

Stat 6021: HW 1

Tom Lever

08/26/22

Download the dataset `UScovid.csv` from Collab. The dataset was released by *The New York Times* and contains data on cumulative (i.e., accruing) counts of coronavirus cases and deaths in the United States, at the state and county level, over each day from Jan 21, 2020 to June 3, 2021. You may read more about the data and the variable descriptions [here](#). Please note the dataset is regularly updated. We will use the file on Collab.

Read the data file into R and store the dataset into the object `Covid`.

```
Covid <- read.csv("USCovid.csv")
head(Covid, n = 3)
```

```
##      date    county    state  fips cases deaths
## 1 2020-01-21 Snohomish Washington 53061      1      0
## 2 2020-01-22 Snohomish Washington 53061      1      0
## 3 2020-01-23 Snohomish Washington 53061      1      0
```

```
nrow(Covid)
```

```
## [1] 1384683
```

There are 1,384,683 snapshots in this dataset. The header row of `Covid` is not considered in this determination.

1. For this question, we focus on data at the county level.

- (a) We are interested in the data at the most recent date, June 3, 2021 (i.e., 2021-06-03). Create a data frame called `latest` that
- has only rows pertaining to data from June 3, 2021,
 - removes rows pertaining to counties that are “Unknown”,
 - removes the column `date` and `fips`, and
 - is ordered by `county` and then `state` alphabetically.

Use the `head()` function to display the first 6 rows of the data frame `latest`.

```
library(dplyr)
latest <-
  Covid%>%
    filter(date == "2021-06-03")%>%
    filter(county != "Unknown")%>%
    # filter(!is.na(county))
    # Unknown is not equivalent to NA (i.e., Not Available)
    select(-date, -fips)%>%
    arrange(county, state)
head(latest, n = 6)
```

```
##      county    state cases deaths
## 1 Abbeville South Carolina 2599      41
## 2 Acadia Louisiana 6703      195
```

```
## 3 Accomack      Virginia 2862    43
## 4      Ada      Idaho 52964   475
## 5      Adair     Iowa   873    32
## 6      Adair     Kentucky 1944   54
```

- (b) Calculate the death rate— call it `death.rate` —for each county. Report the death rate as a percent and round to two decimal places. Add `death.rate` as a new column to the data frame `latest`. Display the first 6 rows of the data frame `latest`.

```
death.rate <- round(latest%>%select(deaths) / latest%>%select(cases) * 100, 2)
colnames(death.rate) <- "death.rate"
latest <- bind_cols(latest, death.rate)

death.rate <-
  rename(
    round(latest%>%select(deaths) / latest%>%select(cases) * 100, 2),
    death.rate = deaths
  )
latest <- latest%>%mutate(death.rate = death.rate)
head(latest, n = 6)
```

```
##      county      state cases deaths death.rate
## 1 Abbeville South Carolina 2599    41      1.58
## 2  Acadia      Louisiana 6703   195      2.91
## 3 Accomack      Virginia 2862    43      1.50
## 4      Ada      Idaho 52964   475      0.90
## 5      Adair     Iowa   873    32      3.67
## 6      Adair     Kentucky 1944   54      2.78
```

- (c) Display the counties with the 10 largest numbers of cases. Be sure to display also the appropriate states, numbers of deaths, and death rates.

```
# "slice_min()" and slice_max() can order_by multiple variables
# if you supply them as a data.frame or tibble (#6176)."
# https://github.com/tidyverse/dplyr/blob/main/NEWS.md
# For slice_max, devtools::install_github("tidyverse/dplyr")
slice_max(
  latest,
  n = 10,
  order_by = data.frame(cases, county, state, deaths, death.rate),
  with_ties = FALSE
)
```

```
##      county      state cases deaths death.rate
## 1 Los Angeles California 1245127 24375      1.96
## 2 New York City New York 949986 33257      3.50
## 3 Cook      Illinois 554390 10893      1.96
## 4 Maricopa Arizona 551509 10084      1.83
## 5 Miami-Dade Florida 501925 6472      1.29
## 6 Harris Texas 401345 6462      1.61
## 7 Dallas Texas 303533 4082      1.34
## 8 Riverside California 300879 4614      1.53
## 9 San Bernardino California 298599 4760      1.59
## 10 San Diego California 280410 3760      1.34
```

- (d) Display the counties with the 10 largest numbers of deaths. Be sure to display also the appropriate states, numbers of cases, and death rates.

```
slice_max(
  latest,
  n = 10,
  order_by = data.frame(deaths, county, state, cases, death.rate),
  with_ties = FALSE
)
```

```
##           county      state  cases deaths death.rate
## 1   New York City   New York 949986  33257      3.50
## 2    Los Angeles California 1245127  24375      1.96
## 3         Cook    Illinois  554390  10893      1.96
## 4      Maricopa    Arizona  551509  10084      1.83
## 5   Miami-Dade    Florida  501925   6472      1.29
## 6        Harris     Texas  401345   6462      1.61
## 7       Orange California  272242   5070      1.86
## 8        Wayne    Michigan  164612   5048      3.07
## 9 San Bernardino California  298599   4760      1.59
## 10    Riverside California  300879   4614      1.53
```

- (e) Display the counties with the 10 highest death rates. Be sure to display also the appropriate states, numbers of cases, and numbers of deaths. Is there something you notice about these counties?

```
# For calculatePercentile, devtools::install_github("tslever/TomLevers_Git_Repository/TomLeversRPackage")
library(TomLeversRPackage)
counties_with_10_highest_death_rates <-
  slice_max(
    latest,
    n = 10,
    order_by = data.frame(death.rate, county, state, cases, deaths),
    with_ties = FALSE
  )
counties_with_10_highest_death_rates
```

```
##           county      state  cases deaths death.rate
## 1         Grant   Nebraska    41      4      9.76
## 2        Sabine     Texas   524     45      8.59
## 3   Petroleum    Montana    12      1      8.33
## 4    Harding New Mexico    12      1      8.33
## 5        Foard     Texas   124     10      8.06
## 6    Hancock    Georgia   928     68      7.33
## 7   Glascock    Georgia   269     19      7.06
## 8       Motley     Texas   116      8      6.90
## 9 Throckmorton     Texas    73      5      6.85
## 10    Candler    Georgia   978     67      6.85
```

```
calculatePercentile(
  latest%>%pull(cases),
  max(counties_with_10_highest_death_rates%>%select(cases), na.rm = TRUE)
)
```

```
## [1] 23
```

Yes. The percentile of the maximum number of cases among the counties with the 10 highest death rates, given all numbers of cases, is 23. The maximum number of cases among the counties with the 10 highest death rates is in the lowest quarter of numbers of cases.

- (f) Display the counties with the 10 highest death rates among counties with at least 100,000 cases.

Be sure to display also the appropriate states, numbers of cases, and numbers of deaths.

```
library(TomLeversRPackage)
latest%>%
  filter(cases > 100000)%>%
  slice_max(
    n = 10,
    order_by = data.frame(death.rate, county, state, cases, deaths),
    with_ties = FALSE
  )
```

##	county	state	cases	deaths	death.rate
## 1	New York City	New York	949986	33257	3.50
## 2	Wayne	Michigan	164612	5048	3.07
## 3	Middlesex	Massachusetts	134980	3761	2.79
## 4	Bergen	New Jersey	104301	2868	2.75
## 5	Macomb	Michigan	100190	2441	2.44
## 6	Philadelphia	Pennsylvania	153521	3692	2.40
## 7	St. Louis	Missouri	100195	2249	2.24
## 8	Fairfield	Connecticut	100093	2198	2.20
## 9	Pima	Arizona	116997	2406	2.06
## 10	Oakland	Michigan	118035	2368	2.01

(g) Display the number of cases, deaths, and death rate for the following counties.

i. Albemarle, Virginia

```
latest%>%
  filter(county == "Albemarle" & state == "Virginia")
```

##	county	state	cases	deaths	death.rate
## 1	Albemarle	Virginia	5801	83	1.43

ii. Charlottesville City, Virginia

```
latest%>%
  filter(county == "Charlottesville city" & state == "Virginia")
```

##	county	state	cases	deaths	death.rate
## 1	Charlottesville city	Virginia	4014	57	1.42

2. For this question, we focus on data at the state level. Note that the dataset has data on the 50 states, plus DC, Puerto Rico, Guam, Northern Mariana Islands, and the Virgin Islands.

(a) We are interested in the data at the most recent date, June 3, 2021. Create a data frame called `state.level` that

- has 55 rows, including 1 for each state, 1 for DC, and 1 for each territory
- has 3 columns, including `state`, `cases`, and `deaths`, and
- is ordered alphabetically by `state`.

Display the first 6 rows of the data frame `state.level`.

```
state.level <-
  Covid%>%
    filter(date == "2021-06-03")%>%
    group_by(state)%>%
    summarize(cases = sum(cases), deaths = sum(deaths, na.rm = TRUE))
head(state.level, n = 6)
```

```
## # A tibble: 6 x 3
```

```
##   state      cases deaths
##   <chr>      <int> <int>
## 1 Alabama    545028  11188
## 2 Alaska      69826   352
## 3 Arizona     882691  17653
## 4 Arkansas    341889   5842
## 5 California 3793055  63345
## 6 Colorado    547961   6746
```

```
nrow(state.level)
```

```
## [1] 55
```

- (b) Calculate the death rate (call it `state.rate`). Report the death rate as a percent and round to two decimal places. Add `state.rate` as a new column to the data frame `state.level`. Display the first 6 rows of the data frame `state.level`.

```
state.rate <-
  round(state.level%>%select(deaths) / state.level%>%select(cases) * 100, 2)
colnames(state.rate) <- "state.rate"
state.level <- bind_cols(state.level, state.rate)
```

```
state.rate <-
  rename(
    round(
      state.level%>%select(deaths) / state.level%>%select(cases) * 100,
      2
    ),
    state.rate = deaths
  )
state.level <- state.level%>%mutate(state.rate = state.rate)
head(state.level, n = 6)
```

```
## # A tibble: 6 x 4
##   state      cases deaths state.rate
##   <chr>      <int> <int>      <dbl>
## 1 Alabama    545028  11188      2.05
## 2 Alaska      69826   352        0.5
## 3 Arizona     882691  17653        2
## 4 Arkansas    341889   5842      1.71
## 5 California 3793055  63345      1.67
## 6 Colorado    547961   6746      1.23
```

- (c) What is the death rate in Virginia?

```
state.level%>%
  filter(state == "Virginia")%>%
  select(state, state.rate)
```

```
## # A tibble: 1 x 2
##   state      state.rate
##   <chr>      <dbl>
## 1 Virginia    1.66
```

The death rate in Virginia is 1.66 percent.

- (d) What is the death rate in Puerto Rico?

```
state.level%>%
  filter(state == "Puerto Rico")%>%
  select(state, state.rate)
```

```
## # A tibble: 1 x 2
##   state      state.rate
##   <chr>         <dbl>
## 1 Puerto Rico      1.46
```

The death rate in Puerto Rico is 1.46 percent.

- (e) Which states have the 10 highest death rates?

```
slice_max(
  state.level,
  n = 10,
  order_by = data.frame(state.rate, state, cases, deaths),
  with_ties = FALSE
)
```

```
## # A tibble: 10 x 4
##   state      cases deaths state.rate
##   <chr>      <int> <int>      <dbl>
## 1 New Jersey 1017044 26253      2.58
## 2 Massachusetts 707523 17893      2.53
## 3 New York   2102003 52811      2.51
## 4 Connecticut 347748 8245       2.37
## 5 District of Columbia 49041 1136       2.32
## 6 Mississippi 318048 7324       2.3
## 7 Pennsylvania 1208879 27349      2.26
## 8 Louisiana 472617 10605      2.24
## 9 New Mexico 203330 4275       2.1
## 10 Maryland 460406 9626       2.09
```

The states with the 10 highest death rates are listed in the above column `state`.

- (f) Which states have the 10 lowest death rates?

```
slice_min(
  state.level,
  n = 10,
  order_by = data.frame(state.rate, state, cases, deaths),
  with_ties = FALSE
)
```

```
## # A tibble: 10 x 4
##   state      cases deaths state.rate
##   <chr>      <int> <int>      <dbl>
## 1 Alaska    69826    352       0.5
## 2 Utah     406895   2308      0.57
## 3 Virgin Islands 3512    28       0.8
## 4 Vermont   24240    255      1.05
## 5 Nebraska  223517   2385      1.07
## 6 Idaho     192704   2103      1.09
## 7 Northern Mariana Islands 183    2       1.09
## 8 Wisconsin 675152   7923      1.17
## 9 Wyoming   60543    720      1.19
```

```
## 10 Colorado          547961    6746      1.23
```

The states with the 10 lowest death rates are listed in the above column `state`.

- (g) Export this dataset as a `.csv` file named `stateCovid.csv`. We will be using this file for the next homework.

I assume “this dataset” is `state.level`.

```
write.csv(state.level, "stateCovid.csv", row.names = FALSE)
```