STAT 471: Midterm Exam

Name

October 25, 2021, 7:00-9:00pm

Contents

Instructions

The materials you need for this exam are available [here](). Please navigate to this site and download the files you find there. Place `midterm-exam.Rmd` under `stat-471-fall-2021/midterm/midterm-fall-2021/` and `county-health-data.tsv` under `stat-471-fall-2021/data/`.

Use this document as a starting point for your writeup, adding your solutions after "**Solution**". Add your R code using code chunks and add your text answers using **bold text**. Compile your writeup to PDF and submit to [Gradescope]().

You must complete this exam individually, but you may consult any course materials or the internet.

We'll need to use the following R packages and functions:

```r
library(kableExtra)                      # for printing tables
library(cowplot)                         # for side by side plots
library(glmnetUtils)                     # to run ridge and lasso
library(lubridate)                       # for dealing with dates
library(maps)                            # for creating maps
source("../functions/plot_glmnet.R")     # for lasso/ridge trace plots
library(tidyverse)                       # for everything else
```

Socioeconomics and the COVID-19 case-fatality rate

The coronavirus pandemic emerged in 2020 and is still impacting our lives today. COVID-19 has had a disparate impact on different counties across the United States. A key measure of this impact is the *case-fatality ratio*, defined as the ratio of the number of deaths to the number of cases. Three STAT 471 students from spring 2021 (Yoon Chang, Jess Mixon, and Maayan Waldman) set out to study how a variety of variety of health, clinical, socioeconomic, and physical factors affected the case-fatality ratio. In this exam, we will be retracing their steps. The analysis will focus on the data from 2020, before the availability of COVID vaccines.

The data come in two parts: Case and death tracking data from The New York Times (available online) and 41 county-level health and socioeconomic factors compiled by the County Health Rankings and Roadmaps, available to you as `county_health_data.tsv` (see the Appendix below for descriptions of all features). The county health data have been cleaned for you, and counties with missing data have been removed. Counties are identified in both datasets using a five-digit *FIPS code*.

1 Wrangling (25 points for correctness; 5 points for presentation)

1.1 Import (5 points)

- Import the NYT data directly from the URL below into a tibble called `case_data_raw`. Print this tibble (no need to make a fancy table out of it).
- Import the county health data from `../../data/county_health_data.tsv` into a tibble called `county_health_data`. Print this tibble (no need to make a fancy table out of it).

Solution.

Read in the data using `read_csv` and `read_tsv` and print.

```
url = "https://raw.githubusercontent.com/nytimes/covid-19-data/master/us-counties.csv"
case_data_raw = read_csv(url) # read NYT data
```

```
## Rows: 1849105 Columns: 6

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (3): county, state, fips
## dbl  (2): cases, deaths
## date (1): date

##
## 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.
```

```
case_data_raw # print
```

```
## # A tibble: 1,849,105 x 6
##    date       county      state       fips  cases deaths
##    <date>     <chr>       <chr>       <chr> <dbl>  <dbl>
##  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
##  4 2020-01-24 Cook        Illinois   17031     1     0
##  5 2020-01-24 Snohomish   Washington 53061     1     0
##  6 2020-01-25 Orange      California 06059     1     0
##  7 2020-01-25 Cook        Illinois   17031     1     0
##  8 2020-01-25 Snohomish   Washington 53061     1     0
##  9 2020-01-26 Maricopa    Arizona    04013     1     0
## 10 2020-01-26 Los Angeles California 06037     1     0
## # ... with 1,849,095 more rows
```

```r
county_health_data = read_tsv("data/county_health_data.tsv") # import county health data
```

```
## Rows: 935 Columns: 42
```

```
## -- Column specification -------------------------------------------------
## Delimiter: "\t"
## chr  (1): fips
## dbl (41): low_birthweight_percentage, food_environment, physical_exercise_op...
```

```
##
## 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.
```

```r
county_health_data # print
```

```
## # A tibble: 935 x 42
##    fips  low_birthweight_per~ food_environment physical_exercise_op~ teen_births
##    <chr>                <dbl>            <dbl>                 <dbl>       <dbl>
##  1 01003               0.0835              8                  0.737      0.0279
##  2 01005               0.115               5.6                0.532      0.0409
##  3 01009               0.0760              8.4                0.156      0.0335
##  4 01015               0.0916              6.9                0.477      0.0335
##  5 01017               0.122               6.4                0.619      0.0454
##  6 01025               0.131               5.4                0.253      0.0392
##  7 01031               0.0831              7.5                0.537      0.0260
##  8 01033               0.102               7.4                0.557      0.0330
##  9 01039               0.103               7.6                0.502      0.0469
## 10 01043               0.0833              8.2                0.417      0.0417
## # ... with 925 more rows, and 37 more variables: limited_healthy_access <dbl>,
## #   stis <dbl>, uninsured <dbl>, primarycare_ratio <dbl>, dentist_ratio <dbl>,
## #   mentalhealth_ratio <dbl>, otherproviders_ratio <dbl>, HS_completion <dbl>,
## #   some_college <dbl>, disconnected_youth <dbl>, unemployment <dbl>,
## #   income_inequality <dbl>, children_freelunches <dbl>,
## #   single_parent_households <dbl>, social_associations <dbl>,
## #   water_violations <dbl>, high_housing_costs <dbl>, ...
```

1.2 Transform (15 points)

The NYT data contain case and death information for both 2020 and 2021, whereas we would like to focus our analysis only on 2020. Also, the data are broken down by day, whereas we would like to calculate an overall case-fatality ratio per county, defined as the total deaths in 2020, divided by the total cases in 2020, multiplied by 100 to obtain a percentage.

4

- Transform `case_data_raw` into a tibble called `case_data` with one row per county and four columns: `fips`, `county`, `state`, and `case_fatality_rate`, the latter containing the overall case-fatality ratio for 2020. [Hints: (1) There are several ways to filter the observations from 2020, but some are slower than others. For a faster option, check out the `year()` function from the `lubridate` package. (2) To keep columns in a tibble after `summarise()`, include them in `group_by()`. Just remember to `ungroup()` after summarizing.]
- Print the resulting tibble (no need to make a fancy table out of it). How many counties are represented in `case_data`? How does it compare to the number of counties in `county_health_data`? What is a likely explanation for this discrepancy?

Solution

```
case_data = case_data_raw %>%
  na.omit() %>%
  mutate(year = year(date)) %>%
  filter(year == 2020) %>%
  group_by(fips, county, state) %>%
  summarise(total_cases = sum(cases),
            total_deaths = sum(deaths)) %>%
  ungroup() %>%
  mutate(case_fatality_rate = total_deaths / total_cases * 100) %>%
  select(fips, county, state, case_fatality_rate)
```

```
## `summarise()` has grouped output by 'fips', 'county'. You can override using the `.groups` argument.
```

```
case_data
```

```
## # A tibble: 3,140 x 4
##    fips  county   state    case_fatality_rate
##    <chr> <chr>    <chr>                 <dbl>
##  1 01001 Autauga  Alabama                1.53
##  2 01003 Baldwin  Alabama                1.10
##  3 01005 Barbour  Alabama                1.16
##  4 01007 Bibb     Alabama                1.81
##  5 01009 Blount   Alabama                1.10
##  6 01011 Bullock  Alabama                2.64
##  7 01013 Butler   Alabama                4.01
##  8 01015 Calhoun  Alabama                1.47
##  9 01017 Chambers Alabama                3.69
## 10 01019 Cherokee Alabama                2.33
## # ... with 3,130 more rows
```

```
nrow(case_data)
```

```
## [1] 3140
```

```
nrow(county_health_data)
```

```
## [1] 935
```

There are **3140 counties represented in `case_data`. This is way more counties than the 935 counties in `county_health_data`. This is likely because the data in `county_health_data` is harder to collect and may come from a sample of the county population, so only large counties or counties with good data collection are able to have adequate health data.**

1.3 Merge (5 points)

- Merge `county_health_data` with `case_data` into one tibble called `covid_data` using `inner_join()`, which keeps counties represented in both datasets. See `?inner_join` or Google for documentation and examples. Print `covid_data` (no need to create a nice table).

Solution

```
covid_data = case_data %>% inner_join(county_health_data, by = "fips") # inner_join by "fips"
covid_data # print
```

```
## # A tibble: 935 x 45
##    fips  county    state    case_fatality_rate low_birthweight_~ food_environment
##    <chr> <chr>     <chr>                 <dbl>             <dbl>            <dbl>
##  1 01003 Baldwin   Alabama                1.10            0.0835            8
##  2 01005 Barbour   Alabama                1.16            0.115             5.6
##  3 01009 Blount    Alabama                1.10            0.0760            8.4
##  4 01015 Calhoun   Alabama                1.47            0.0916            6.9
##  5 01017 Chambers  Alabama                3.69            0.122             6.4
##  6 01025 Clarke    Alabama                1.33            0.131             5.4
##  7 01031 Coffee    Alabama                0.692           0.0831            7.5
##  8 01033 Colbert   Alabama                1.41            0.102             7.4
##  9 01039 Covington Alabama                1.80            0.103             7.6
## 10 01043 Cullman   Alabama                0.963           0.0833            8.2
## # ... with 925 more rows, and 39 more variables:
## #   physical_exercise_opportunities <dbl>, teen_births <dbl>,
## #   limited_healthy_access <dbl>, stis <dbl>, uninsured <dbl>,
## #   primarycare_ratio <dbl>, dentist_ratio <dbl>, mentalhealth_ratio <dbl>,
## #   otherproviders_ratio <dbl>, HS_completion <dbl>, some_college <dbl>,
## #   disconnected_youth <dbl>, unemployment <dbl>, income_inequality <dbl>,
## #   children_freelunches <dbl>, single_parent_households <dbl>, ...
```

Use `inner_join` to join datasets by "fips".

2 Exploration (30 points for correctness; 5 points for presentation)

2.1 Response distribution (15 points)

- Compute the median of the case-fatality rate in `covid_data`.
- Create a histogram of the case-fatality rate in `covid_data`, with a dashed vertical line at the median. Comment on the shape of this distribution.
- Create a (nice) table of the top 10 counties by case-fatality rate, as well as a heatmap of the case-fatality rate across the U.S. (the code to produce the heatmap is provided in the Rmd file; no need to modify it at all). Based on the table, what region of the U.S. tended to have the highest overall case-fatality rates in 2020? In what sense does the heatmap reflect this?
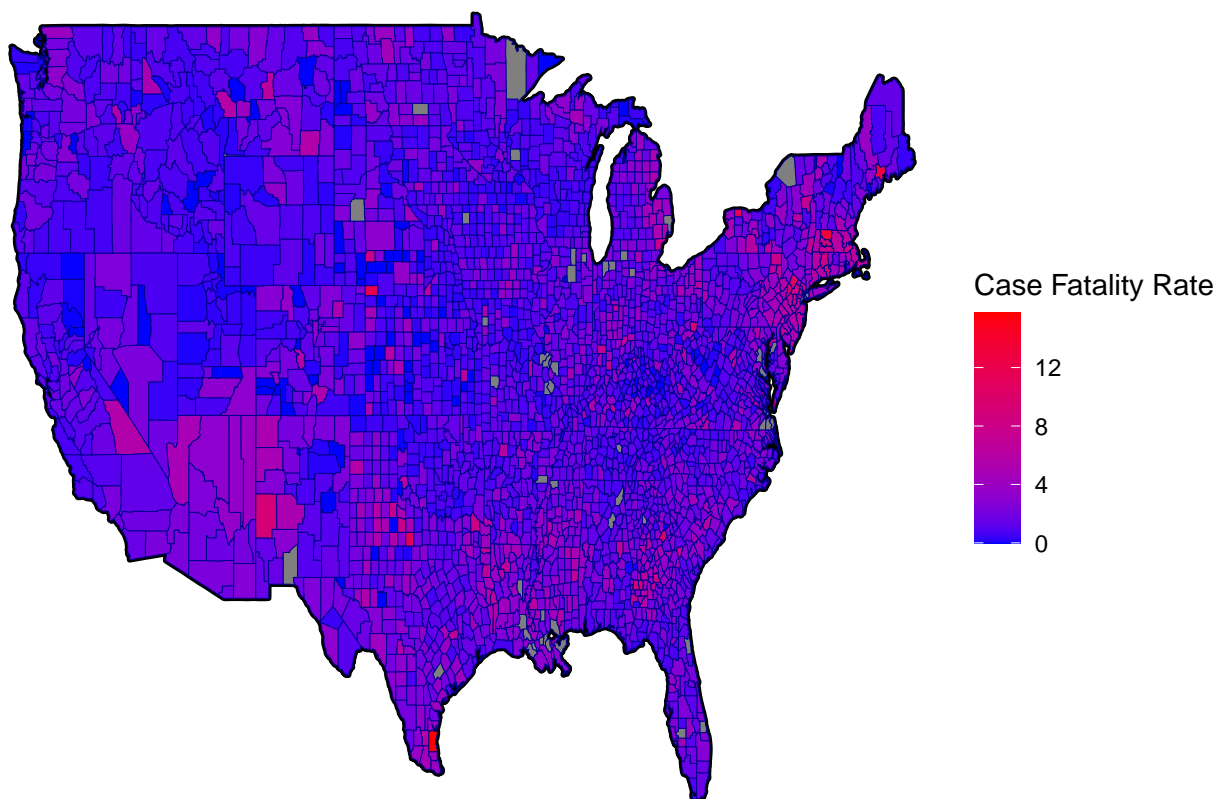
Solution

```r
# find median case fatality rate
median_case_fatality_rate = covid_data %>% pull(case_fatality_rate) %>% median()
median_case_fatality_rate # print
```

```
## [1] 1.83
```

```r
# plot histogram of case fatality rate
covid_data %>%
  ggplot(aes(case_fatality_rate)) +
  geom_histogram() +
  geom_vline(xintercept = median_case_fatality_rate,
             linetype="dashed", color = "red") +
  theme_bw()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```
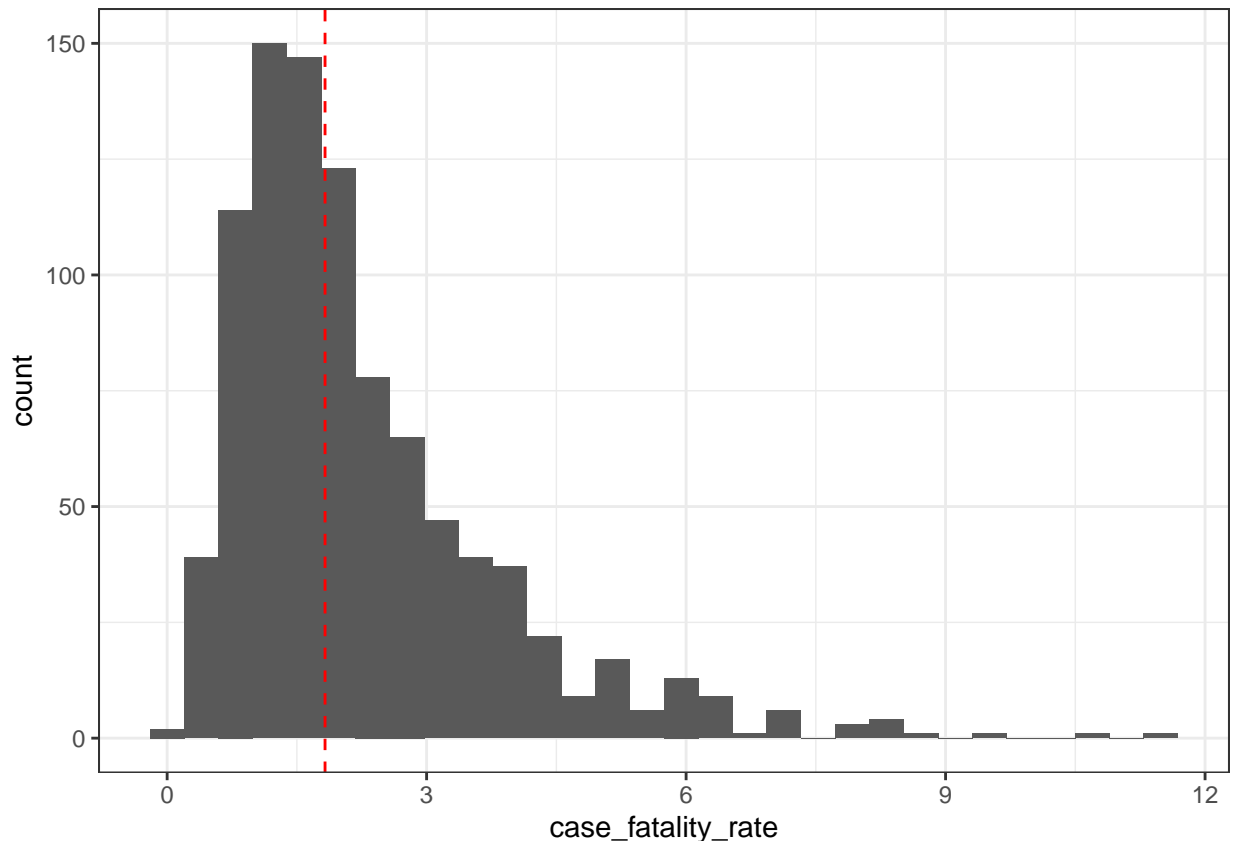
```
# make table of top 10 counties by case fatality rate
covid_data %>%
  arrange(desc(case_fatality_rate)) %>%
  select(fips, county, state, case_fatality_rate) %>%
  slice_head(n = 10) %>%
  kable(format = "latex", row.names = NA,
        booktabs = TRUE, digits = 2,
        col.names = c("FIPS", "Name", "State", "Case Fatality Rate"),
        caption = "Top Ten US Counties by Case Fatality Rate") %>%
  kable_styling(position = "center")
```

Based on the table, the Northeast tends to have the highest overall case fatality rates in 2020. Nine of ten counties in the table come from a state in this region. The heatmap reflects this because there is a concentration of red around this region, especially in northern New Jersey, New York, Connecticut, and Massachusetts.

2.2 Response-feature relationships (15 points)

- To prevent selection bias, it's good practice to split off a test set before exploring response-feature relationships. Create a test set `covid_test` by filtering counties belonging to the first six states (in alphabetical order) that are represented in `covid_data`; these should be Alabama, Arizona, Arkansas, California, Colorado, and Connecticut. Create a training set `covid_train` containing the rest of the counties.

- The features come in four different categories: health behaviors, clinical care, social and economic factors, and physical environment. Create scatter plots of the case fatality ratio against one

Table 1: Top Ten US Counties by Case Fatality Rate

FIPS	Name	State	Case Fatality Rate
36073	Orleans	New York	11.60
34037	Sussex	New Jersey	10.70
34041	Warren	New Jersey	9.61
34027	Morris	New Jersey	8.54
34035	Somerset	New Jersey	8.49
34013	Essex	New Jersey	8.44
09003	Hartford	Connecticut	8.30
34009	Cape May	New Jersey	8.27
34003	Bergen	New Jersey	7.84
26163	Wayne	Michigan	7.76

feature in each of these categories (`obesity_perc`, `uninsured`, `segregation_nonwhite_white`, `high_housing_costs`), adding the least squares line to each and putting the y-axis on a log scale using `scale_y_log10()` for visualization purposes and collating these plots into a single figure.

- Which of these four features appears to have the strongest relationship with the case-fatality ratio? What appears to be the direction of the relationship, and why might this relationship exist?

Solution.

```
# list of first six states by alphabetical order
first_six_states = c("Alabama", "Arizona", "Arkansas", "California", "Colorado", "Connecticut")
# test set called covid_test
covid_test = covid_data %>%
  filter(state %in% first_six_states)
# training set called covid_train
covid_train = covid_data %>%
  filter(!state %in% first_six_states)

# scatter plot of case `case_fatality_ratio` against `obesity_perc`
plot_op = covid_data %>%
  ggplot(aes(x = obesity_perc, y = case_fatality_rate)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
  scale_y_log10() +
  theme_bw()

# scatter plot of case `case_fatality_ratio` against `uninsured`,
plot_u = covid_data %>%
  ggplot(aes(x = uninsured, y = case_fatality_rate)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
  scale_y_log10() +
  theme_bw()

# scatter plot of case `case_fatality_ratio` against `segregation_nonwhite_white`
plot_snw = covid_data %>%
  ggplot(aes(x = segregation_nonwhite_white, y = case_fatality_rate)) +
  geom_point() +
```

```
  geom_smooth(method = "lm", se = FALSE) +
  scale_y_log10() +
  theme_bw()

# scatter plot of case `case_fatality_ratio` against `high_housing_costs`
plot_rhc = covid_data %>%
  ggplot(aes(x = high_housing_costs, y = case_fatality_rate)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
  scale_y_log10() +
  theme_bw()

# plot all in same figure with cowplot
plot_grid(plot_op, plot_u, plot_snw, plot_rhc)
```
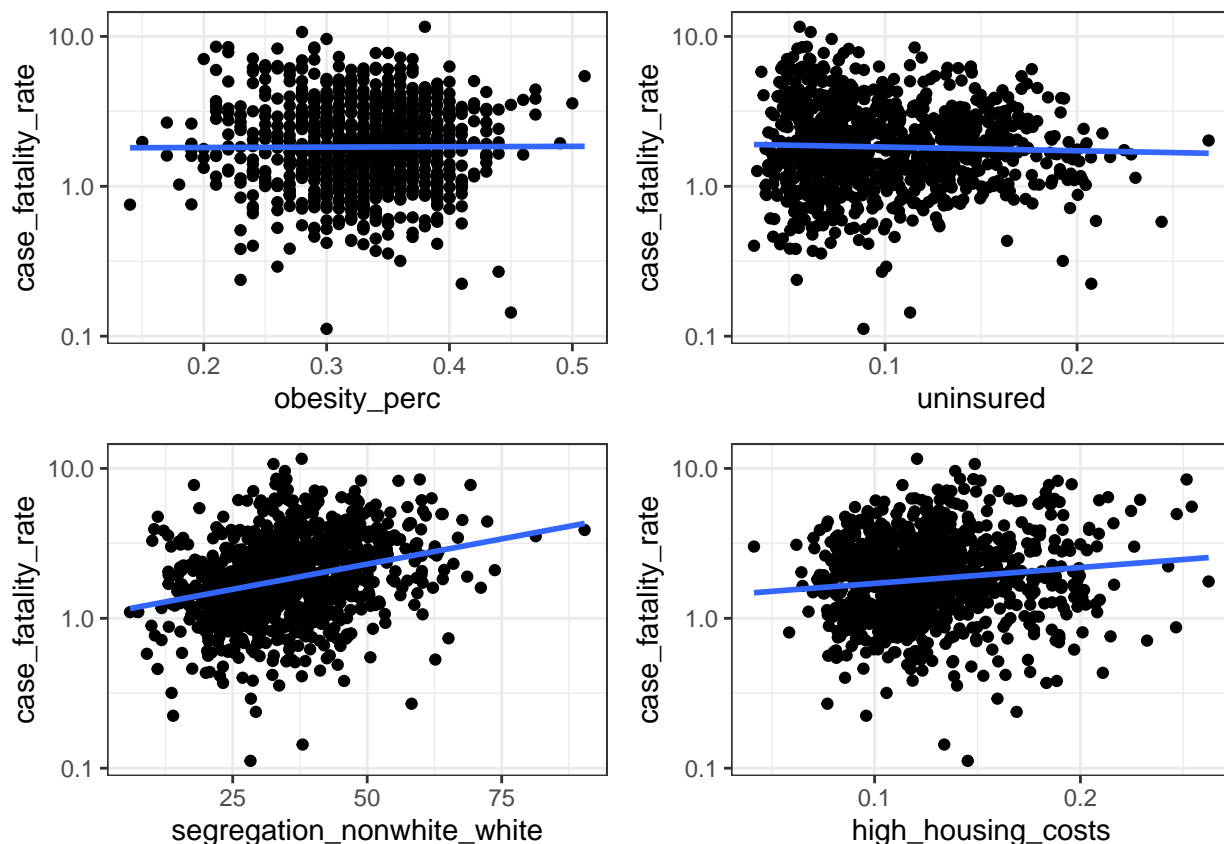
```
## 'geom_smooth()' using formula 'y ~ x'
## 'geom_smooth()' using formula 'y ~ x'
## 'geom_smooth()' using formula 'y ~ x'
## 'geom_smooth()' using formula 'y ~ x'
```


Of the four features, `segregation_nonwhite_white` seems to have by far the strongest relationship with `case_fatality_rate`. The relationship is positive, with an increase in `segregation_nonwhite_white` correlating with an increase in `case_fatality_rate`. This could be the case because `segregation_nonwhite_white`is an indicator of economic and racial inequality. Inequality leads to the less fortunate to lack health care access and other means of staying safe during covid.

3   Modeling (30 points for correctness; 5 points for presentation)

Next, let's train penalized regression models to predict the case-fatality ratio based on the available features.

3.1   Ridge regression (8 points)

- Fit a 10-fold cross-validated ridge regression to `covid_train`.
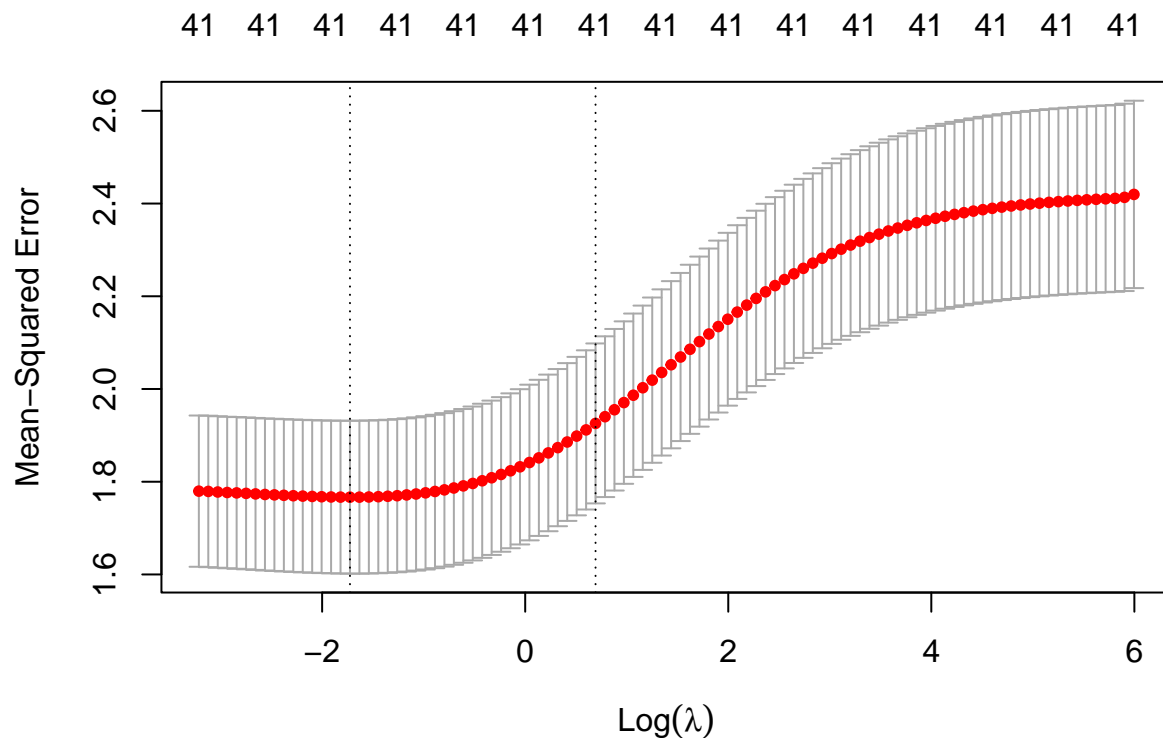
Solution.

```
set.seed(1)  # for replicability (do not change)
ridge_fit = cv.glmnet(case_fatality_rate ~ . - fips - county - state,
                       alpha = 0,                # alpha = 0 for ridge
                       nfolds = 10,              # 10 folds
                       data = covid_train)   # data
```

- Produce the corresponding CV plot. What are `lambda.min` and `lambda.1se`, and where are these two indicated in the CV plot?

Solution.

```
# CV plot
plot(ridge_fit)
```

"lamba.min" is 0.178 and is represented on the plot by the dashed line on the left. "lamda.1se" is 2 and is represented on the plot by the dashed line on the right.

- Produce the ridge trace plot, highlighting the top 6 features. Based on `lambda.1se`, which feature appears to have the strongest negative impact on the case-fatality ratio? Is the reason for this relationship apparent to you? Does this ridge regression result imply a statistically significant relationship between this feature and the case-fatality rate?

Solution.

```
plot_glmnet(ridge_fit, covid_train, features_to_plot = 6)
```
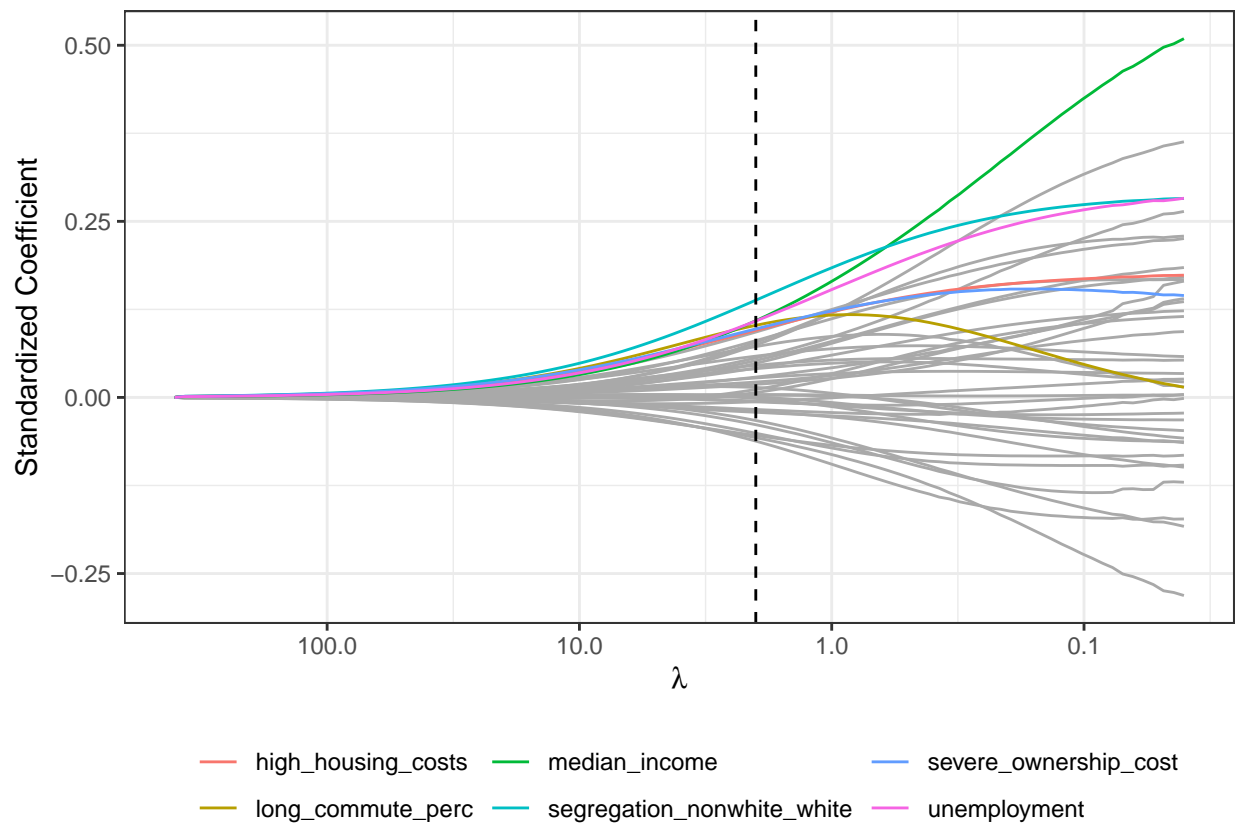

`segregation_nonwhite_white` **seems to have the the strongest positive impact on** `case_fatality_rate'`.
`It is unclear which feature has the strongest negative impact because it is not one of`
`the 6 most impactful features. The ridge regression result does NOT imply a statistically`
`significant relationship betweensegregation_nonwhite_whiteandcase_fatality_rate'.`

3.2 Lasso regression (14 points)

- Fit a 10-fold cross-validated lasso regression to `covid_train`.
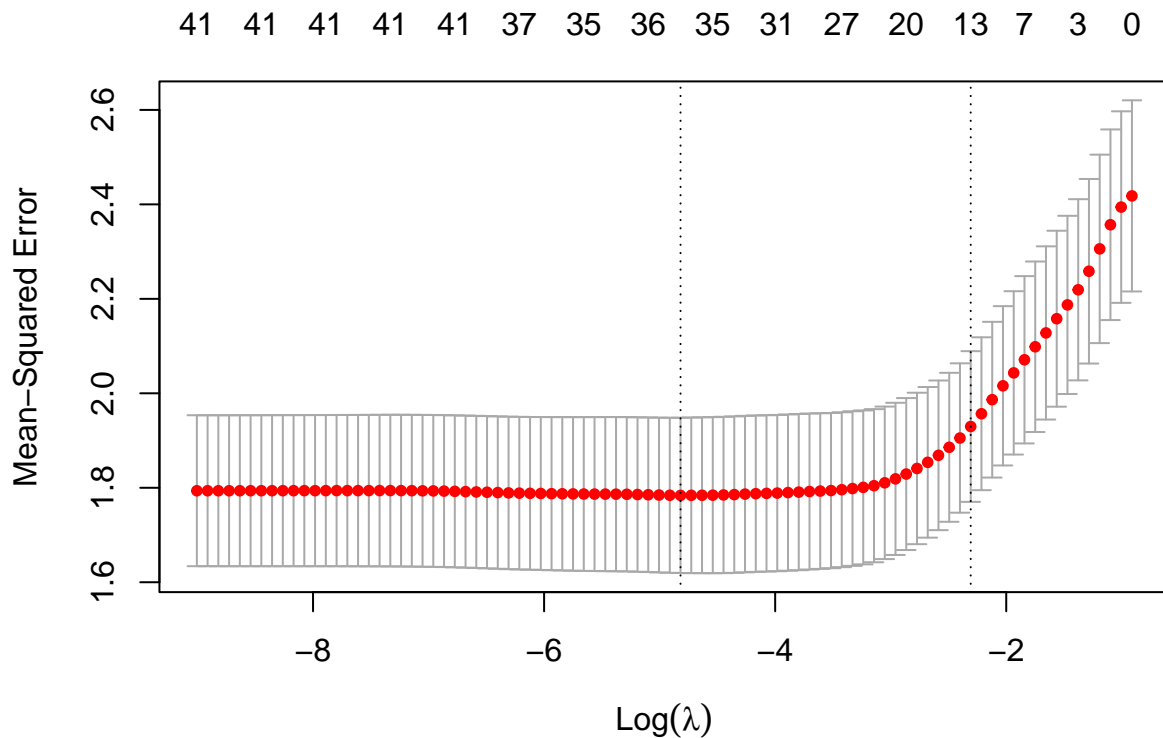
Solution.

```
set.seed(1)  # for replicability (do not change)
lasso_fit = cv.glmnet(case_fatality_rate ~ . - fips - county - state,
                      alpha = 1,               # alpha = 0 for ridge
                      nfolds = 10,             # 10 folds
                      data = covid_train)   # data
```

- Produce the corresponding CV plot. What is another name for the model represented in the left-most edge of the CV plot? Why does it perform poorly?

Solution.

```
plot(lasso_fit)
```


Another name for the model representing the left-most edge of the CV plot is the standard linear least squares model. It seems to perform just as well when measured by MSE but is not prefered because it has fair more features and therefore is not as simple and easy to interpret.

- How many features with nonzero coefficients are there in the lasso model selected by the one-standard error rule?

Solution.

```
beta_hat_std = extract_std_coefs(lasso_fit, covid_train)
nrow(beta_hat_std %>% filter(coefficient != 0))
```
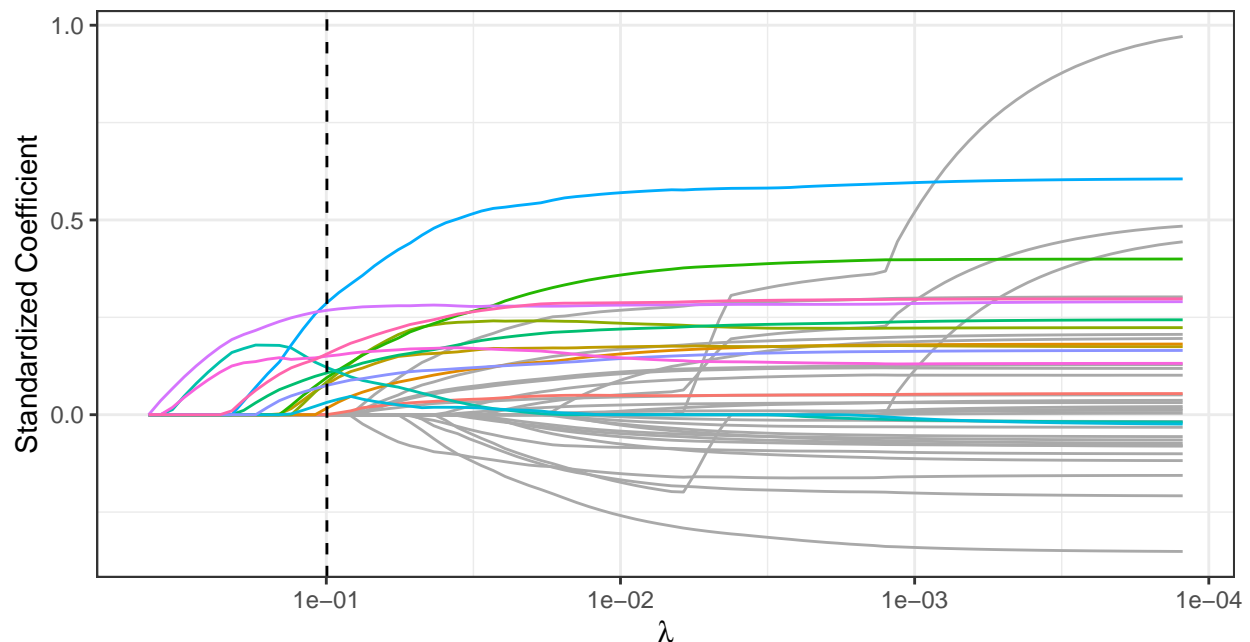
```
## [1] 13
```

There are 13 features with nonzero coefficients.

- Produce the lasso trace plot, highlighting the top 6 features. What is the first feature entering the model with a positive coefficient? What is the first feature entering the model with a negative coefficient?

Solution.

```
plot_glmnet(lasso_fit, covid_train)
```


_pollution	— homeownership	— long_commute_perc	— otherproviders_ratio	—
_vaccine_perc	— inactive_perc	— low_birthweight_percentage	— segregation_nonwhite_white	
h_housing_costs	— income_inequality	— median_income	— severe_ownership_cost	

The first feature entering the model with a positive coefficient is `segregation_nonwhite_white`. It is unclear which is the first feature entering the model with a negative coefficient

- Produce a nice table of all features with nonzero coefficients in the lasso model selected by the one-standard-error rule, ordered by their coefficient magnitudes. What is the coefficient of `flu_vaccine_perc`, and how do we interpret it? Comment on the sign of this coefficient.

Solution.

14

feature	coefficient
median_income	0.29
segregation_nonwhite_white	0.27
unemployment	0.16
severe_ownership_cost	0.15
long_commute_perc	0.12
income_inequality	0.11
inactive_perc	0.10
homeownership	0.08
high_housing_costs	0.08
otherproviders_ratio	0.08
low_birthweight_percentage	0.03
flu_vaccine_perc	0.02
air_pollution	0.00

```
beta_hat_std %>%
  filter(coefficient != 0) %>%
  arrange(desc(coefficient)) %>%
  kable(format = "latex", row.names = NA,
        booktabs = TRUE, digits = 2) %>%
  kable_styling(position = "center")
```

`flu_vaccine_perc` has a coefficient of 0.02. Which means that for every 1% change in flue

3.3 Performance evaluation (8 points)

- Evaluate the RMSE of the ridge and lasso methods, both with `lambda` chosen using the one-standard-error-rule. For the sake of comparison, also evaluate the RMSE of the intercept-only prediction rule, which predicts the mean case-fatality ratio in the training data for all counties. Print these three RMSE values in a nice table.

Solution.

- Which of the two penalized regression methods performs better, and how does its performance compare to the intercept-only model? Contextualize the latter comparison in terms of the bias-variance trade-off.

Solution.

4 Appendix: Descriptions of features

Below are the 41 features we used for analysis. Words written in parentheses represent variable names. Unless noted otherwise, all variables are continuous.

Health behaviors:

- *Tobacco Use*
  - Adult smoking (`smoke_perc`): Percentage of adults who are current smokers.

- *Diet and Exercise*
  - Adult obesity (`obesity_perc`): Percentage of the adult population (age 20 and older) reporting a body mass index (BMI) greater than or equal to 30 kg/m2.
  - Food environment index (`food_environment`): Index of factors that contribute to a healthy food environment, from 0 (worst) to 10 (best).
  - Physical inactivity (`inactive_perc`): Percentage of adults age 20 and over reporting no leisure-time physical activity.
  - Access to exercise opportunities (`physical_exercise_opportunities`): Percentage of population with adequate access to locations for physical activity
  - Food insecurity (`Food_Insecure_perc`): Percentage of population who lack adequate access to food.
  - Limited access to healthy foods (`limited_healthy_access`): Percentage of population who are low-income and do not live close to a grocery store.

- *Alcohol & Drug Use*
  - Excessive Drinking (`drinking_perc`): Percentage of adults reporting binge or heavy drinking.

- *Sexual Activity*
  - Sexually transmitted infections (`stis`): Number of newly diagnosed chlamydia cases per 100,000 population.
  - Teen births (`teen_births`): Number of births per 1,000 female population ages 15-19.
  - Low Birth Weight Percentage (`low_birthweight_percentage`): Percentage of live births with low birthweight ($< 2{,}500$ grams).

Clinical care:

- *Access to Care*
  - Uninsured (`uninsured`): Percentage of population under age 65 without health insurance.
  - Primary care physicians (`primarycare_ratio`): Ratio of population to primary care physicians.
  - Dentists (`dentist_ratio`): Ratio of population to dentists.
  - Mental health providers (mentalhealth_ratio): Ratio of population to mental health providers.
  - Other primary care providers (`otherproviders_ratio`): Ratio of population to primary care providers other than physicians.

- *Quality of Care*
  - Preventable hospital stays (`preventable_hospitalization`): Rate of hospital stays for ambulatory-care sensitive conditions per 100,000 Medicare enrollees.
  - Mammography screening (`mammogram_perc`): Percentage of female Medicare enrollees ages 65-74 that received an annual mammography screening.
  - Flu vaccinations (`flu_vaccine_perc`): Percentage of fee-for-service (FFS) Medicare enrollees that had an annual flu vaccination.
  - Teen births (`teen_births`): Number of births per 1,000 female population ages 15-19.

Social and economic factors:

- *Education*
  - High school completion (`HS_completion`): Percentage of adults ages 25 and over with a high school diploma or equivalent.
  - Some college (`some_college`): Percentage of adults ages 25-44 with some post-secondary education.
  - Disconnected youth (`disconnected_youth`): Percentage of teens and young adults ages 16-19 who are neither working nor in school.

- *Employment*

  - Unemployment (`unemployment`): Percentage of population ages 16 and older who are unemployed but seeking work.

- *Income*

  - Children in poverty (`children_poverty_percent`): Percentage of people under age 18 in poverty.
  - Income inequality (`income_inequality`): Ratio of household income at the 80th percentile to income at the 20th percentile.
  - Median household income (`median_income`): The income where half of households in a county earn more and half of households earn less.
  - Children eligible for free or reduced price lunch (`children_freelunches`): Percentage of children enrolled in public schools that are eligible for free or reduced price lunch.

- *Family & Social Support*

  - Children in single-parent households (`single_parent_households`): Percentage of children that live in a household headed by a single parent.
  - Social associations (`social_associations`): Number of membership associations per 10,000 residents.
  - Residential segregation—Black/White (`segregation_black_white`): Index of dissimilarity where higher values indicate greater residential segregation between Black and White county residents.
  - Residential segregation—non-White/White (`segregation_nonwhite_white`): Index of dissimilarity where higher values indicate greater residential segregation between non-White and White county residents.

- *Community Safety*

  - Violent crime rate (`Violent_crime`) Number of reported violent crime offenses per 100,000 residents.

Physical environment:

- *Air & Water Quality*

  - Air pollution - particulate matter (`air_pollution`): Average daily density of fine particulate matter in micrograms per cubic meter (PM2.5).
  - Drinking water violations (`water_violations`): Indicator of the presence of health-related drinking water violations. 1 indicates the presence of a violation, 0 indicates no violation.

- *Housing & Transit*

  - Housing overcrowding (`housing_overcrowding`): Percentage of households with overcrowding,
  - Severe housing costs (`high_housing_costs`): Percentage of households with high housing costs
  - Driving alone to work (`driving_alone_perc`): Percentage of the workforce that drives alone to work.
  - Long commute—driving alone (`long_commute_perc`): Among workers who commute in their car alone, the percentage that commute more than 30 minutes.
  - Traffic volume (`traffic_volume`): Average traffic volume per meter of major roadways in the county.
  - Homeownership (`homeownership`): Percentage of occupied housing units that are owned.
  - Severe housing cost burden (`severe_ownership_cost`): Percentage of households that spend 50% or more of their household income on housing.