final-project

```
library(tidyverse)
```

```
── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
✔ dplyr     1.1.4     ✔ readr     2.1.5
✔ forcats   1.0.0     ✔ stringr   1.5.1
✔ ggplot2   3.5.1     ✔ tibble    3.2.1
✔ lubridate 1.9.3     ✔ tidyr     1.3.1
✔ purrr     1.0.2     
── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
✖ dplyr::filter() masks stats::filter()
✖ dplyr::lag()    masks stats::lag()
ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts
to become errors
```

```
library(janitor)
```

```

Attaching package: 'janitor'

The following objects are masked from 'package:stats':

    chisq.test, fisher.test
```

```
library(stringr)
library(dplyr)
library(httr2)
```

```
Warning: package 'httr2' was built under R version 4.4.1
```

```
census_key <- "2abeb09fab2a060893dafc5545972f25d26b0fb3"
url <- "https://api.census.gov/data/2021/pep/population"

request <- request(url) |>
  req_url_query(
    get = I("POP_2020,POP_2021,NAME"),
    `for` = I("state:*"),
    key = census_key
  )
```

```
response <- request |> req_perform()
pop_2021 <- response |>
  resp_body_json(simplifyVector = TRUE) |>
  as_tibble()
```

```
Warning: The `x` argument of `as_tibble.matrix()` must have unique column names if
`.name_repair` is omitted as of tibble 2.0.0.
ℹ Using compatibility `.name_repair`.
```

```
pop_2021
```

```
# A tibble: 53 × 4
   V1       V2       V3           V4   
   <chr>    <chr>    <chr>        <chr>
 1 POP_2020 POP_2021 NAME         state
 2 3962031  3986639  Oklahoma     40   
 3 1961455  1963692  Nebraska     31   
 4 1451911  1441553  Hawaii       15   
 5 887099   895376   South Dakota 46   
 6 6920119  6975218  Tennessee    47   
 7 3114071  3143991  Nevada       32   
 8 2117566  2115877  New Mexico   35   
 9 3188669  3193079  Iowa         19   
10 2935880  2934582  Kansas       20   
# ℹ 43 more rows
```

```
str(pop_2021)
```

```
tibble [53 × 4] (S3: tbl_df/tbl/data.frame)
 $ V1: chr [1:53] "POP_2020" "3962031" "1961455" "1451911" ...
 $ V2: chr [1:53] "POP_2021" "3986639" "1963692" "1441553" ...
 $ V3: chr [1:53] "NAME" "Oklahoma" "Nebraska" "Hawaii" ...
 $ V4: chr [1:53] "state" "40" "31" "15" ...
```

```
class(pop_2021)
```

```
[1] "tbl_df"     "tbl"        "data.frame"
```

```
pop_2021_new <- pop_2021 |> row_to_names(row_number = 1)|>
  select(-state)|>
  # rename state column to state_name
  rename(state_name = NAME) |>
  # use pivot_longer to tidy
  pivot_longer(-state_name,
               names_to = "year",
               values_to = "population")|>
  # remove POP_ from year
  mutate(
    year = str_remove(year, "POP_"),
      # parese all relevant colunns to numeric
    year = as.numeric(year),
    population = as.numeric(population),
      # add state abbreviations using state.abb variable
      # use case_when to add abbreviations for DC and PR
    state = case_when(
      state_name == "District of Columbia" ~"DC",
      state_name == "Puerto Rico" ~"PR",
      TRUE ~ state.abb[match(state_name, state.name)]
    )
  )|> filter(year %in% c(2020, 2021)) |>
```

```
    arrange(state_name, population)
pop_2021_new
```

```
# A tibble: 104 × 4
   state_name  year population state
   <chr>      <dbl>      <dbl> <chr>
 1 Alabama     2020    5024803 AL
 2 Alabama     2021    5039877 AL
 3 Alaska      2020     732441 AK
 4 Alaska      2021     732673 AK
 5 Arizona     2020    7177986 AZ
 6 Arizona     2021    7276316 AZ
 7 Arkansas    2020    3012232 AR
 8 Arkansas    2021    3025891 AR
 9 California  2021   39237836 CA
10 California  2020   39499738 CA
# ℹ 94 more rows
```

#Getting population data for 2022-23

```
# Import the new population data
# https://www.census.gov/data/datasets/time-series/demo/popest/2020s-state-tot

population_new_raw <- read.csv("./data/raw/NST-EST2023-ALLDATA.csv")

# View the first few rows of the dataset
head(population_new_raw)
```

```
  SUMLEV REGION DIVISION STATE               NAME ESTIMATESBASE2020
1     10      0        0     0      United States         331464948
2     20      1        0     0   Northeast Region          57614141
3     30      1        1     0        New England          15119994
4     30      1        2     0    Middle Atlantic          42494147
5     20      2        0     0     Midwest Region          68987296
6     30      2        3     0 East North Central          47369629
  POPESTIMATE2020 POPESTIMATE2021 POPESTIMATE2022 POPESTIMATE2023 NPOPCHG_2020
1       331526933       332048977       333271411       334914895        61985
2        57430477        57243423        57026847        56983517      -183664
3        15057898        15106108        15120739        15159777       -62096
4        42372579        42137315        41906108        41823740      -121568
5        68969794        68850246        68783028        68909283       -17502
6        47345074        47187461        47098310        47146039       -24555
  NPOPCHG_2021 NPOPCHG_2022 NPOPCHG_2023 BIRTHS2020 BIRTHS2021 BIRTHS2022
1       522044      1222434      1643484     894123    3584459    3679254
2      -187054      -216576       -43330     146099     572860     588927
3        48210        14631        39038      35418     139200     144753
4      -235264      -231207       -82368     110681     433660     444174
5      -119548       -67218       126255     190125     748083     753976
6      -157613       -89151        47729     127370     500704     503757
  BIRTHS2023 DEATHS2020 DEATHS2021 DEATHS2022 DEATHS2023 NATURALCHG2020
1    3653356     852024    3438423    3456087    3148861          42099
2     581516     193163     560547     563354     525863         -47064
3     142522      46210     143827     149344     142818         -10792
4     438994     146953     416720     414010     383045         -36272
```

```
5     746365     186179     762461     771652     700527            3946
6     497398     133435     530262     537410     484782           -6065
  NATURALCHG2021 NATURALCHG2022 NATURALCHG2023 INTERNATIONALMIG2020
1         146036         223167         504495                19886
2          12313          25573          55653                 4432
3          -4627          -4591           -296                 1562
4          16940          30164          55949                 2870
5         -14378         -17676          45838                 3074
6         -29558         -33653          12616                 1988
  INTERNATIONALMIG2021 INTERNATIONALMIG2022 INTERNATIONALMIG2023
1               376008               999267              1138989
2                80448               210145               225009
3                26735                68504                76068
4                53713               141641               148941
5                55313               144422               165910
6                37025                97108               122912
  DOMESTICMIG2020 DOMESTICMIG2021 DOMESTICMIG2022 DOMESTICMIG2023 NETMIG2020
1               0               0               0               0      19886
2         -131531         -276548         -450321         -323300    -127099
3          -46076           24369          -46644          -37031     -44514
4          -85455         -300917         -403677         -286269     -82585
5          -35580         -177584         -181443          -85729     -32506
6          -28662         -176319         -143450          -88006     -26674
  NETMIG2021 NETMIG2022 NETMIG2023 RESIDUAL2020 RESIDUAL2021 RESIDUAL2022
1     376008     999267    1138989            0            0            0
2    -196100    -240176     -98291        -9501        -3267        -1973
3      51104      21860      39037        -6790         1733        -2638
4    -247204    -262036    -137328        -2711        -5000          665
5    -122271     -37021      80181        11058        17101       -12521
6    -139294     -46342      34906         8184        11239        -9156
  RESIDUAL2023 RBIRTH2021 RBIRTH2022 RBIRTH2023 RDEATH2021 RDEATH2022
1            0  10.803463  11.060097  10.935142  10.363315  10.389241
2         -692   9.991114  10.307615  10.201108   9.776366   9.860027
3          297   9.229543   9.577777   9.413446   9.536333   9.881547
4         -989  10.262940  10.570107  10.485962   9.862041   9.852288
5          236  10.855939  10.956304  10.841056  11.064588  11.213161
6          207  10.593263  10.685748  10.555498  11.218614  11.399599
  RDEATH2023 RNATURALCHG2021 RNATURALCHG2022 RNATURALCHG2023
1   9.425099       0.4401486       0.6708557      1.51004292
2   9.224828       0.2147481       0.4475880      0.97627967
3   9.432996      -0.3067895      -0.3037697     -0.01955052
4   9.149545       0.4008998       0.7178194      1.33641709
5  10.175252      -0.2086489      -0.2568565      0.66580334
6  10.287768      -0.6253508      -0.7138511      0.26772958
  RINTERNATIONALMIG2021 RINTERNATIONALMIG2022 RINTERNATIONALMIG2023
1             1.1332780              3.003867              3.409196
2             1.4030743              3.678035              3.947167
3             1.7726425              4.532659              5.024221
4             1.2711648              3.370662              3.557656
5             0.8026844              2.098649              2.409866
6             0.7833282              2.059865              2.608369
  RDOMESTICMIG2021 RDOMESTICMIG2022 RDOMESTICMIG2023 RNETMIG2021 RNETMIG2022
1         0.000000         0.000000         0.000000    1.133278   3.0038671
2        -4.823207        -7.881683        -5.671414   -3.420133  -4.2036481
```

```
3         1.615767        -3.086263        -2.445863    3.388409   1.4463963
4        -7.121462        -9.606391        -6.837920   -5.850297  -6.2357289
5        -2.577042        -2.636615        -1.245226   -1.774357  -0.5379658
6        -3.730335        -3.042877        -1.867613   -2.947007  -0.9830115
  RNETMIG2023
1   3.4091959
2  -1.7242468
3   2.5783576
4  -3.2802639
5   1.1646402
6   0.7407553
```

```
#Wrangle the data

population_2223_clean <- population_new_raw |>
  filter(SUMLEV == 40) |>  # Keep only state-level data
  select("NAME", "POPESTIMATE2022", "POPESTIMATE2023") |>  # Select relevant c
  rename(
    state_name = NAME,       # Rename NAME to state_name
    `2022` = POPESTIMATE2022, # Rename population columns for clarity
    `2023` = POPESTIMATE2023
  ) |>
  pivot_longer(
    cols = `2022`:`2023`,  # Convert population columns to long format
    names_to = "year",
    values_to = "population"
  ) |>
  mutate(
    year = as.numeric(year), # Ensure year is numeric
    population = as.numeric(population) # Ensure population is numeric
  )

# Print cleaned dataset
print(population_2223_clean)
```

```
# A tibble: 104 × 3
   state_name  year population
   <chr>      <dbl>      <dbl>
 1 Alabama     2022    5073903
 2 Alabama     2023    5108468
 3 Alaska      2022     733276
 4 Alaska      2023     733406
 5 Arizona     2022    7365684
 6 Arizona     2023    7431344
 7 Arkansas    2022    3046404
 8 Arkansas    2023    3067732
 9 California  2022   39040616
10 California  2023   38965193
# ℹ 94 more rows
```

```
full_population <- bind_rows(pop_2021_new, population_2223_clean) |>
  arrange(state_name, year)|>
  mutate(
    state = case_when(
      state_name == "District of Columbia" ~ "DC",
      state_name == "Puerto Rico" ~ "PR",
      is.na(state) ~ state.abb[match(state_name, state.name)],
      TRUE ~ state
    )
  )

# Print the combined dataset
print(full_population)
```

```
# A tibble: 208 × 4
   state_name  year population state
   <chr>      <dbl>      <dbl> <chr>
 1 Alabama     2020    5024803 AL
 2 Alabama     2021    5039877 AL
 3 Alabama     2022    5073903 AL
 4 Alabama     2023    5108468 AL
 5 Alaska      2020     732441 AK
 6 Alaska      2021     732673 AK
 7 Alaska      2022     733276 AK
 8 Alaska      2023     733406 AK
 9 Arizona     2020    7177986 AZ
10 Arizona     2021    7276316 AZ
# ℹ 198 more rows
```

#Download covid case data

```
api <- "https://data.cdc.gov/resource/pwn4-m3yp.json"
response <- request(api) |>
  req_url_query(`$limit` = 10000000000) |>
  req_perform()

cases_raw <- response |>
  resp_body_json()|>
  map_df(~ as_tibble(.))
```

```
# wrangle covid case data
cases_clean <- cases_raw |>
  select(state, end_date, new_cases) |>
  rename(date = end_date, cases = new_cases) |>
  mutate(
    cases = as.numeric(cases),
    date = as_date(ymd_hms(date))
  ) |>
  mutate(mmwr_week = epiweek(date), mmwr_year = epiyear(date)) |>
select(state, mmwr_year, mmwr_week, cases) |>
arrange(state, mmwr_year, mmwr_week)

head(cases_clean)
```

```
# A tibble: 6 × 4
  state mmwr_year mmwr_week cases
  <chr>     <dbl>     <dbl> <dbl>
1 AK         2020         4     0
2 AK         2020         5     0
3 AK         2020         6     0
4 AK         2020         7     0
5 AK         2020         8     0
6 AK         2020         9     0
```

Get covid death and hospitalisation data

```
get_cdc_data <- function(api){
  request(api) |>
  req_url_query("$limit" = 10000000) |>
  req_perform() |>
  resp_body_json(simplifyVector = TRUE)
}

hosp_raw <- get_cdc_data("https://data.cdc.gov/resource/39z2-9zu6.json")
deaths_raw <- get_cdc_data("https://data.cdc.gov/resource/r8kw-7aab.json")
vax_raw <- get_cdc_data("https://data.cdc.gov/resource/rh2h-3yt2.json")
```

Wrangle the above data

```
# Death
deaths <- deaths_raw |>
  filter(state %in% full_population$state_name) |>
  mutate(end_date = as_date(end_date),mmwr_year = epiyear(end_date)) |>
  rename(deaths_prov = covid_19_deaths,flu = influenza_deaths) |>
  mutate(mmwr_week = parse_number(mmwr_week),deaths = parse_number(deaths_prov
  filter(mmwr_year %in% c("2020", "2021","2022","2023", "2024"))|>
  select(state, mmwr_week, mmwr_year, deaths)

head(deaths)
```

```
    state mmwr_week mmwr_year deaths
1 Alabama         1      2020      0
2 Alabama         2      2020      0
3 Alabama         3      2020      0
4 Alabama         4      2020     NA
5 Alabama         5      2020      0
6 Alabama         6      2020      0
```

```
# hospitalisation
hosp <- hosp_raw |>
filter(jurisdiction %in% full_population$state) |>
rename(hosp = new_covid_19_hospital, state = jurisdiction) |>
mutate(hosp = parse_number(hosp),
date = as_date(ymd_hms(collection_date)),
mmwr_week = epiweek(date), mmwr_year = epiyear(date)) |>
select(state, mmwr_year, mmwr_week, hosp) |>
```

```
group_by(state, mmwr_year, mmwr_week) |>
summarize(hosp = sum(hosp), n = n(), .groups = "drop") |>
filter(n == 7) |>
select(-n) |>
arrange(mmwr_year, mmwr_week)

head(hosp)
```

```
# A tibble: 6 × 4
  state mmwr_year mmwr_week  hosp
  <chr>     <dbl>     <dbl> <dbl>
1 AK         2020        32    28
2 AL         2020        32   664
3 AR         2020        32   449
4 AZ         2020        32   760
5 CA         2020        32  4682
6 CO         2020        32   316
```

```
# vaccination
vax <- vax_raw |> filter(date_type == "Admin" & location %in% full_population$
rename(state = location, series_complete = series_complete_cumulative,
booster = booster_cumulative) |>
mutate(date = as_date(ymd_hms(date)),
mmwr_week = as.numeric(mmwr_week), mmwr_year = epiyear(date),
series_complete = parse_number(series_complete),
booster = parse_number(booster)) |>
select(state, date, mmwr_week, mmwr_year, series_complete, booster) |>
group_by(state, mmwr_week, mmwr_year) |>
summarize(series_complete = max(series_complete),
booster = max(booster),.groups = "drop") |>
arrange(state, mmwr_year, mmwr_week)

head(vax)
```

```
# A tibble: 6 × 5
  state mmwr_week mmwr_year series_complete booster
  <chr>     <dbl>     <dbl>           <dbl>   <dbl>
1 AK           51      2020              46       0
2 AK           52      2020              69       0
3 AK           53      2020             114       0
4 AK            1      2021            8396       0
5 AK            2      2021           13560       0
6 AK            3      2021           20111       0
```

Make dates data frame

```
all_dates <- data.frame(date = seq(make_date(2020, 1, 25),
make_date(2024, 12, 31),
by = "week")) |>
mutate(date = ceiling_date(date, unit = "week", week_start = 7) - days(1))|>
mutate(mmwr_year = epiyear(date), mmwr_week = epiweek(date))
dates_and_pop <- cross_join(all_dates, data.frame(state =
unique(full_population$state))) |> left_join(full_population, by = c("state",
"mmwr_year" = "year"))

all_dates
```

```
          date mmwr_year mmwr_week
1   2020-01-25      2020         4
2   2020-02-01      2020         5
3   2020-02-08      2020         6
4   2020-02-15      2020         7
5   2020-02-22      2020         8
6   2020-02-29      2020         9
7   2020-03-07      2020        10
8   2020-03-14      2020        11
9   2020-03-21      2020        12
10  2020-03-28      2020        13
11  2020-04-04      2020        14
12  2020-04-11      2020        15
13  2020-04-18      2020        16
14  2020-04-25      2020        17
15  2020-05-02      2020        18
16  2020-05-09      2020        19
17  2020-05-16      2020        20
18  2020-05-23      2020        21
19  2020-05-30      2020        22
20  2020-06-06      2020        23
21  2020-06-13      2020        24
22  2020-06-20      2020        25
23  2020-06-27      2020        26
24  2020-07-04      2020        27
25  2020-07-11      2020        28
26  2020-07-18      2020        29
27  2020-07-25      2020        30
28  2020-08-01      2020        31
29  2020-08-08      2020        32
30  2020-08-15      2020        33
31  2020-08-22      2020        34
32  2020-08-29      2020        35
33  2020-09-05      2020        36
34  2020-09-12      2020        37
35  2020-09-19      2020        38
36  2020-09-26      2020        39
37  2020-10-03      2020        40
38  2020-10-10      2020        41
39  2020-10-17      2020        42
40  2020-10-24      2020        43
41  2020-10-31      2020        44
42  2020-11-07      2020        45
43  2020-11-14      2020        46
44  2020-11-21      2020        47
45  2020-11-28      2020        48
46  2020-12-05      2020        49
47  2020-12-12      2020        50
48  2020-12-19      2020        51
```

```
    49  2020-12-26      2020        52
    50  2021-01-02      2020        53
    51  2021-01-09      2021         1
    52  2021-01-16      2021         2
    53  2021-01-23      2021         3
    54  2021-01-30      2021         4
    55  2021-02-06      2021         5
    56  2021-02-13      2021         6
    57  2021-02-20      2021         7
    58  2021-02-27      2021         8
    59  2021-03-06      2021         9
    60  2021-03-13      2021        10
    61  2021-03-20      2021        11
    62  2021-03-27      2021        12
    63  2021-04-03      2021        13
    64  2021-04-10      2021        14
    65  2021-04-17      2021        15
    66  2021-04-24      2021        16
    67  2021-05-01      2021        17
    68  2021-05-08      2021        18
    69  2021-05-15      2021        19
    70  2021-05-22      2021        20
    71  2021-05-29      2021        21
    72  2021-06-05      2021        22
    73  2021-06-12      2021        23
    74  2021-06-19      2021        24
    75  2021-06-26      2021        25
    76  2021-07-03      2021        26
    77  2021-07-10      2021        27
    78  2021-07-17      2021        28
    79  2021-07-24      2021        29
    80  2021-07-31      2021        30
    81  2021-08-07      2021        31
    82  2021-08-14      2021        32
    83  2021-08-21      2021        33
    84  2021-08-28      2021        34
    85  2021-09-04      2021        35
    86  2021-09-11      2021        36
    87  2021-09-18      2021        37
    88  2021-09-25      2021        38
    89  2021-10-02      2021        39
    90  2021-10-09      2021        40
    91  2021-10-16      2021        41
    92  2021-10-23      2021        42
    93  2021-10-30      2021        43
    94  2021-11-06      2021        44
    95  2021-11-13      2021        45
    96  2021-11-20      2021        46
    97  2021-11-27      2021        47
    98  2021-12-04      2021        48
    99  2021-12-11      2021        49
    100 2021-12-18      2021        50
    101 2021-12-25      2021        51
    102 2022-01-01      2021        52
    103 2022-01-08      2022          1
    104 2022-01-15      2022          2
    105 2022-01-22      2022          3
    106 2022-01-29      2022          4
    107 2022-02-05      2022          5
    108 2022-02-12      2022          6
    109 2022-02-19      2022          7
    110 2022-02-26      2022          8
    111 2022-03-05      2022          9
    112 2022-03-12      2022         10
    113 2022-03-19      2022         11
    114 2022-03-26      2022         12
    115 2022-04-02      2022         13
    116 2022-04-09      2022         14
    117 2022-04-16      2022         15
    118 2022-04-23      2022         16
    119 2022-04-30      2022         17
    120 2022-05-07      2022         18
    121 2022-05-14      2022         19
    122 2022-05-21      2022         20
    123 2022-05-28      2022         21
    124 2022-06-04      2022         22
    125 2022-06-11      2022         23
    126 2022-06-18      2022         24
    127 2022-06-25      2022         25
    128 2022-07-02      2022         26
    129 2022-07-09      2022         27
    130 2022-07-16      2022         28
    131 2022-07-23      2022         29
    132 2022-07-30      2022         30
    133 2022-08-06      2022         31
    134 2022-08-13      2022         32
    135 2022-08-20      2022         33
    136 2022-08-27      2022         34
    137 2022-09-03      2022         35
    138 2022-09-10      2022         36
    139 2022-09-17      2022         37
    140 2022-09-24      2022         38
    141 2022-10-01      2022         39
    142 2022-10-08      2022         40
    143 2022-10-15      2022         41
    144 2022-10-22      2022         42
    145 2022-10-29      2022         43
    146 2022-11-05      2022         44
    147 2022-11-12      2022         45
    148 2022-11-19      2022         46
    149 2022-11-26      2022         47
    150 2022-12-03      2022         48
    151 2022-12-10      2022         49
    152 2022-12-17      2022         50
    153 2022-12-24      2022         51
    154 2022-12-31      2022         52
    155 2023-01-07      2023          1
    156 2023-01-14      2023          2
```

```
157 2023-01-21      2023         3
158 2023-01-28      2023         4
159 2023-02-04      2023         5
160 2023-02-11      2023         6
161 2023-02-18      2023         7
162 2023-02-25      2023         8
163 2023-03-04      2023         9
164 2023-03-11      2023        10
165 2023-03-18      2023        11
166 2023-03-25      2023        12
167 2023-04-01      2023        13
168 2023-04-08      2023        14
169 2023-04-15      2023        15
170 2023-04-22      2023        16
171 2023-04-29      2023        17
172 2023-05-06      2023        18
173 2023-05-13      2023        19
174 2023-05-20      2023        20
175 2023-05-27      2023        21
176 2023-06-03      2023        22
177 2023-06-10      2023        23
178 2023-06-17      2023        24
179 2023-06-24      2023        25
180 2023-07-01      2023        26
181 2023-07-08      2023        27
182 2023-07-15      2023        28
183 2023-07-22      2023        29
184 2023-07-29      2023        30
185 2023-08-05      2023        31
186 2023-08-12      2023        32
187 2023-08-19      2023        33
188 2023-08-26      2023        34
189 2023-09-02      2023        35
190 2023-09-09      2023        36
191 2023-09-16      2023        37
192 2023-09-23      2023        38
193 2023-09-30      2023        39
194 2023-10-07      2023        40
195 2023-10-14      2023        41
196 2023-10-21      2023        42
197 2023-10-28      2023        43
198 2023-11-04      2023        44
199 2023-11-11      2023        45
200 2023-11-18      2023        46
201 2023-11-25      2023        47
202 2023-12-02      2023        48
203 2023-12-09      2023        49
204 2023-12-16      2023        50
205 2023-12-23      2023        51
206 2023-12-30      2023        52
207 2024-01-06      2024         1
208 2024-01-13      2024         2
209 2024-01-20      2024         3
210 2024-01-27      2024         4
```

```
211 2024-02-03      2024         5
212 2024-02-10      2024         6
213 2024-02-17      2024         7
214 2024-02-24      2024         8
215 2024-03-02      2024         9
216 2024-03-09      2024        10
217 2024-03-16      2024        11
218 2024-03-23      2024        12
219 2024-03-30      2024        13
220 2024-04-06      2024        14
221 2024-04-13      2024        15
222 2024-04-20      2024        16
223 2024-04-27      2024        17
224 2024-05-04      2024        18
225 2024-05-11      2024        19
226 2024-05-18      2024        20
227 2024-05-25      2024        21
228 2024-06-01      2024        22
229 2024-06-08      2024        23
230 2024-06-15      2024        24
231 2024-06-22      2024        25
232 2024-06-29      2024        26
233 2024-07-06      2024        27
234 2024-07-13      2024        28
235 2024-07-20      2024        29
236 2024-07-27      2024        30
237 2024-08-03      2024        31
238 2024-08-10      2024        32
239 2024-08-17      2024        33
240 2024-08-24      2024        34
241 2024-08-31      2024        35
242 2024-09-07      2024        36
243 2024-09-14      2024        37
244 2024-09-21      2024        38
245 2024-09-28      2024        39
246 2024-10-05      2024        40
247 2024-10-12      2024        41
248 2024-10-19      2024        42
249 2024-10-26      2024        43
250 2024-11-02      2024        44
251 2024-11-09      2024        45
252 2024-11-16      2024        46
253 2024-11-23      2024        47
254 2024-11-30      2024        48
255 2024-12-07      2024        49
256 2024-12-14      2024        50
257 2024-12-21      2024        51
258 2024-12-28      2024        52
```

Combine the above data frame

```
        # get deaths dataset a state column
        # Create a state mapping table
```

```
state_mapping <- tibble(
  state_name = state.name,
  state_abbr = state.abb
)

# Add entries for DC and Puerto Rico if not present
state_mapping <- state_mapping |>
  add_row(state_name = "District of Columbia", state_abbr = "DC") |>
  add_row(state_name = "Puerto Rico", state_abbr = "PR")

# Add state abbreviations to deaths dataset
deaths <- deaths |>
  left_join(state_mapping, by = c("state" = "state_name"))
```

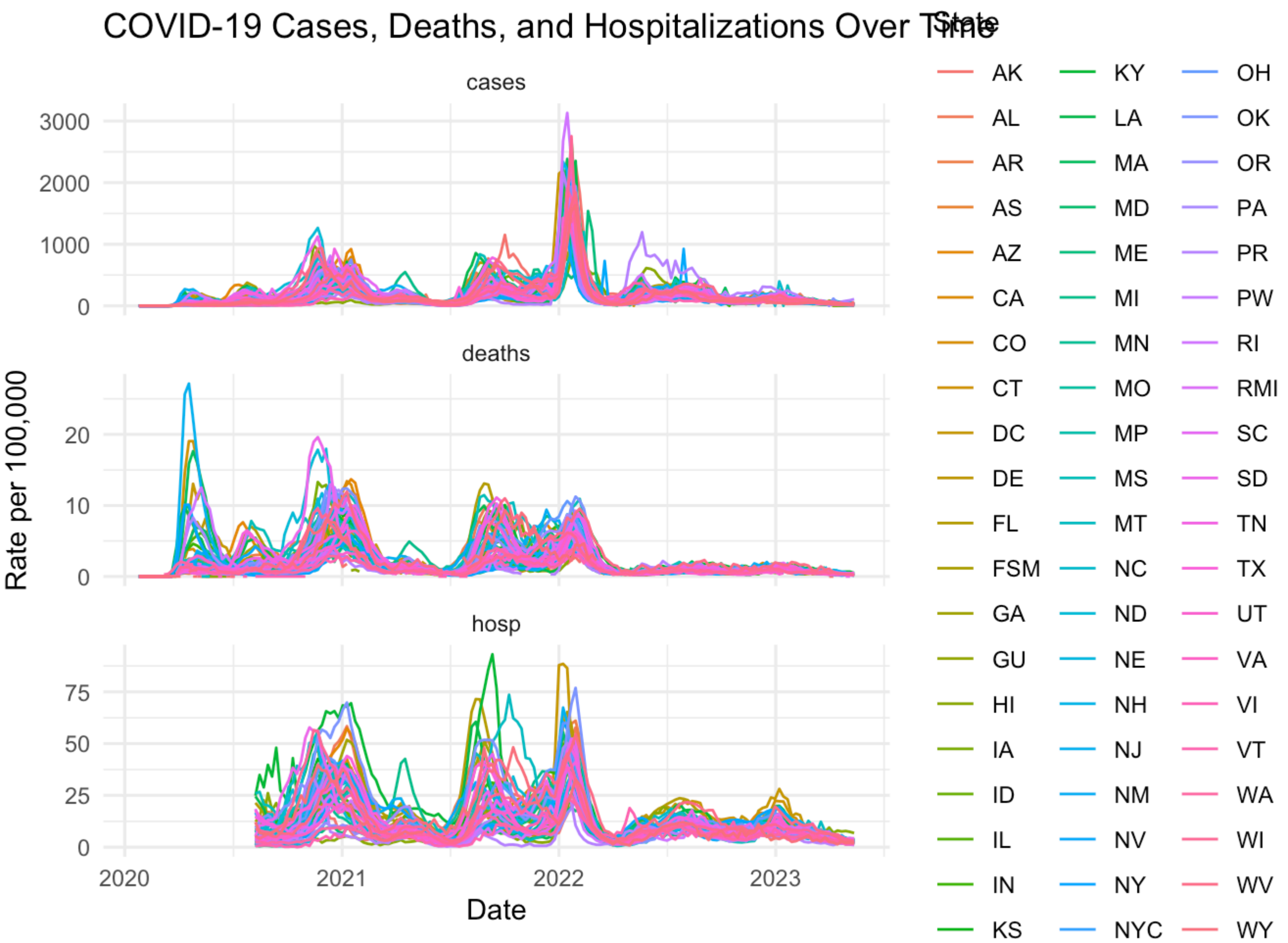
```
# Use the updated deaths dataset with state abbreviations for joining
dat <- cases_clean |>
  left_join(deaths, by = c("state"= "state_abbr", "mmwr_year", "mmwr_week")) |
  left_join(dates_and_pop, by = c("state", "mmwr_year", "mmwr_week")) |>
  left_join(hosp, by = c("state", "mmwr_year", "mmwr_week")) |>
  left_join(vax, by = c("state", "mmwr_year", "mmwr_week"))

head(dat)
```

```
# A tibble: 6 × 12
  state mmwr_year mmwr_week cases state.y deaths date       state_name
  <chr>     <dbl>     <dbl> <dbl> <chr>    <dbl> <date>     <chr>
1 AK         2020         4     0 Alaska       0 2020-01-25 Alaska
2 AK         2020         5     0 Alaska       0 2020-02-01 Alaska
3 AK         2020         6     0 Alaska       0 2020-02-08 Alaska
4 AK         2020         7     0 Alaska       0 2020-02-15 Alaska
5 AK         2020         8     0 Alaska       0 2020-02-22 Alaska
6 AK         2020         9     0 Alaska       0 2020-02-29 Alaska
# ℹ 4 more variables: population <dbl>, hosp <dbl>, series_complete <dbl>,
#   booster <dbl>
```

Q1 - Divide the pandemic period, January 2020 to December 2024 into waves. Justify your choice with data visualization.

```
# Calculate rates and reshape the dataset
p <- dat |>
  mutate(
    cases = cases / population * 100000,      # Calculate cases per 100,000
    hosp = hosp / population * 100000,        # Calculate hospitalizations per
    deaths = deaths / population * 100000     # Calculate deaths per 100,000
  ) |>
  select(date, cases, hosp, deaths, state) |>  # Select relevant columns
  pivot_longer(
    cols = c(cases, deaths, hosp),            # Reshape the data
    names_to = "outcome",
    values_to = "rate"
  ) |>
  ggplot(aes(x = date, y = rate, color = state, group = state)) +
  geom_line() + # Add line plot
  facet_wrap(~outcome, nrow = 3, scales = "free_y") +
  labs(
    title = "COVID-19 Cases, Deaths, and Hospitalizations Over Time",
    x = "Date",
    y = "Rate per 100,000",
    color = "State"
  ) +
  theme_minimal()

# Print the plot
print(p)
```

```
Warning: Removed 4152 rows containing missing values or values outside the scale range
(`geom_line()`).
```

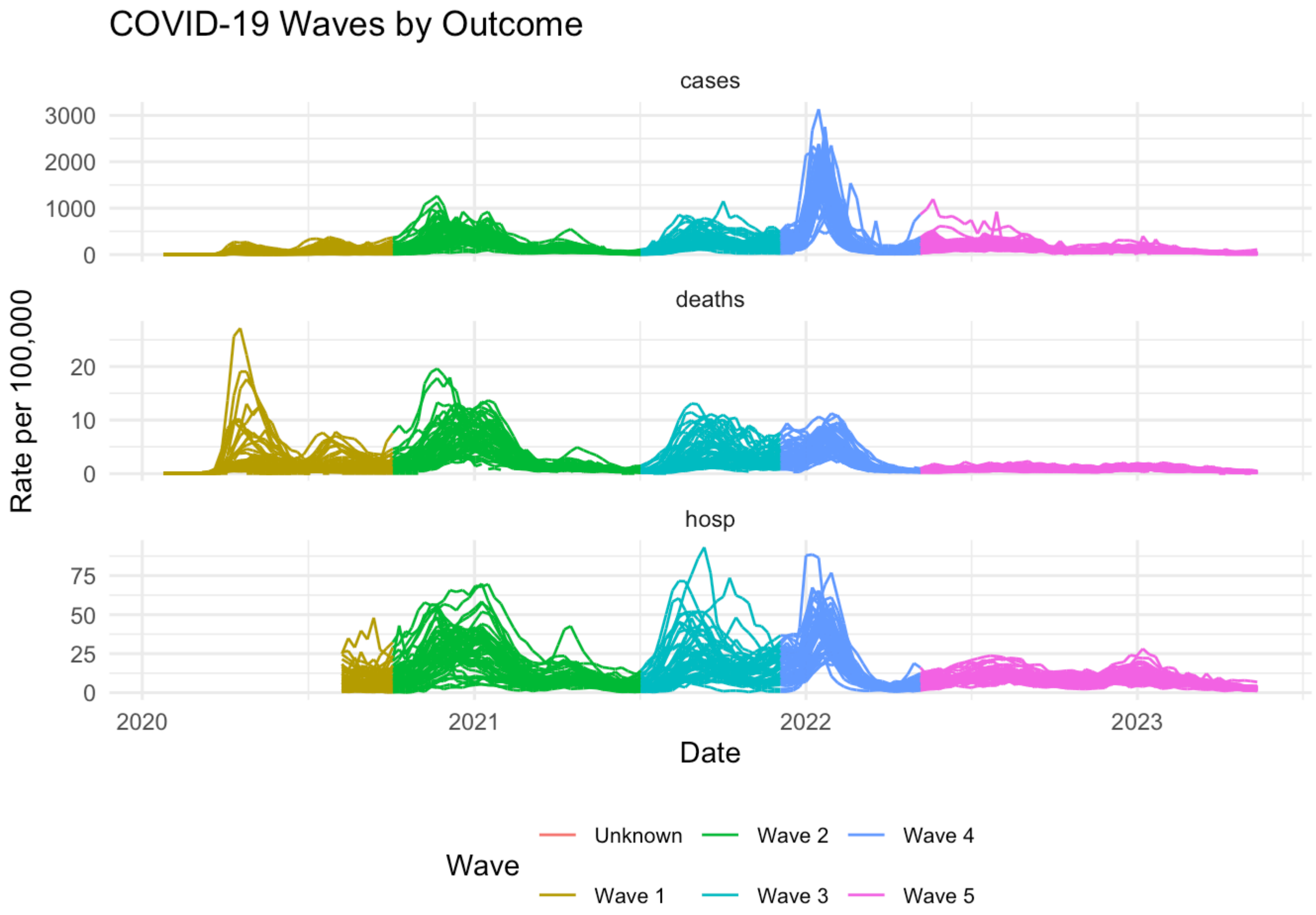


Segmentation of covid waves

```
dat_wave <- dat |>
  mutate(
    wave = case_when(
      # wave 1 is the initial outbreak globally
      date >= as.Date("2020-01-01") & date < as.Date("2020-10-01") ~ "Wave 1",
      # wave 2 is the surge in fall and winter, where indoor gathering and hol
      date >= as.Date("2020-10-02") & date < as.Date("2021-06-30") ~ "Wave 2",
      # wave 3 is when the contagious Delta variant began to circulate and eve
      date >= as.Date("2021-07-01") & date < as.Date("2021-11-30") ~ "Wave 3",
      # wave 4 is when Omicron BA.1 variant significantly increased cases numb
      date >= as.Date("2021-12-01") & date < as.Date("2022-05-01") ~ "Wave 4",
      # wave 5 captures small wave associated with Omicron subvariants like BA
      date >= as.Date("2022-05-02")  ~ "Wave 5",
      TRUE ~ "Unknown"
    )
  )
```

```
p_wave <- dat_wave |>
  mutate(
    cases = cases / population * 100000,
    hosp = hosp / population * 100000,
    deaths = deaths / population * 100000
  ) |>
  select(date, cases, hosp, deaths, state, wave) |>
  pivot_longer(
    cols = c(cases, deaths, hosp),
    names_to = "outcome",
    values_to = "rate"
  ) |>
  ggplot(aes(x = date, y = rate, color = wave, group = state)) +
  geom_line() +
  facet_wrap(~outcome, nrow = 3, scales = "free_y") +
  labs(
    title = "COVID-19 Waves by Outcome",
    x = "Date",
    y = "Rate per 100,000",
    color = "Wave"
  ) +
  theme_minimal()+
  theme(
    legend.position = "bottom",
    legend.text = element_text(size = 8)
  )

print(p_wave)
```

```
Warning: Removed 4152 rows containing missing values or values outside the scale range
(`geom_line()`).
```



Question 2 - For each period compute the deaths rates by state. Describe which states did better or worse during the different periods.

```
# Summarise total deaths and calculate death rates
death_rates_by_wave <- dat_wave |>
  group_by(state, wave) |>
  filter(wave != "Unknown")|>
  summarize(
    total_deaths = sum(deaths, na.rm = TRUE),
    total_population = mean(population, na.rm = TRUE)
  ) |>
  mutate(death_rate = (total_deaths / total_population) * 100000) |>
  arrange(wave, desc(death_rate))
```

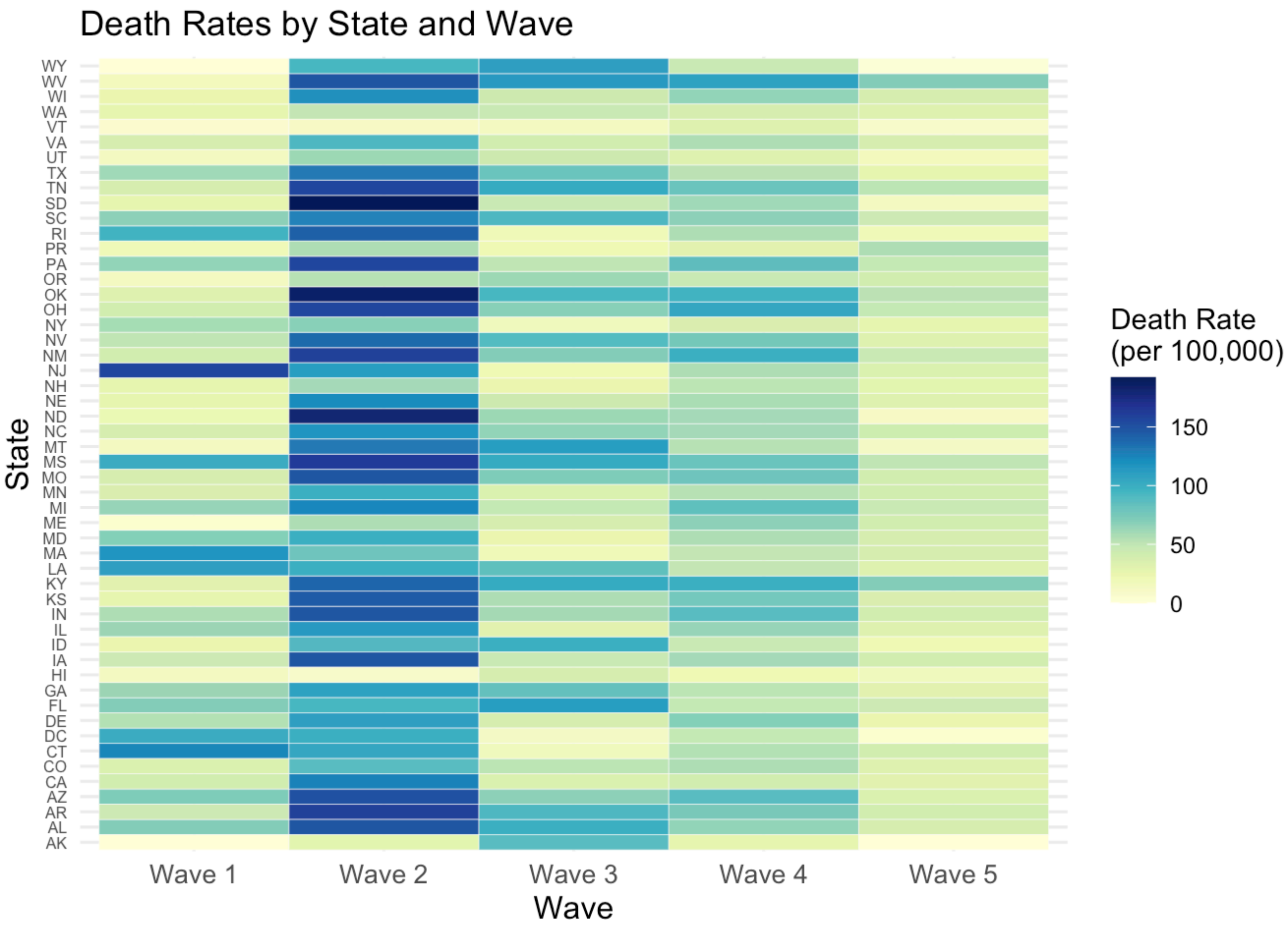
```
`summarise()` has grouped output by 'state'. You can override using the
`.groups` argument.
```

```
print(death_rates_by_wave)
```

```
# A tibble: 260 × 5
# Groups:   state [52]
   state wave   total_deaths total_population death_rate
   <chr> <chr>         <dbl>            <dbl>      <dbl>
```

```
 1 NJ    Wave 1        14529         9279743      157. 
 2 CT    Wave 1         4494         3600260      125. 
 3 MA    Wave 1         8156         7022220      116. 
 4 LA    Wave 1         5155         4651203      111. 
 5 MS    Wave 1         3012         2956870      102. 
 6 DC    Wave 1          699          690093      101. 
 7 RI    Wave 1         1039         1096229       94.8
 8 AZ    Wave 1         5164         7177986       71.9
 9 FL    Wave 1        15320        21569932       71.0
10 AL    Wave 1         3560         5024803       70.8
# ℹ 250 more rows
```

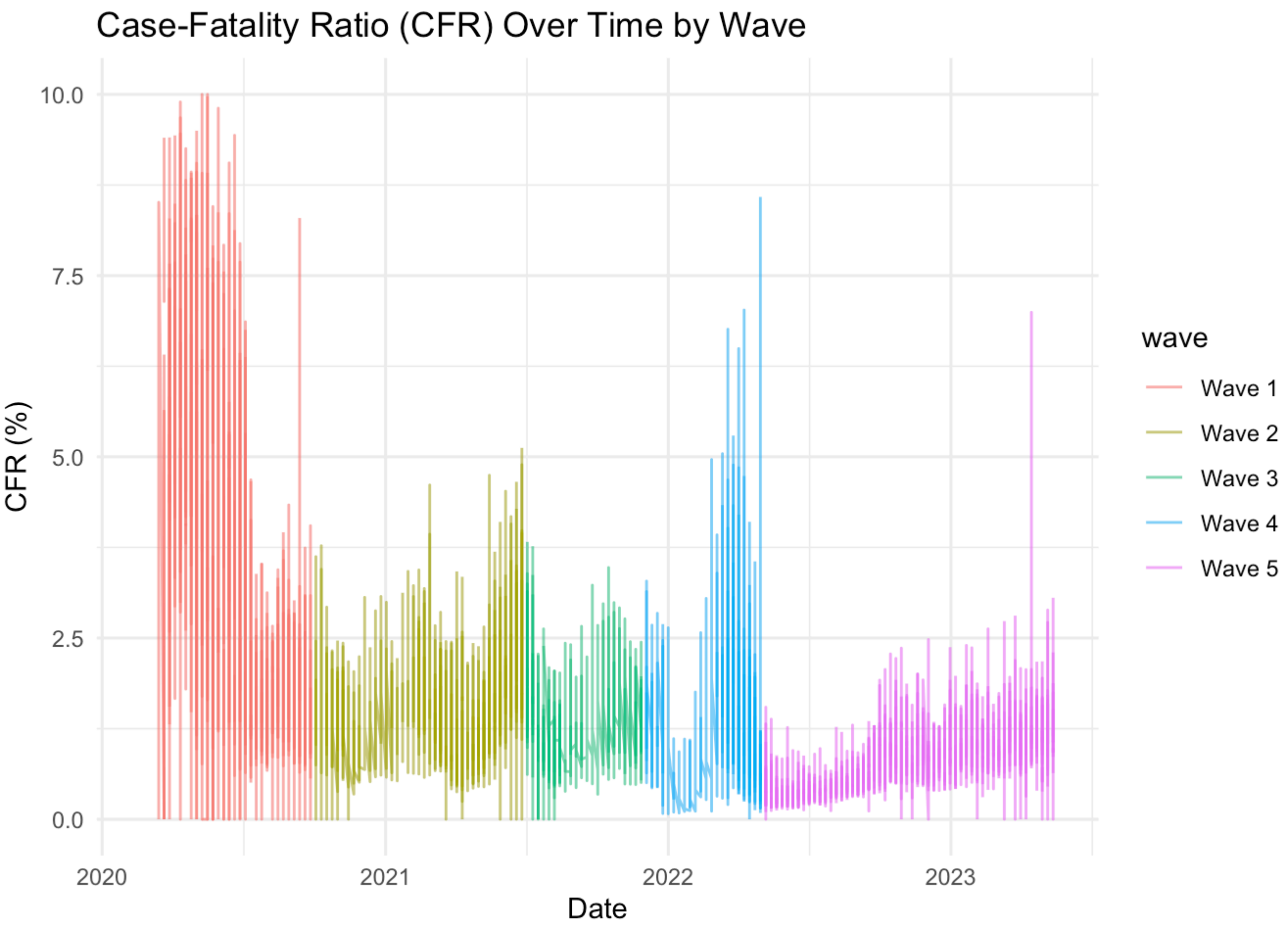
```
library(RColorBrewer)

# Visualise death rates across states and waves
ggplot(death_rates_by_wave, aes(x = wave, y = state, fill = death_rate)) +
  geom_tile(color = "white") +
  scale_fill_gradientn(    colors = brewer.pal(n = 9, name = "YlGnBu"), # Usir
name = "Death Rate\n(per 100,000)") +
  labs(title = "Death Rates by State and Wave",
       x = "Wave",
       y = "State") +
  theme_minimal()+
  theme(
    axis.text.y = element_text(size = 6), # Adjust the font size for y-axis
    axis.text.x = element_text(size = 10), # Adjust font size for x-axis
    axis.title = element_text(size = 12) # Adjust font size for titles
  )
```



Question 3 - Describe if COVID-19 became less or more virulent across the different periods.

```
# 1. Case-Fatality Ratio (CFR) Over Time by Wave
# Step 1: Create a new dataset for Q3.1
dat_wave_q31 <- dat_wave |>
  # Filter out rows where cases <= 10 to avoid division by very small numbers
  filter(cases > 10)  |>
  filter(wave != "Unknown")|>
  # Correct CFR calculation to avoid dividing by zero
  mutate(CFR = ifelse(cases > 0, (deaths / cases) * 100, NA))

# Step 2: Smooth the plot or limit the y-axis range to remove artifacts
p_cfr <- ggplot(dat_wave_q31, aes(x = date, y = CFR, color = wave, group = wav
  geom_line(alpha = 0.6) + # Add transparency for better clarity
  scale_y_continuous(limits = c(0, 10)) + # Limit y-axis to a realistic range
  labs(title = "Case-Fatality Ratio (CFR) Over Time by Wave",
       y = "CFR (%)",
       x = "Date") +
  theme_minimal() +
  theme(legend.position = "right")

# Print the plot
print(p_cfr)
```

```
Warning: Removed 11 rows containing missing values or values outside the scale range
(`geom_line()`).
```



```
# 2. Hospitalization-to-Death Ratio Over Time
# Step 1: Create a new dataset with HDR
dat_wave_hdr <- dat_wave |>
  filter(wave != "Unknown")|>
  mutate(HDR = ifelse(deaths > 0, hosp / deaths, NA)) |> # Calculate HDR
  group_by(date, wave) |>
  summarise(
    avg_HDR = mean(HDR, na.rm = TRUE),
    min_HDR = min(HDR, na.rm = TRUE),
    max_HDR = max(HDR, na.rm = TRUE)
  )
```

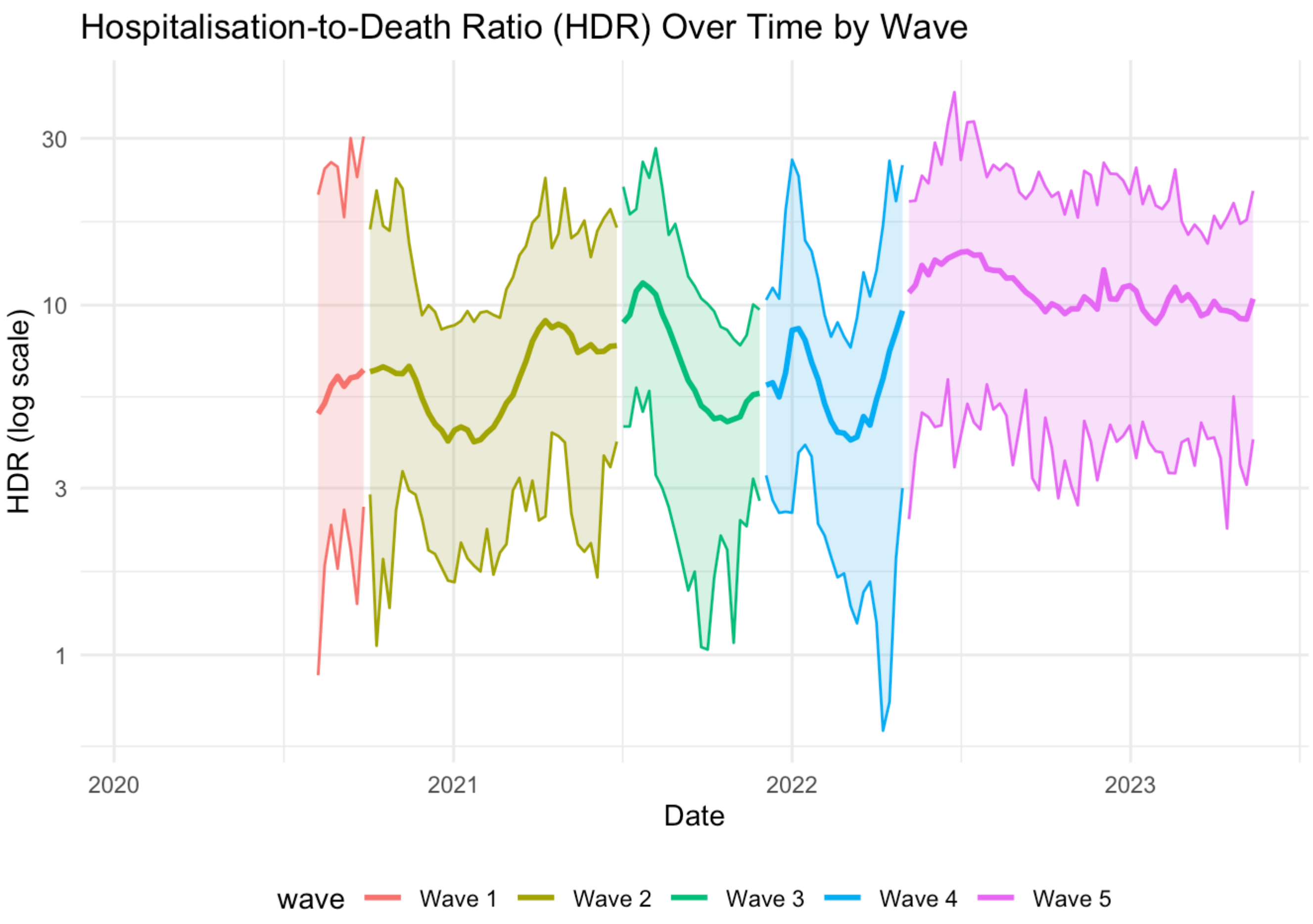
```
Warning: There were 56 warnings in `summarise()`.
The first warning was:
ℹ In argument: `min_HDR = min(HDR, na.rm = TRUE)`.
ℹ In group 1: `date = 2020-01-25` and `wave = "Wave 1"`.
Caused by warning in `min()`:
! no non-missing arguments to min; returning Inf
ℹ Run `dplyr::last_dplyr_warnings()` to see the 55 remaining warnings.

`summarise()` has grouped output by 'date'. You can override using the
`.groups` argument.
```

```
# Step 2: Generate the HDR plot over time
p_hdr <- ggplot(dat_wave_hdr, aes(x = date, y = avg_HDR, color = wave, group =
  geom_line(size = 1) + # Add lines for each wave
  geom_ribbon(aes(ymin = min_HDR, ymax = max_HDR, fill = wave), alpha = 0.2, s
  scale_y_continuous(trans = "log10", labels = scales::comma) + # Log scale fo
  labs(
    title = "Hospitalisation-to-Death Ratio (HDR) Over Time by Wave",
    x = "Date",
    y = "HDR (log scale)"
  ) +
  theme_minimal() +
  theme(
    legend.position = "bottom",
  )
```

```
Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
ℹ Please use `linewidth` instead.
```

```
# Print the plot
print(p_hdr)
```

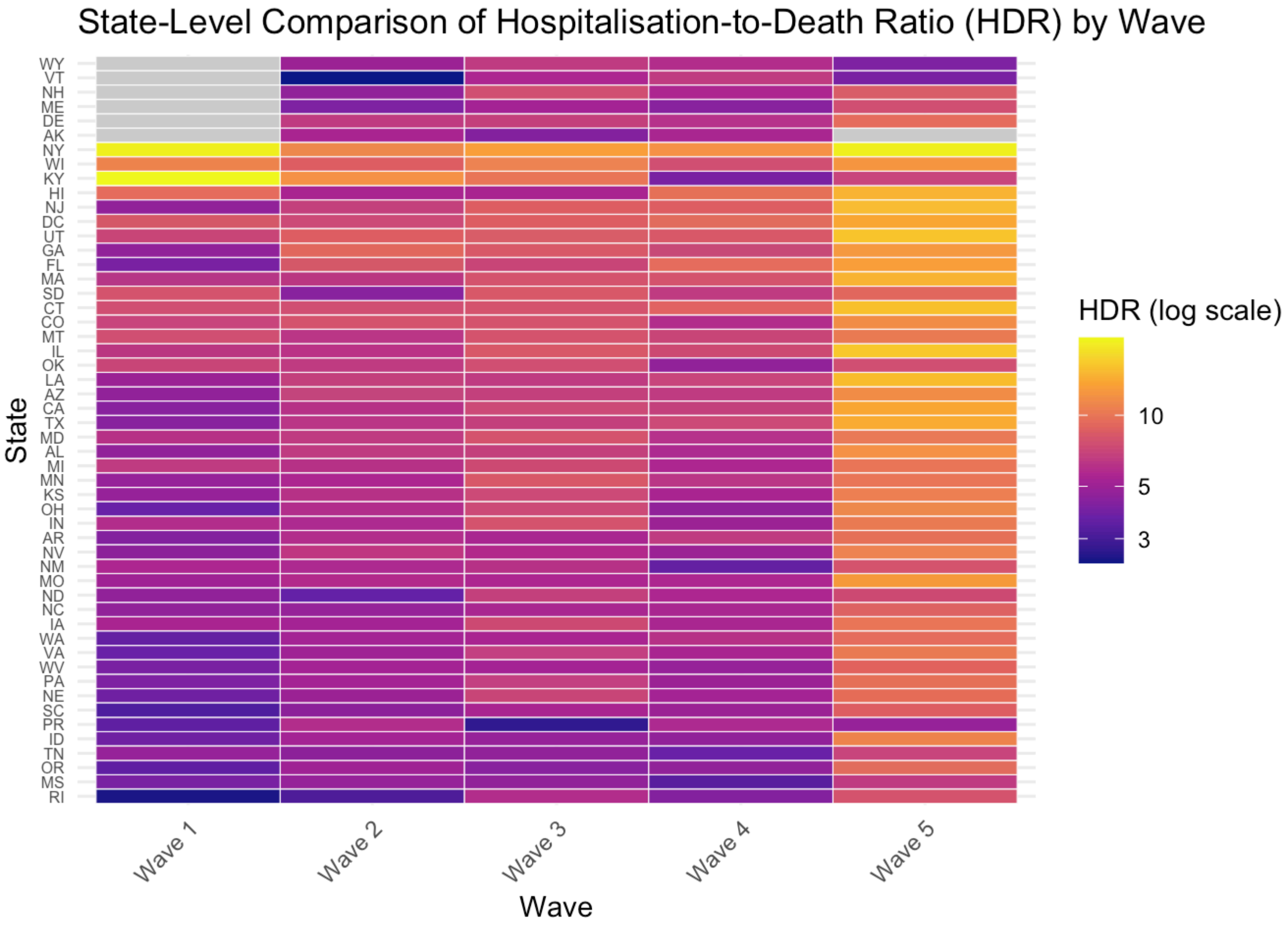
```
Warning in transformation$transform(x): NaNs produced

Warning: Removed 28 rows containing missing values or values outside the scale range
(`geom_line()`).
```



```
# 3. State-Level Comparison Heatmap
# Step 1: Create a dataset for the heatmap
state_wave_hdr <- dat_wave |>
  filter(wave != "Unknown") |>
  mutate(HDR = ifelse(deaths > 0, hosp / deaths, NA)) |> # Calculate HDR
  group_by(state, wave) |>
  summarise(avg_HDR = mean(HDR, na.rm = TRUE)) |> # Average HDR per state per
  ungroup()
```

```
`summarise()` has grouped output by 'state'. You can override using the
`.groups` argument.
```

```
# Step 2: Plot the heatmap
p_heatmap_hdr <- ggplot(state_wave_hdr, aes(x = wave, y = reorder(state, avg_H
  geom_tile(color = "white", size = 0.2) + # Heatmap tiles with borders
  scale_fill_viridis_c(option = "C", name = "HDR (log scale)", trans = "log10"
  labs(
    title = "State-Level Comparison of Hospitalisation-to-Death Ratio (HDR) by
    x = "Wave",
    y = "State"
  ) +
  theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1),
    legend.position = "right",
    axis.text.y = element_text(size = 6) # Adjust the font size for y-axis
  )

print(p_heatmap_hdr)
```



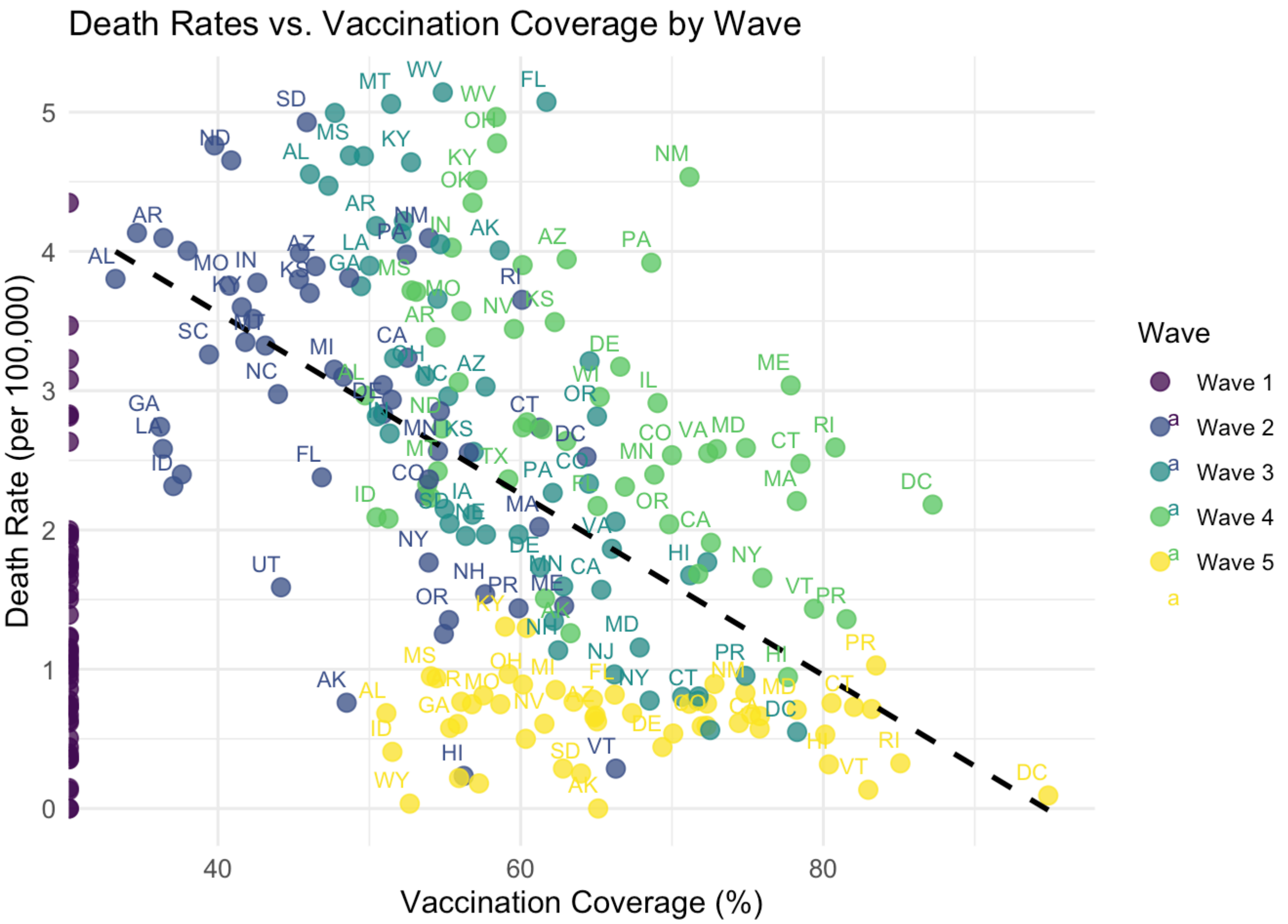
```
# 4. Death Rates vs. Vaccination Coverage
# Step 1: Create a dataset for the scatter plot
death_vax <- dat_wave |>
  group_by(state, wave) |>
  filter(wave != "Unknown")|>
  summarise(
    death_rate = sum(deaths, na.rm = TRUE) / sum(population, na.rm = TRUE) * 1
    vax_rate = max(series_complete, na.rm = TRUE) / max(population, na.rm = TR
  ) |>
  ungroup()
```

```
Warning: There were 52 warnings in `summarise()`.
The first warning was:
ℹ In argument: `vax_rate = *...`.
ℹ In group 1: `state = "AK"` and `wave = "Wave 1"`.
Caused by warning in `max()`:
! no non-missing arguments to max; returning -Inf
ℹ Run `dplyr::last_dplyr_warnings()` to see the 51 remaining warnings.

`summarise()` has grouped output by 'state'. You can override using the
`.groups` argument.
```

```
# Step 2: Plot the scatter plot
p_death_vax <- ggplot(death_vax, aes(x = vax_rate, y = death_rate, color = wav
  geom_point(size = 3, alpha = 0.8) + # Add points
  geom_smooth(method = "lm", se = FALSE, color = "black", linetype = "dashed")
```

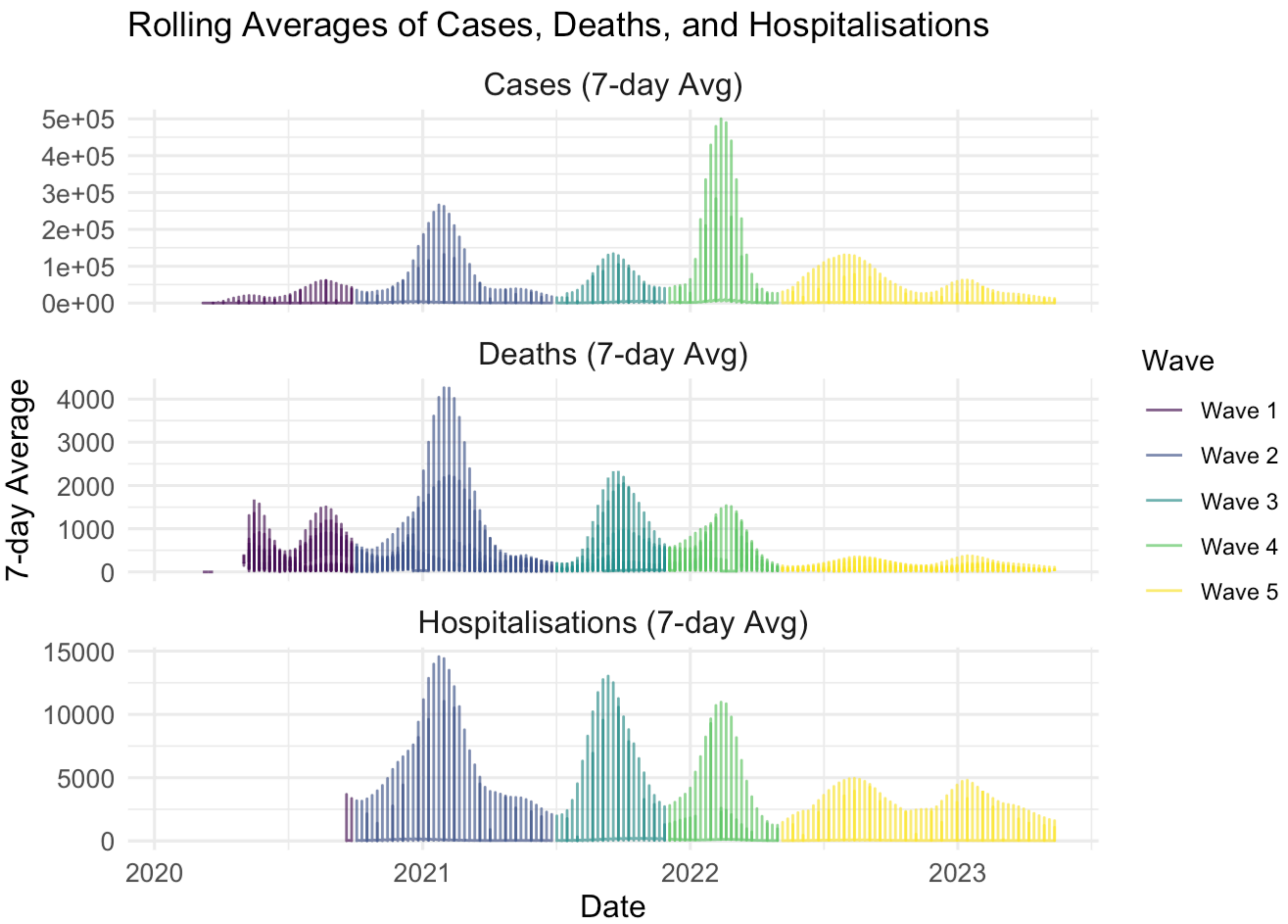
```
          scale_color_viridis_d(name = "Wave") + # Discrete color scale for waves
          labs(
            title = "Death Rates vs. Vaccination Coverage by Wave",
            x = "Vaccination Coverage (%)",
            y = "Death Rate (per 100,000)"
          ) +
          theme_minimal() +
          theme(
            legend.position = "right",
            axis.text = element_text(size = 10),
            axis.title = element_text(size = 12)
          )

        # Add labels to points (optional, can be removed for cleaner plot)
        p_death_vax <- p_death_vax +
          geom_text(size = 3, vjust = -1, hjust = 1, check_overlap = TRUE)

        # Print the plot
        print(p_death_vax)
```

```
`geom_smooth()` using formula = 'y ~ x'

Warning: Removed 52 rows containing non-finite outside the scale range
(`stat_smooth()`).

Warning: The following aesthetics were dropped during statistical transformation:
label.
ℹ This can happen when ggplot fails to infer the correct grouping structure in
  the data.
ℹ Did you forget to specify a `group` aesthetic or to convert a numerical
  variable into a factor?
```



```
# 5. Rolling Averages of Deaths, Cases, and Hospitalisations
# Step 1: Compute rolling averages
dat_wave_roll <- dat_wave |>
  group_by(state) |>
  filter(wave != "Unknown")|>
  mutate(
    cases_avg = zoo::rollmean(cases, k = 7, fill = NA, align = "right"), # 7-d
    deaths_avg = zoo::rollmean(deaths, k = 7, fill = NA, align = "right"), # 7
    hosp_avg = zoo::rollmean(hosp, k = 7, fill = NA, align = "right") # 7-day
  ) |>
  ungroup()

# Step 2: Prepare dataset for visualization
dat_roll_long <- dat_wave_roll |>
  select(date, wave, cases_avg, deaths_avg, hosp_avg) |>
  pivot_longer(
    cols = c(cases_avg, deaths_avg, hosp_avg),
    names_to = "metric",
    values_to = "rolling_avg"
  )

# Step 3: Plot rolling averages
p_roll <- ggplot(dat_roll_long, aes(x = date, y = rolling_avg, color = wave))
  geom_line(alpha = 0.7) +
  facet_wrap(~ metric, scales = "free_y", nrow = 3, labeller = as_labeller(c(
    cases_avg = "Cases (7-day Avg)",
    deaths_avg = "Deaths (7-day Avg)",
    hosp_avg = "Hospitalisations (7-day Avg)"
  ))) +
  scale_color_viridis_d(name = "Wave") +
  labs(
    title = "Rolling Averages of Cases, Deaths, and Hospitalisations",
    x = "Date",
    y = "7-day Average"
  ) +
  theme_minimal() +
  theme(
    legend.position = "right",
    axis.text = element_text(size = 10),
    axis.title = element_text(size = 12),
    strip.text = element_text(size = 12)
  )
print(p_roll)
```

```
Warning: Removed 312 rows containing missing values or values outside the scale range
(`geom_line()`).
```


Rolling Averages of Cases, Deaths, and Hospitalisations
Cases (7-day Avg)
5e+05
4e+05
3e+05
2e+05
1e+05
0e+00
Deaths (7-day Avg)
Hospitalisations (7-day Avg)
7-day Average
Date
Wave
Wave 1
Wave 2
Wave 3
Wave 4
Wave 5