

STAT 471: Midterm Exam

Name

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

Contents

Instructions	1
Socioeconomics and the COVID-19 case-fatality rate	3
1 Wrangling (25 points for correctness; 5 points for presentation)	3
1.1 Import (5 points)	3
1.2 Transform (15 points)	4
1.3 Merge (5 points)	5
2 Exploration (30 points for correctness; 5 points for presentation)	6
2.1 Response distribution (15 points)	6
2.2 Response-feature relationships (15 points)	8
3 Modeling (30 points for correctness; 5 points for presentation)	10
3.1 Ridge regression (8 points)	10
4 Appendix: Descriptions of features	15

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
```

```
## Warning: package 'kableExtra' was built under R version 4.1.1
```

```
library(cowplot) # for side by side plots
```

```
## Warning: package 'cowplot' was built under R version 4.1.1
```

```
library(glmnetUtils) # to run ridge and lasso
```

```
## Warning: package 'glmnetUtils' was built under R version 4.1.1
```

```
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
```

```
## Warning: package 'tibble' was built under R version 4.1.1
```

```
## Warning: package 'tidyr' was built under R version 4.1.1
```

```
## Warning: package 'readr' was built under R version 4.1.1
```

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).

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

## 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

## # 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
```

```

county_health_data <- read_tsv(file = "../data/county_health_data.tsv")

## 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.
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 (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.

- 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.]

```

# summarizing the data
case_data <- case_data_raw %>%
  filter(year(date) == 2020) %>%
  group_by(fips, county, state) %>%
  summarise(cases = sum(cases),
            deaths = sum(deaths)) %>%
  ungroup() %>%

```

```
mutate(case_fatality_rate = deaths/cases) %>%
select(-c("cases", "deaths"))
```

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

- 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?

`case_data`

```
## # A tibble: 3,273 x 4
##   fips county state case_fatality_rate
##   <chr> <chr> <chr> <dbl>
## 1 01001 Autauga Alabama 0.0153
## 2 01003 Baldwin Alabama 0.0110
## 3 01005 Barbour Alabama 0.0116
## 4 01007 Bibb Alabama 0.0181
## 5 01009 Blount Alabama 0.0110
## 6 01011 Bullock Alabama 0.0264
## 7 01013 Butler Alabama 0.0401
## 8 01015 Calhoun Alabama 0.0147
## 9 01017 Chambers Alabama 0.0369
## 10 01019 Cherokee Alabama 0.0233
## # ... with 3,263 more rows
```

```
length(unique(county_health_data$fips))
```

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

There are **3273** counties in `case_data`, but there are only **935** counties in `county_health_data`. The likely cause of this is that it is much harder to compile a set of all 41 health indicators in the `county_health_data` than it is covid cases/deaths. More counties are willing to not only spend the time and resources making the covid data publicly available compared to the socioeconomic data but also collect the covid data in the first place.

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).

#joining 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 (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.

```
# calculating median
median(covid_data$case_fatality_rate)
```

```
## [1] 0.0183
```

The median case fatality rate is 0.0183.

- 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.

```
# creating the graph
ggplot(covid_data, aes(x=case_fatality_rate)) +
  geom_histogram() + theme_bw() +
  geom_vline(xintercept = median(covid_data$case_fatality_rate), linetype = "dashed") +
  labs(x = "Case Fatality Rate", y = "Count")
```

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

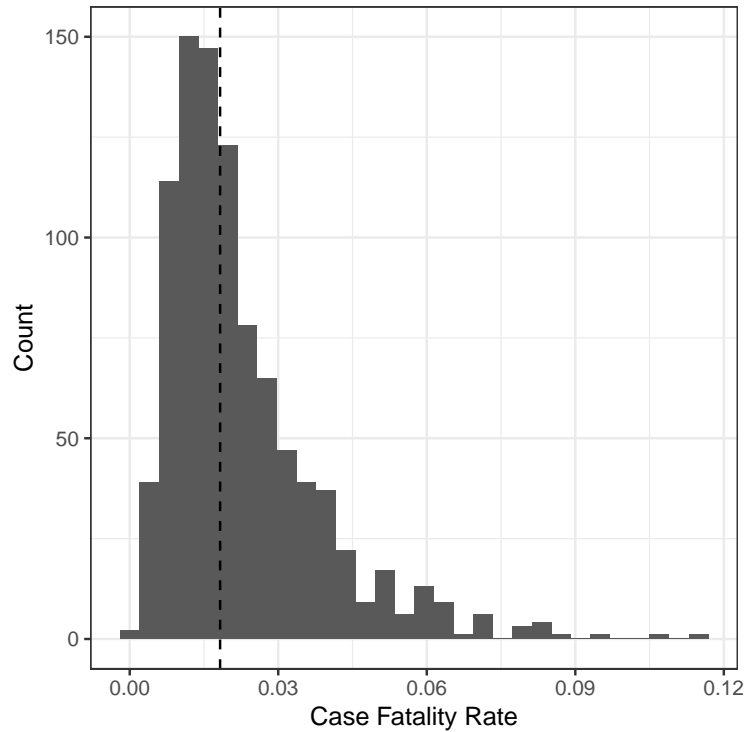


Figure 1: Title

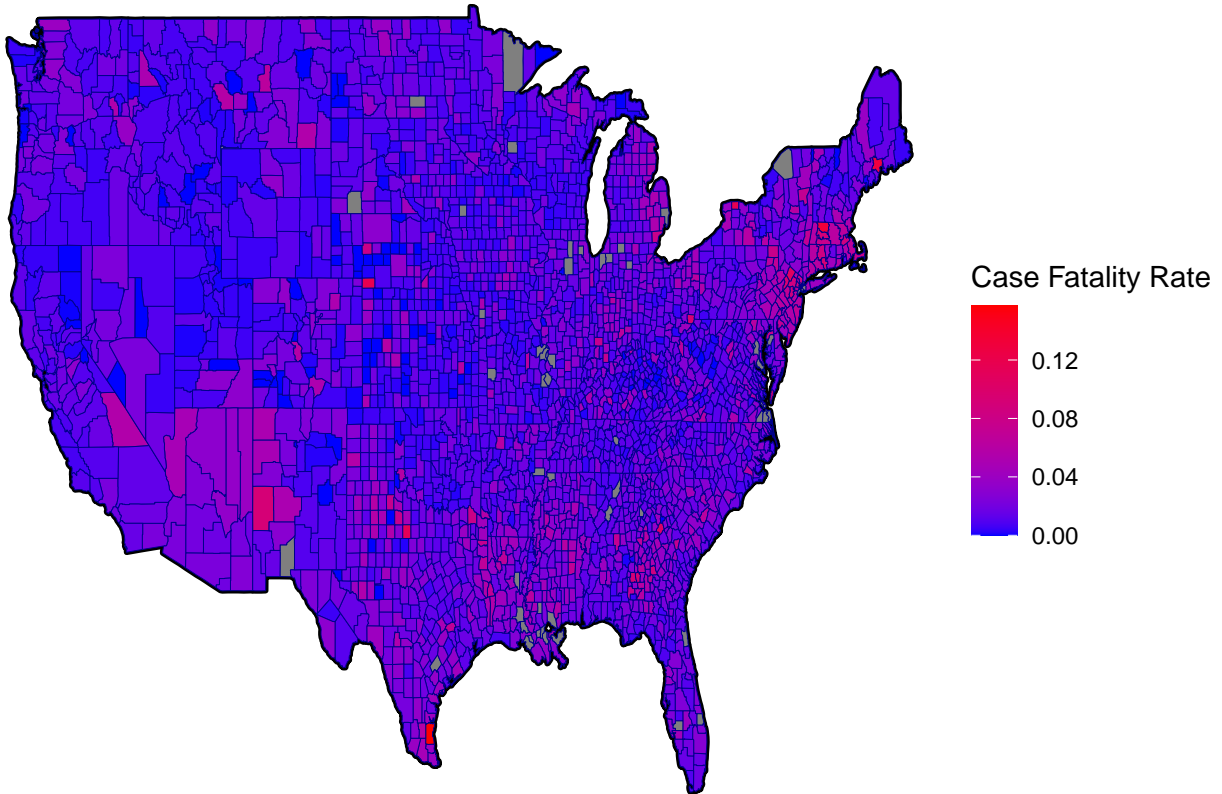
In Figure 1, the distribution appears to be skewed to the right. Most of the case fatality rates are very close to 0. The median and peak are around 0.02. However, there are some outliers that reach a ~10% case fatality rate.

- 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?

```
#Using covid data as not sure which one is wanted and that seems like a good default
covid_data %>%
  arrange(desc(case_fatality_rate)) %>%
  slice(1:10) %>% # getting the top 10
  select(fips, county, state, case_fatality_rate) %>%
  kable(format = "latex", row.names = NA,
        booktabs = TRUE, digits = 2,
        caption = "Deadliest Counties USA") %>%
  kable_styling(position = "center", latex_options = "HOLD_position")
```

Table 1: Deadliest Counties USA

fips	county	state	case_fatality_rate
36073	Orleans	New York	0.12
34037	Sussex	New Jersey	0.11
34041	Warren	New Jersey	0.10
34027	Morris	New Jersey	0.09
34035	Somerset	New Jersey	0.08
34013	Essex	New Jersey	0.08
09003	Hartford	Connecticut	0.08
34009	Cape May	New Jersey	0.08
34003	Bergen	New Jersey	0.08
26163	Wayne	Michigan	0.08



In Table 1, we can see that the deadliest counties appear the northeast (specifically around New York). This is reflected in Figure ??, where there are spots of red around the city of New York. This also matches my intuition as I know New York was hit hard, partly due to Cuomo’s mismanagement and partly because it was hit early.

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.

```
covid_test <- covid_data %>%
  filter(state %in% c("Alabama", "Arizona",
    "Arkansas", "California", "Colorado", "Connecticut")) %>%
  select(-c(state, county, fips))
covid_train <- covid_data %>%
  filter(!state %in% c("Alabama", "Arizona",
    "Arkansas", "California", "Colorado", "Connecticut")) %>%
  select(-c(state, county, fips))
```

- 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.

```
obesity <- ggplot(covid_data, aes(x=case_fatality_rate,
  y= obesity_perc)) +
  geom_point() +
  theme_bw() +
  scale_y_log10() +
  labs(x = "Case Fatality Rate", y = "Obesity Percentage")

uninsured <- ggplot(covid_data, aes(x=case_fatality_rate,
  y= uninsured)) +
  geom_point() +
  theme_bw() +
  scale_y_log10() +
  labs(x = "Case Fatality Rate", y = "Percentage Under 65 Uninsured")

segregation <- ggplot(covid_data, aes(x=case_fatality_rate,
  y= segregation_nonwhite_white)) +
  geom_point() +
  theme_bw() +
  scale_y_log10() +
  labs(x = "Case Fatality Rate", y = "Nonwhite Residential Segregation Index")

housing <- ggplot(covid_data, aes(x=case_fatality_rate,
  y= high_housing_costs)) +
  geom_point() +
  theme_bw() +
  scale_y_log10() +
  labs(x = "Case Fatality Rate", y = "Percentage of Houses with High Housing Costs")

plot_grid(obesity, uninsured, segregation, housing, ncol = 2)
```

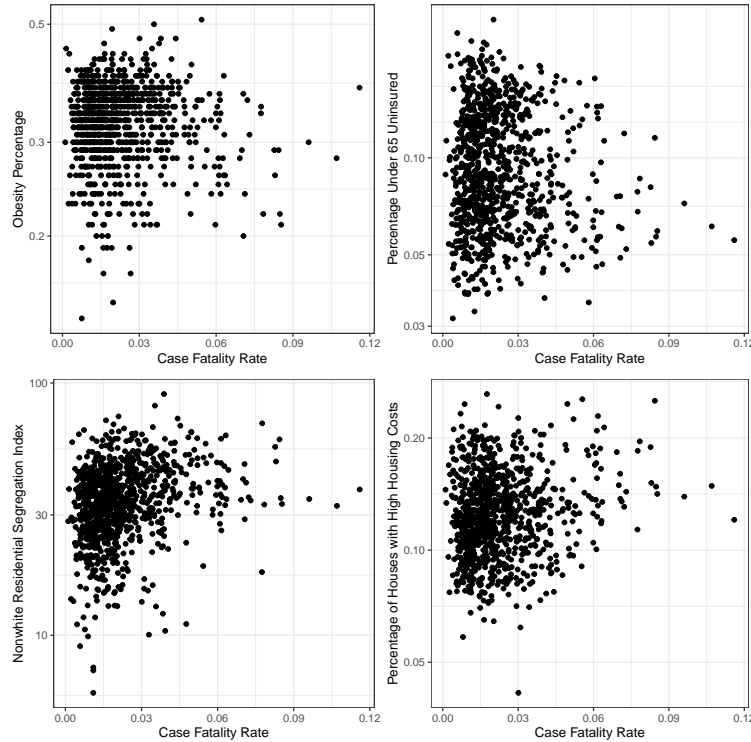


Figure 2: Scatter plots of Case Fatality Rate vs. Various Factors

- 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?

High housing costs appears to have the strongest relationship with case-fatality-ratio with a positive relationship between housing costs and case fatality rates. This relationship is likely due to the fact that communities burdened by high housing costs are likely to be poorer. This means that they can't afford adequate medical care. This variable is also likely to correlate with majority minority communities, which often are the victims of prejudiced medical care.

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`.

```
set.seed(1) # for replicability (do not change)
ridge_fit = cv.glmnet(case_fatality_rate ~ ., # formula notation, as usual
                      alpha = 0, # alpha = 0 for ridge
                      nfolds = 10, # number of folds
                      data = covid_train) # data to run ridge on
```

- 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)
```

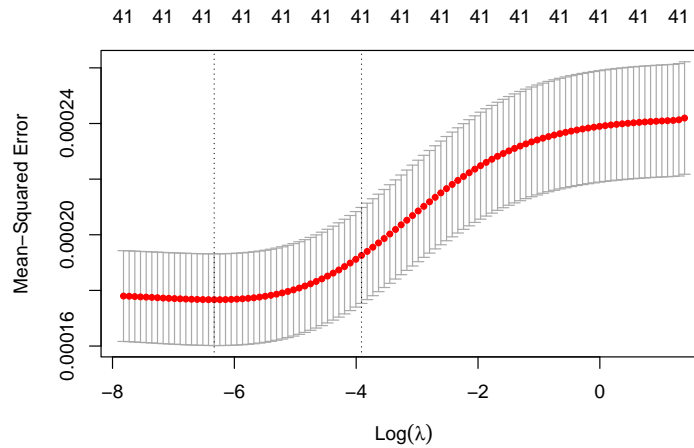


Figure 3: Ridge Fit CV Errors

The values of the min lambda and 1se lambda are `ridge_fit$lambda.min` and `ridge_fit$lambda.1se`, respectively. In Figure 3, These are the left and right dashed lines, again 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)
```

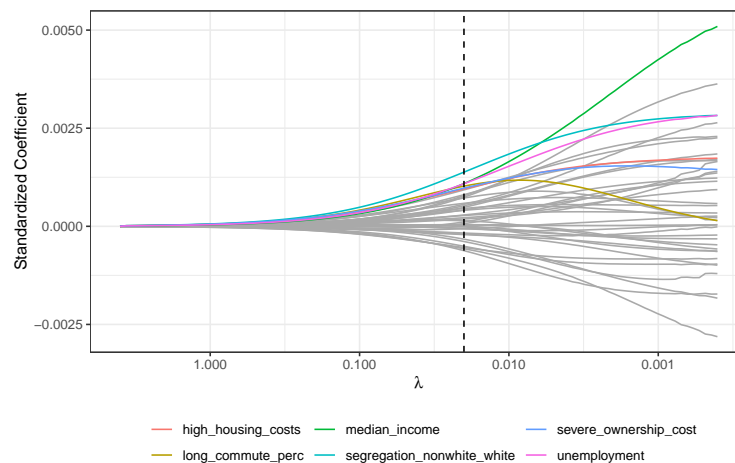


Figure 4: Ridge Regression Trace Plot

```
min = min(coef(ridge_fit, s = "lambda.1se"))
tibble(coef = as.vector(coef(ridge_fit, s = "lambda.1se")),
       rownames = as.vector(rownames(coef(ridge_fit, s = "lambda.1se")))) %>%
  arrange(coef) %>%
  slice(1)
```

```
## # A tibble: 1 x 2
```

```
##      coef rownames
##      <dbl> <chr>
## 1 -0.0431 teen_births
```

In Figure ??, the figure with the most negative impact is `teen_births`. This does not have an immediate relationship, but it could signal a lot of young people who are less likely to die. However, there is also no statistically significant relationship between this feature and the case-fatality rate, as penalized regressions do not usually have p-values associated to them. ##

Lasso regression (14 points)

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

```
set.seed(1) # for replicability (do not change)
lasso_fit = cv.glmnet(case_fatality_rate ~ ., # formula notation, as usual
                      alpha = 1, # alpha = 0 for ridge
                      nfolds = 10, # number of folds
                      data = covid_train) # data to run ridge on
```

- 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)
```

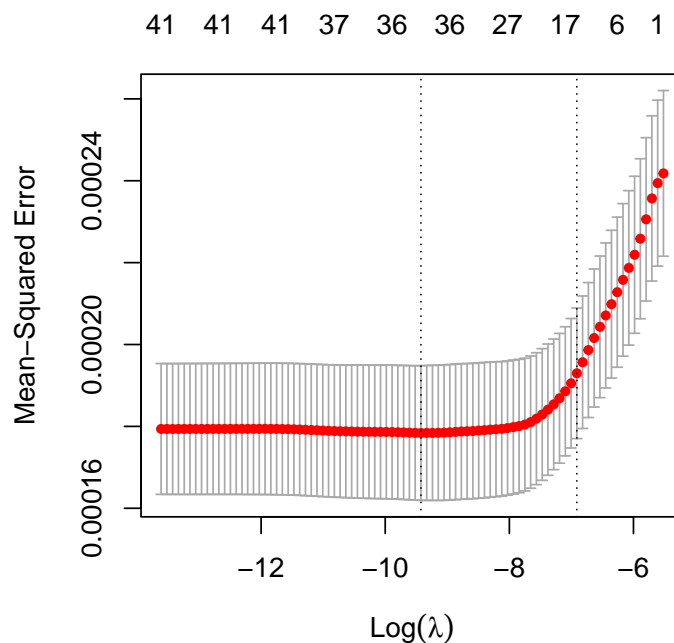


Figure 5: Lasso Fit CV Errors

The model at the left-most edge is ordinary least squares model. It performs poorly as there are too many features compared to the amount of data that we have, so it is overfitting.

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

There are only 17 features for the lasso model following the one-standard error rule.

- 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)
```

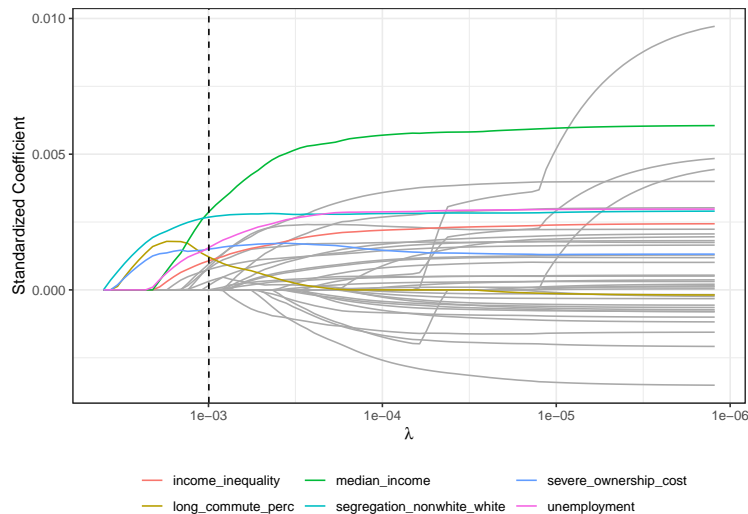


Figure 6: Lasso Regression Trace Plot

```
min = min(coef(ridge_fit, s = 0.0001))
tibble(coef = as.vector(coef(lasso_fit, s = 0.0001)),
       rownames = as.vector(rownames(coef(lasso_fit, s = 0.0001)))) %>%
  arrange(coef) %>%
  slice(1)
```

```
## # A tibble: 1 x 2
##   coef rownames
##   <dbl> <chr>
## 1 -0.195 teen_births
```

In Figure 6, the first positive feature is `segregation_nonwhite_white`. The first negative feature is `teen_births`

- 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.

```
tibble(coef = as.vector(coef(lasso_fit, s = "lambda.1se")),
       rownames = as.vector(rownames(coef(lasso_fit, s = "lambda.1se")))) %>%
  filter(abs(coef) >= 0.01) %>%
  arrange(desc(abs(coef))) %>%
  kable(format = "latex", row.names = NA,
        booktabs = TRUE, digits = 2,
        caption = "Coefficients for Lambda.1se Lasso Regression") %>%
  kable_styling(position = "center", latex_options = "HOLD_position")
```

Table 2: Coefficients for Lambda.1se Lasso Regression

coef	rownames
0.13	unemployment
0.05	severe_ownership_cost
-0.04	(Intercept)
0.02	high_housing_costs
0.02	low_birthweight_percentage
0.02	inactive_perc
0.01	homeownership
0.01	long_commute_perc

```
tibble(coef = as.vector(coef(lasso_fit, s = "lambda.1se")),
       rownames = as.vector(rownames(coef(lasso_fit, s = "lambda.1se")))) %>%
  filter(rownames == "flu_vaccine_perc")
```

```
## # A tibble: 1 x 2
##       coef rownames
##   <dbl> <chr>
## 1 0.00242 flu_vaccine_perc
```

The coefficient for flu_vaccine_perc is 0.00242 This means that the feature is not influencing the regression at the lambda.1se value. Since it is positive, perhaps it is a sign that a county is more health conscious and willing to take precautions. ## 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.

```
lasso_predictions = predict(lasso_fit,
                             newdata = covid_test,
                             s = "lambda.1se") %>% as.numeric()
RMSE_lasso = sqrt(mean((lasso_predictions - covid_test$case_fatality_rate)^2))

ridge_predictions = predict(ridge_fit,
                             newdata = covid_test,
                             s = "lambda.1se") %>% as.numeric()
RMSE_ridge = sqrt(mean((ridge_predictions - covid_test$case_fatality_rate)^2))

intercept_only = predict(lasso_fit,
                           newdata = covid_test,
                           s = 1) %>% as.numeric()
Rmse_intercept = sqrt(mean((intercept_only - covid_test$case_fatality_rate)^2))

tribble(
  ~Lasso, ~Ridge, ~Intercept,
  RMSE_lasso, RMSE_ridge, Rmse_intercept
) %>%
  kable(format = "latex", row.names = NA,
        booktabs = TRUE, digits = 2,
        caption = "RMSE Values Data") %>%
  kable_styling(position = "center", latex_options = "HOLD_position")
```

Table 3: RMSE Values Data

Lasso	Ridge	Intercept
0.01	0.01	0.01

- 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.

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.