STAT 471: Midterm Exam Solutions

Eugene Katsevich

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:

```
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 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 each variable in this dataset). 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

1.1 Import

- 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).

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

# read county health data from file
county_health_data = read_tsv("../../data/county_health_data.tsv")

# print the two tibbles
case_data_raw
```

```
## # A tibble: 1,852,355 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 1,852,345 more rows
```

```
county_health_data
```

```
## # 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

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 a single 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.

- 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`. [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?

```
# wrangle case data
case_data = case_data_raw %>%
  na.omit() %>%                                  # remove NA values
  filter(year(date) == 2020) %>%                 # keep data from 2020
  group_by(fips, county, state) %>%              # group by county
  summarise(total_cases = sum(cases),            # total cases per county
            total_deaths = sum(deaths)) %>%      # total deaths per county
  ungroup() %>%
  mutate(case_fatality_rate =                    # case_fatality_rate =
           total_deaths/total_cases*100) %>%     #  total_deaths/total_cases
  select(-total_cases, -total_deaths)            # remove intermediate variables

# print case data
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
```

There are 3140 counties represented in the NYT data. This is a much greater number than the number of counties represented in the county health data, which is only 935. This is likely because the county health data was not available for many counties.

1.3 Merge

- 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).

```
# join county health data with case data
covid_data = inner_join(county_health_data, case_data, by = "fips")
covid_data
```

```
## # A tibble: 935 x 45
##    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 40 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>, ...
```

2 Exploration

2.1 Response distribution

- Compute the median of the case-fatality rate in `covid_data`.

```
# calculate median case fatality rate
median_case_fatality_rate = covid_data %>%
  summarise(median(case_fatality_rate)) %>%
  pull()
```

The median of the case-fatality rate is 1.825%.

- 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.

```r
# plot histogram of case fatality rate
covid_data %>%
  ggplot(aes(x = case_fatality_rate)) +
  geom_histogram() +
  geom_vline(xintercept = median_case_fatality_rate,
             linetype = "dashed") +
  labs(x = "Case fatality rate (percent)",
       y = "Number of counties") +
  theme_bw()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

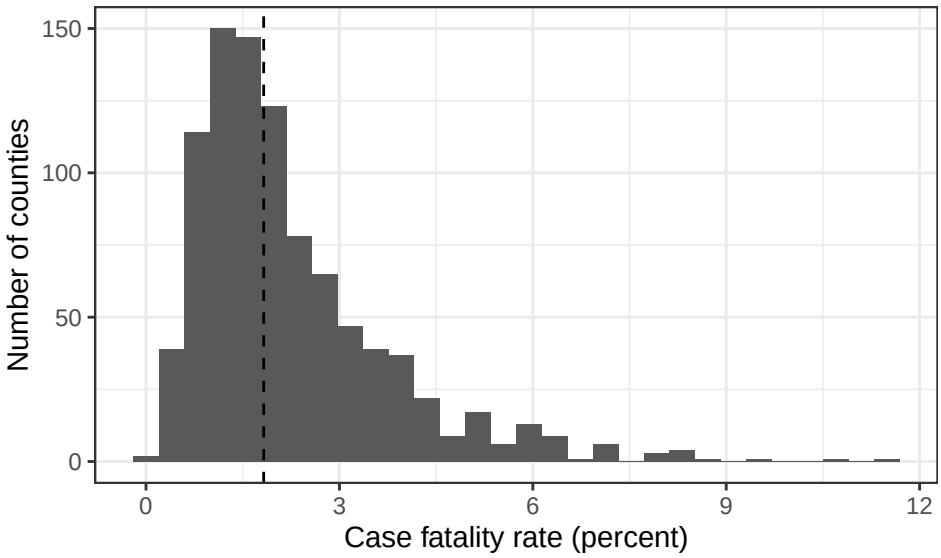

Figure 1: Distribution of case-fatality rate; vertical dashed line indicates the median.

Figure 1 shows a histogram of the case-fatality rate. Most counties have case-fatality rates roughly between 1% and 3%, but there is a long right tail of counties with substantially higher rates.

- 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. by running the code below. Based on this table and plot, what region of the U.S. tended to have the highest overall case-fatality rates?

```r
# examine top 10 counties by case fatality rate
covid_data %>%
  select(county, state, case_fatality_rate) %>%
  arrange(desc(case_fatality_rate)) %>%
  head(10) %>%
  kable(format = "latex", row.names = NA,
        booktabs = TRUE, digits = 2,
        col.names = c("County", "State", "Case-fatality rate"),
        caption = "Top ten counties by case-fatality rate
        (expressed as a percentage).") %>%
  kable_styling(position = "center")
```

Table 1 shows the top ten counties by case-fatality rate and Figure 2 shows the geographic distribution of this metric across the U.S. It is apparent that the northeast suffered from the highest case-fatality rates.

Table 1: Top ten counties by case-fatality rate (expressed as a percentage).

County	State	Case-fatality rate
Orleans	New York	11.60
Sussex	New Jersey	10.70
Warren	New Jersey	9.61
Morris	New Jersey	8.54
Somerset	New Jersey	8.49
Essex	New Jersey	8.44
Hartford	Connecticut	8.30
Cape May	New Jersey	8.27
Bergen	New Jersey	7.84
Wayne	Michigan	7.76

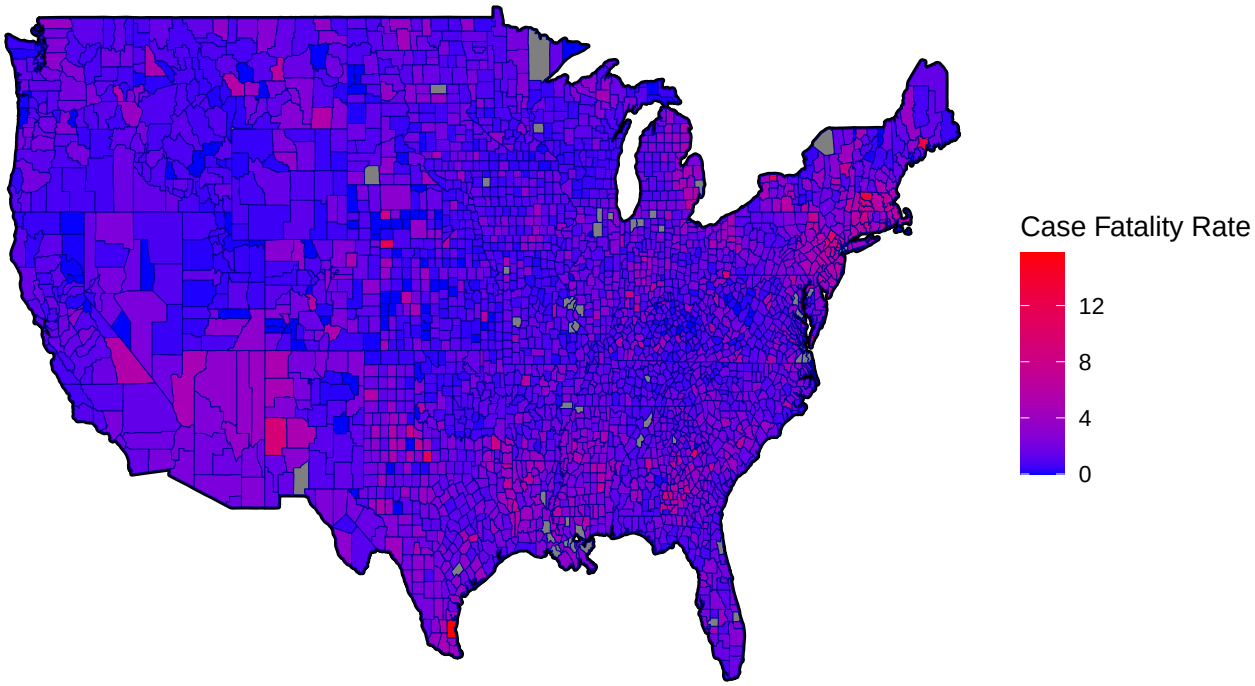

Figure 2: Geographic distribution of the case-fatality rate across the US in 2020.

2.2 Response-feature relationships

- 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.

```
# split into train and test
test_states = c("Alabama", "Arizona", "Arkansas",
                "California", "Colorado", "Connecticut")
covid_train = covid_data %>% filter(state %in% test_states)
covid_test = covid_data %>% filter(!(state %in% test_states))
```

- 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 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.

```
# plot case_fatality_rate against obesity_perc
p1 = covid_train %>%
  ggplot(aes(x = obesity_perc, y = case_fatality_rate)) +
  geom_point() +
  scale_y_log10() +
  geom_smooth(method = "lm", formula = "y~x", se = FALSE) +
  labs(x = "Obesity percentage",
       y = "Case Fatality Ratio",
       title = "Obesity percentage") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5))

# plot case_fatality_rate against uninsured
p2 = covid_train %>%
  ggplot(aes(x = uninsured, y = case_fatality_rate)) +
  geom_point() +
  scale_y_log10() +
  geom_smooth(method = "lm", formula = "y~x", se = FALSE) +
  labs(x = "Percent uninsured",
       y = "Case Fatality Ratio",
       title = "Percent uninsured") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5))

# plot case_fatality_rate against segregation_nonwhite_white
p3 = covid_train %>%
  ggplot(aes(x = segregation_nonwhite_white, y = case_fatality_rate)) +
  geom_point() +
  scale_y_log10() +
  geom_smooth(method = "lm", formula = "y~x", se = FALSE) +
  labs(x = "Residential segregation",
       y = "Case Fatality Ratio",
       title = "Residential segregation") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5))
```

```r
# plot case_fatality_rate against high_housing_costs
p4 = covid_train %>%
  ggplot(aes(x = high_housing_costs, y = case_fatality_rate)) +
  geom_point() +
  scale_y_log10() +
  geom_smooth(method = "lm", formula = "y~x", se = FALSE) +
  labs(x = "High housing costs",
       y = "Case Fatality Ratio",
       title = "High housing costs") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5))

# combine the plots
cowplot::plot_grid(p1, p2, p3, p4, nrow = 2)
```

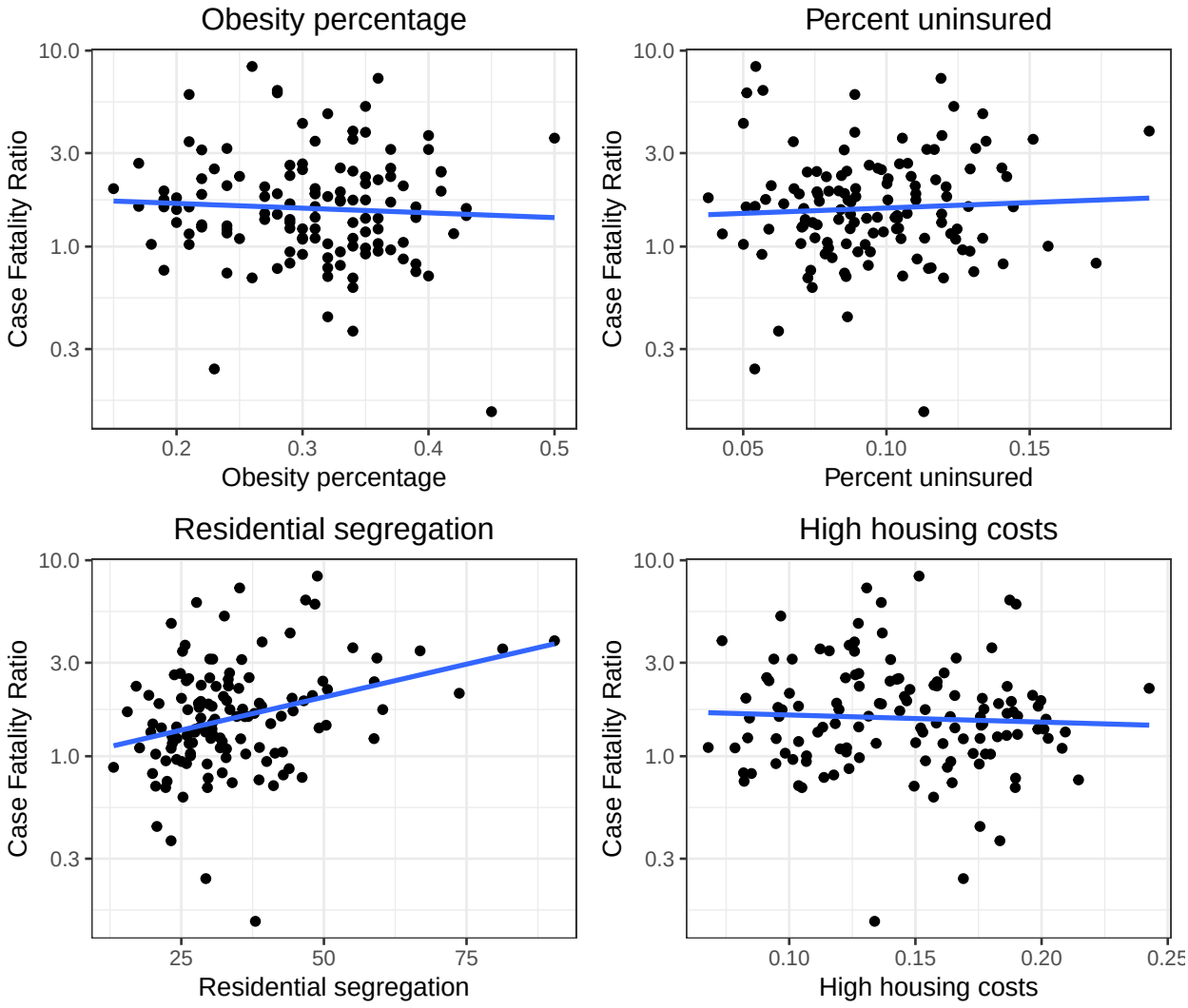

Figure 3: Case-fatality rate plotted against one feature in each of the four categories.

- 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?

Figure 3 shows how the case-fatality ratio varies with each of the four given features. There do not appear to be strong relationships with any of these variables except `segregation_nonwhite_white`, which suggests that highly segregated counties have higher case-fatality ratios. This may be the case because residents of highly segregated counties may have less access to healthcare resources.

3 Modeling

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

3.1 Ridge regression

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

```
set.seed(1)
ridge_fit = cv.glmnet(case_fatality_rate ~ . - state - county - fips,
                      alpha = 0,
                      nfolds = 10,
                      data = covid_train)
```

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

```
plot(ridge_fit)
```

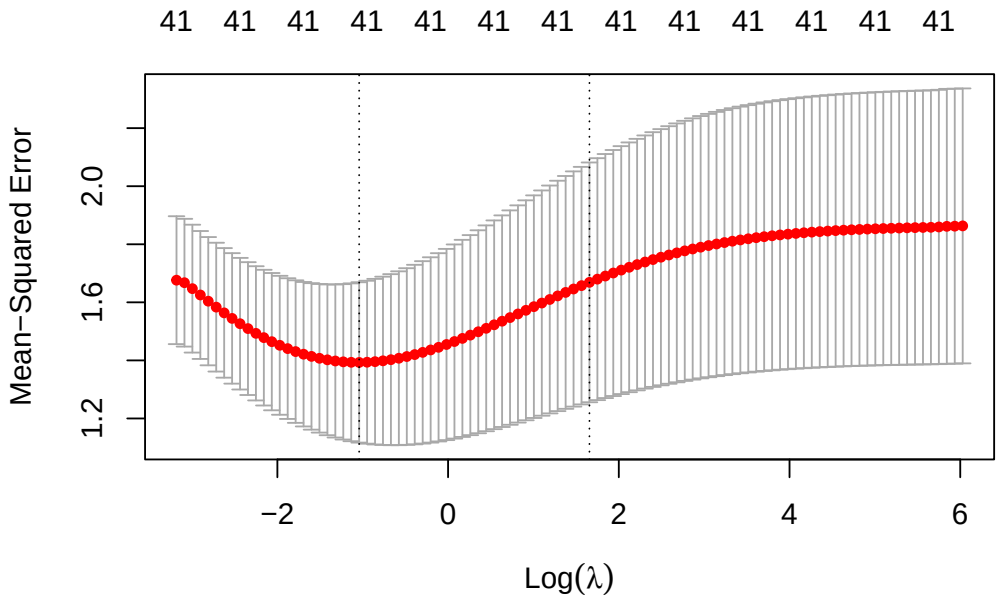

Figure 4: Ridge CV plot.

Figure 4 shows the CV plot for ridge regression. We have `lambda.min` = 0.353 and `lambda.1se` = 5.239. These two are indicated in the plot as the left and right vertical dashed lines, respectively.

- 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?

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

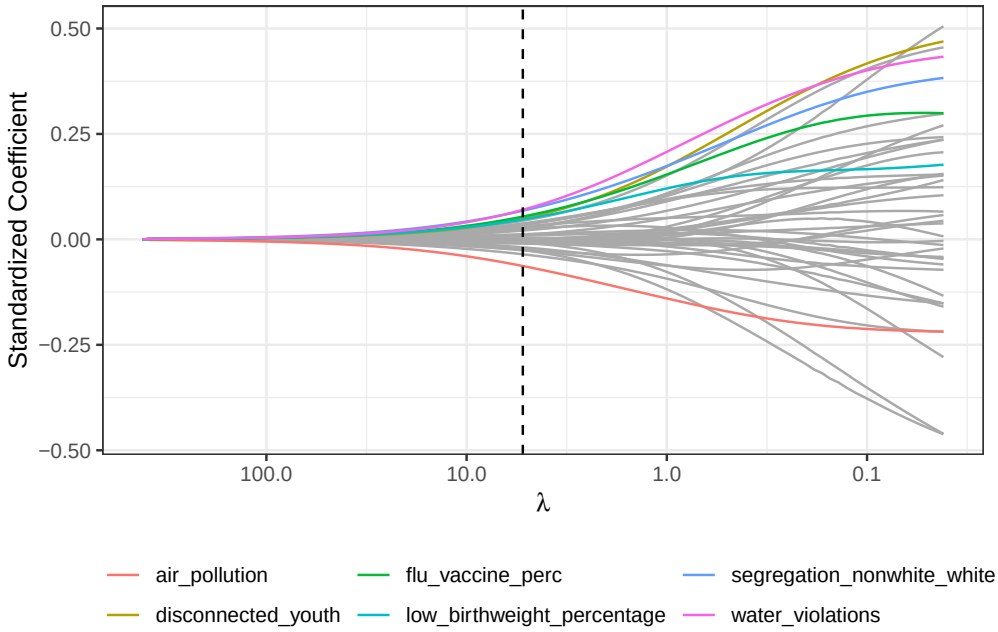

Figure 5: Ridge trace plot.

The feature `air_pollution` appears to have the strongest negative impact on the case-fatality rate. The reason for this relationship is not apparent, and the ridge regression result does not imply a statistically significant relationship.

3.2 Lasso regression

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

```
set.seed(1)
lasso_fit = cv.glmnet(case_fatality_rate ~ . - state - county - fips,
                      alpha = 1,
                      nfolds = 10,
                      data = covid_train)
```

- 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?

```
plot(lasso_fit)
```

Figure 6 shows the lasso CV plot. The leftmost edge corresponds to $\lambda \approx 0$, i.e. the ordinary linear regression model. It performs poorly due to high variance.

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

```
lasso_fit$nzero[lasso_fit$lambda == lasso_fit$lambda.1se]
```

```
## s14
##  11
```

The number of features with nonzero coefficients is 11.

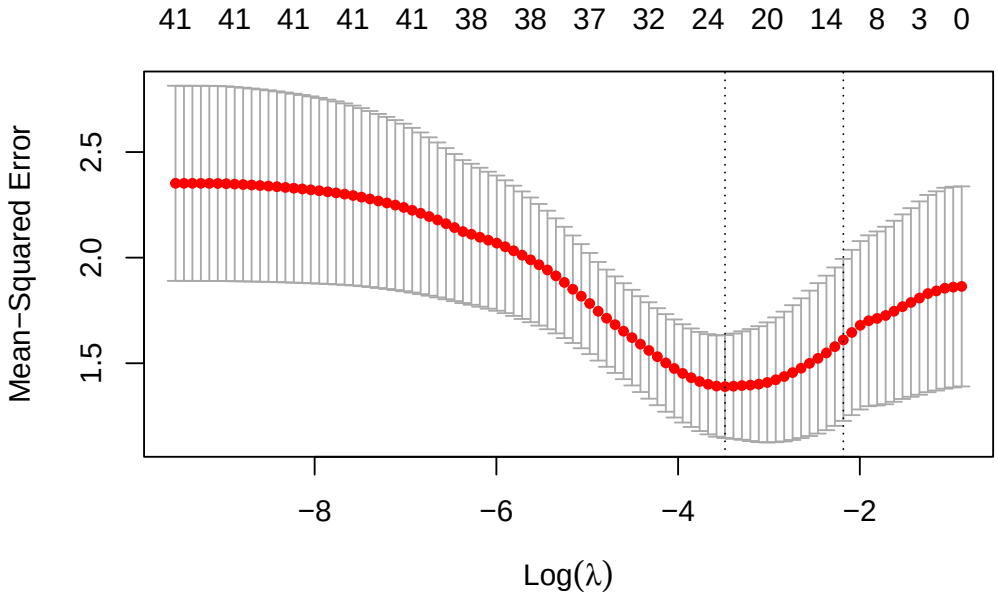

Figure 6: Lasso CV plot.

- 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?

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

Figure 7 shows the lasso trace plot. The first feature entering the model with a positive coefficient is `segregation_nonwhite_white`. The first feature entering the model with a positive coefficient is `air_pollution`.

- 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.

```
beta_hat_std = extract_std_coefs(lasso_fit, covid_train)
beta_hat_std %>%
  filter(coefficient != 0) %>%
  arrange(desc(abs(coefficient))) %>%
  kable(format = "latex", row.names = NA,
        booktabs = TRUE, digits = 2,
        col.names = c("Feature", "Coefficient"),
        caption = "Standardized coefficients for features in the lasso
        model based on the one-standard-error rule.") %>%
  kable_styling(position = "center")
```

Table 2 shows the standardized coefficients for features in the lasso model based on the one-standard-error rule. The coefficient of `flu_vaccine_perc` is 0.11. Checking the data, we see that `flu_vaccine_perc` is coded as a decimal. Therefore, an additional 10% vaccinated against the flu leads to an increase in the case-fatality percent by 0.011. One would expect higher flu vaccination rates to lead to lower case-fatality ratios.

3.3 Performance evaluation

- 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,

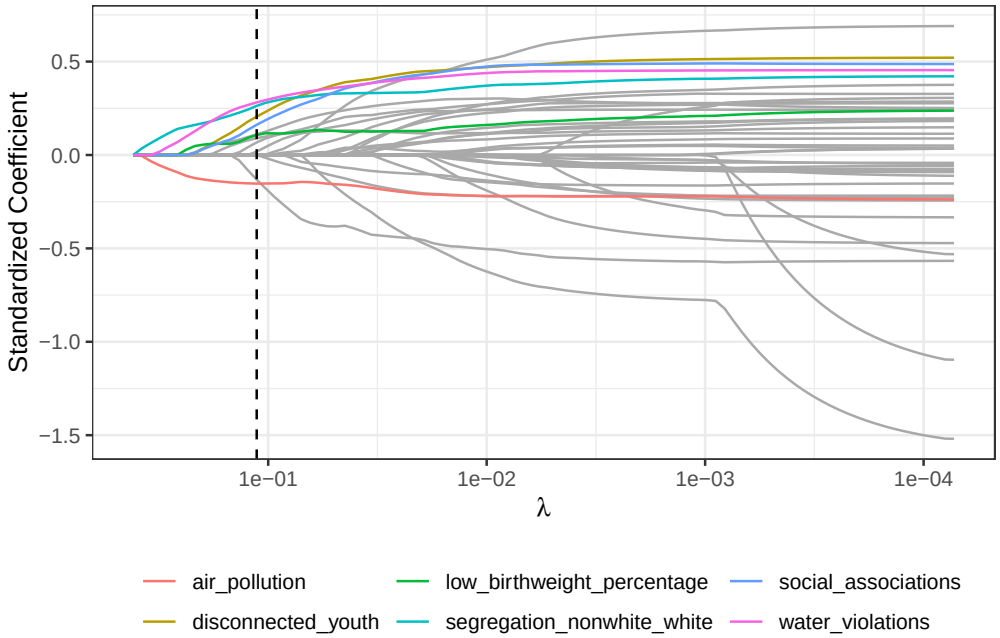

Figure 7: Lasso trace plot.

Table 2: Standardized coefficients for features in the lasso model based on the one-standard-error rule.

Feature	Coefficient
water_violations	0.28
segregation_nonwhite_white	0.26
disconnected_youth	0.21
social_associations	0.16
air_pollution	-0.15
smoke_perc	-0.13
flu_vaccine_perc	0.11
low_birthweight_percentage	0.11
mammogram_perc	0.09
unemployment	0.07
mentalhealth_ratio	0.02

Table 3: Root-mean-squared prediction errors for lasso, ridge, and intercept-only models.

Ridge	Lasso	Intercept-only
1.56	1.54	1.55

which predicts the mean case-fatality ratio in the training data for all counties. Print these three RMSE values in a nice table.

```
# ridge prediction error
ridge_predictions = predict(ridge_fit,
                            newdata = covid_test,
                            s = "lambda.1se") %>%
  as.numeric()
ridge_RMSE = sqrt(mean((ridge_predictions-covid_test$case_fatality_rate)^2))

# lasso prediction error
lasso_predictions = predict(lasso_fit,
                            newdata = covid_test,
                            s = "lambda.1se") %>%
  as.numeric()
lasso_RMSE = sqrt(mean((lasso_predictions-covid_test$case_fatality_rate)^2))

# intercept-only prediction error
training_mean_response = mean(covid_test$case_fatality_rate)
constant_RMSE = sqrt(mean((training_mean_response-covid_test$case_fatality_rate)^2))

# print nice table
tibble(Ridge = ridge_RMSE, Lasso = lasso_RMSE, `Intercept-only` = constant_RMSE) %>%
    kable(format = "latex", row.names = NA,
        booktabs = TRUE, digits = 2,
        caption = "Root-mean-squared prediction errors for lasso, ridge,
        and intercept-only models.") %>%
  kable_styling(position = "center")
```

Table 3 shows the root-mean-squared prediction errors for lasso, ridge, and intercept-only models.

- 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.

We see that the lasso performs (very marginally) better than ridge, and both of these penalized regression methods perform about the same as the intercept-only model. This suggests that variance is quite high in this problem, meaning that simpler models can perform as well or better than more complex models.

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.