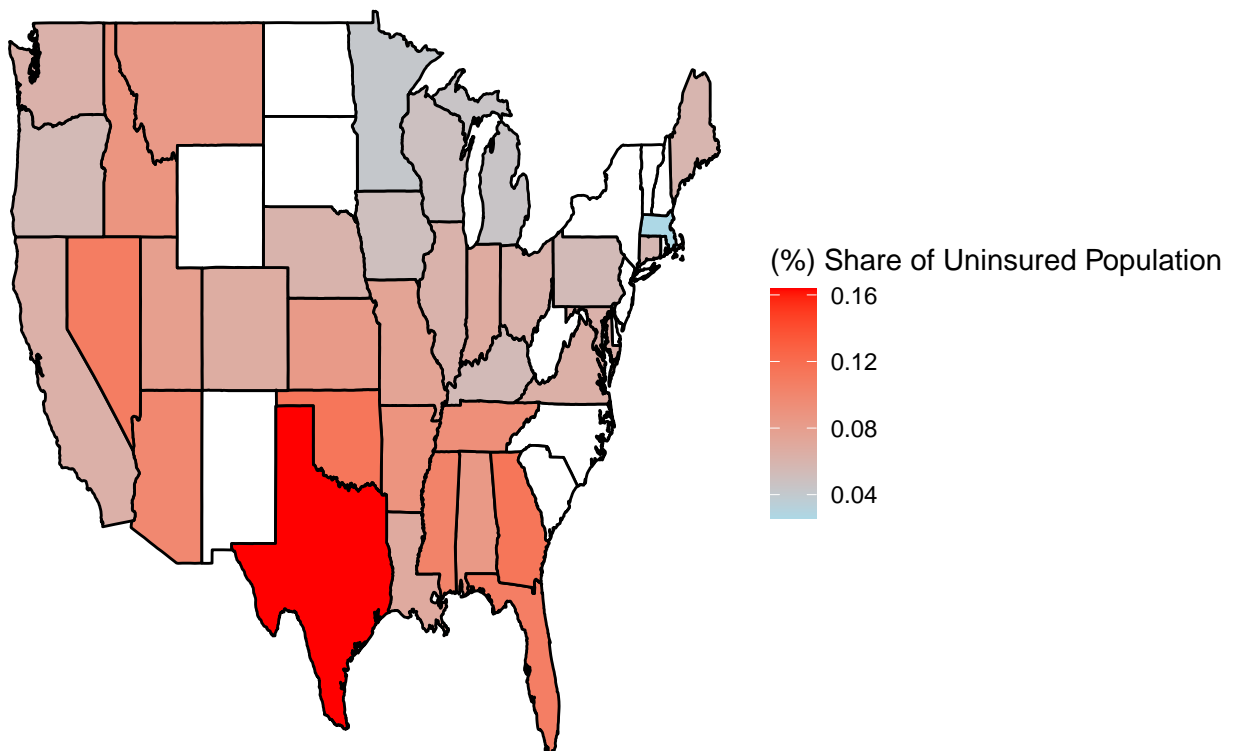
#join the map data with the subset data

```

```
us_map = left_join(us_map, spatial_df, by = c("region" = "state"))

#plot the map
ggplot(us_map, aes(x = long, y = lat, group = group, fill = value)) +
  geom_polygon(color = "black") +
  scale_fill_gradient(low = "lightblue", high = "red", na.value = "white") +
  ggtitle("Share of Uninsured Population in the US, 2023") +
  labs(fill = "(%) Share of Uninsured Population") +
  theme_void()
```

Share of Uninsured Population in the US, 2023



```
#I still need the state Alaska but the package maps does not include the state Alaska

# I am trying it here again including Alaska with another package

library(rnaturalearth)
library(ggplot2)
library(dplyr)

#get US shapefile
us_states = ne_states(country = "United States of America", returnclass = "sf")

#convert state names into lowercase letters such that they match
us_states$name = tolower(us_states$name)

alaska = us_states[us_states$name == "alaska",]
```

```

#merge data with shapefile
alaska_data = alaska %>%
  left_join(spatial_df, by = c("name" = "state"))

#plot the map with the data
ggplot(alaska_data) +
  geom_sf(aes(fill = value)) +
  theme_void() +
  scale_fill_gradient( low = "lightblue", high = "red", na.value = "white")+
  theme(legend.position = "none")

```



Temporal Variation Visualization, Texas's Share of Uninsured Population and Texas's GDP growth rate over time

- describe that Texas stood out in our spatial variation analysis and that we wanted to dig deeper into Texas's share of uninsured population over time from 2010 until 2023, except for 2020 and that we wanted to see if economic growth played a role in this.

```

#Here I am making a sub dataframe for the temporal visualization with the GDP growth rate of Texas and

#subset Texas share of uninsured population
texas_share = all_states_100 %>%
  filter(state == "Texas",
         year %in% 2010:2023,
         series == "Uninsured.Share")

```



```

#cleaning out the NA rows of subset Texas
texas_share = texas_share[-c(2:9),]
texas_share = texas_share[-c(3:10),]
texas_share = texas_share[-c(4:11),]
texas_share = texas_share[-c(5:12),]
texas_share = texas_share[-c(6:13),]
texas_share = texas_share[-c(7:14),]
texas_share = texas_share[-c(8:15),]
texas_share = texas_share[-c(9:16),]
texas_share = texas_share[-c(10:17),]
texas_share = texas_share[-c(11:18),]
texas_share = texas_share[-c(12:19),]
texas_share = texas_share[-c(13:20),]
texas_share = texas_share[-c(14:21),]

#subset GDP growth of texas of the years 2010-2023 except for 2020
library(dplyr)
texas_gdp = gdp_growth %>%
  filter(GeoName == "Texas",
         year %in% 2010:2023) %>%
  select(year, growth_rate)

```

```
## Adding missing grouping variables: `GeoName`
```

```

#Here I will plot the temporal visualization of Texas over the years 2010-2023 except for year 2020

#the subsets I will use are
View(texas_share)
View(texas_gdp)

library(ggplot2)
library(dplyr)

#Add a type column to distinguish between GDP growth rate and Uninsured Share
texas_gdp$type = "GDP growth rate"
texas_share$type = "Uninsured Share"

#rename the columns to match
texas_share = texas_share %>%
  rename(gdp = value)

#combine the 2 dataframes
combined_data_texas = bind_rows(texas_gdp %>%
  select(year, gdp = growth_rate, type),
  texas_share %>%
  select(year, gdp, type))

```

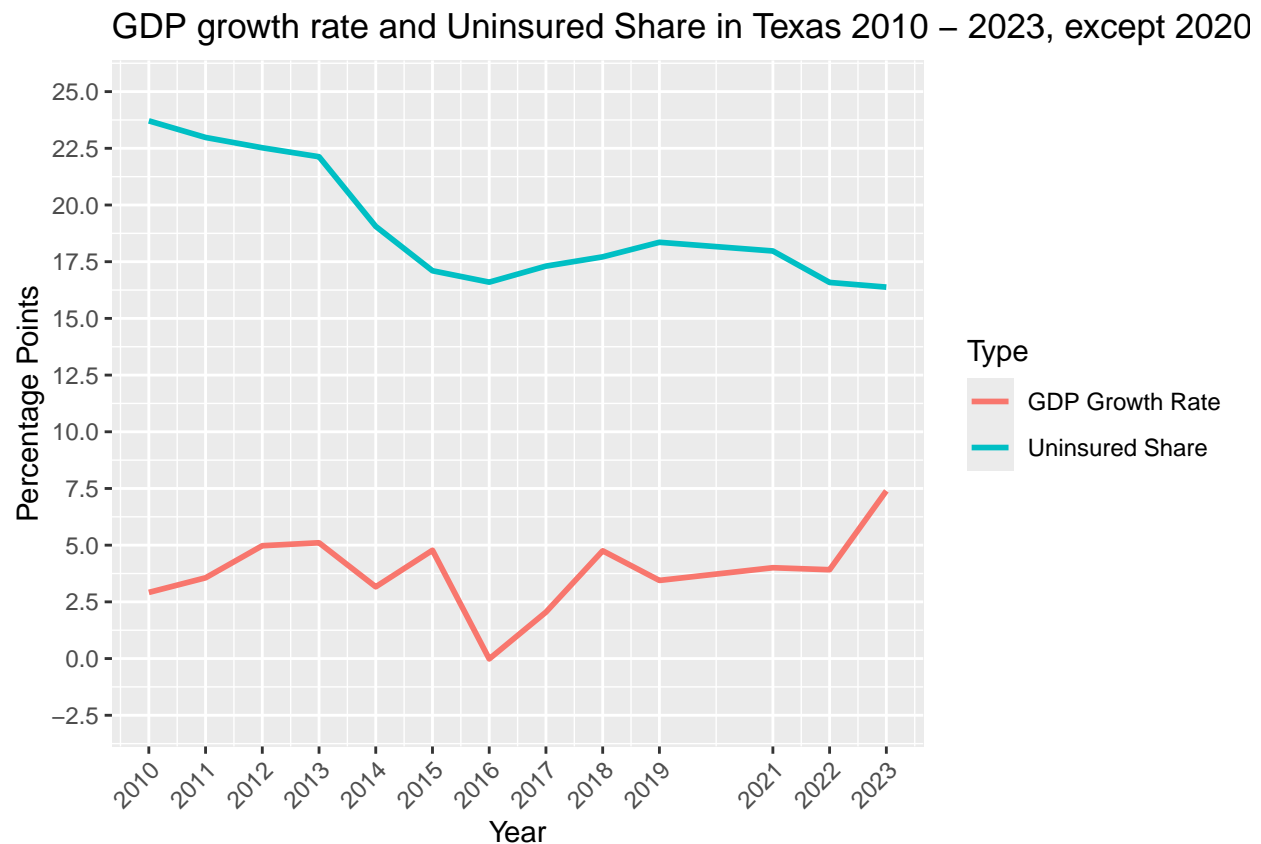
```
## Adding missing grouping variables: `GeoName`
```

```
#####
```

```
#plotting the lines from the 2 dataframes
```

```
ggplot() +
  geom_line(data = texas_gdp, aes(x = year, y = growth_rate, color = "GDP Growth Rate"), size = 1) +
  geom_line(data = texas_share, aes(x = year, y = gdp, color = "Uninsured Share"), size = 1) +
  labs(title = "GDP growth rate and Uninsured Share in Texas 2010 - 2023, except 2020", x = "Year", y =
  scale_y_continuous(breaks = seq(-2.5, 25, by = 2.5), limits = c(-2.5, 25)) +
  scale_x_continuous(breaks = c(2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2021, 2022, 2023)) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



Sub-Group Variation Visualization, The Uninsured Rates of the Middle-Income Group in 2017 in the U.S.

- describe that already the KFF (2024) found that the uninsured people in the US are more likely to be low income and that we were wondering what the state of the middle-income group is as they earn too much for Medicaid but not enough for private insurance. 2017 is the year after Trump became president in 2016.

Event Analysis Visualization, The Impact of The Implementation of Work Requirements for Medicaid in Arkansas, 2018

- describe here what happened in Arkansas in 2018 and why they wanted to implement the work requirements and what these requirements were. and that Arkansas is the treatment group and the other 43 states are the control group and they were analysing if the implementation had significant impact on the uninsured rates compared to the change in the control group which is the weighted average of the uninsured rates.

2.2 3.2 Identify Trends and Patterns

The uninsured rate is higher among younger adults, Hispanic and Black populations, and those with lower income or educational attainment.

3 Part 4 – Communicate Findings

3.1 4.1 Summarize Key Insights

Our analysis highlights systemic inequalities in access to health insurance. Despite overall improvements since the ACA, millions remain uninsured.

3.2 4.2 Propose Solutions or Policy Recommendations

Potential solutions include:

- Expanding Medicaid in all states
- Decoupling health insurance from employment
- Increasing subsidies for marketplace plans

4 Appendix

4.1 A.1 References

- U.S. Census Bureau. (2022). Health Insurance Coverage in the United States.
- Kaiser Family Foundation. (2023). Key Facts about the Uninsured Population.

4.2 A.2 Session Info