CM 2.4 - Multiple linear regression

Math 530/630

Contents

Logistics

- A complete knitted `html` file is due on Sakai by beginning of the next class.
- This lab is structured to be similar to this Case Study on Seattle House Prices from ModernDive. Please open it and follow along with both datasets!
- The structure of this lab is as follows:
  - In the lab, you'll:
    - Complete and interpret EDA Part I (univariate)
    - Complete and interpret EDA Part II (bivariate)
    - Fit a regression model, obtain the regression table, and attempt to interpret the three values that define the regression plane
  - In the next class, together we'll review the above models and:
    - Perform analysis of observed/fitted values and residuals following ModernDive
    - Explore residual analysis following ModernDive

Overview

We'll work with data from this 538 article. In the article, the authors describe collecting data on key socioeconomic factors for each state, including indicators for:

- education (percent of adults 25 and older with at least a high school degree, as of 2009)
- diversity
  - percent nonwhite population (2015), and
  - percent noncitizen population (2015).
- geographic heterogeneity (percent population in metropolitan areas, 2015)
- economic health
  - median household income,
  - 2016 seasonally adjusted unemployment (September 2016),
  - percent poverty among white people (2015), and

    - income inequality (as measured by the Gini index, 2015)
- percent of the population voted for Donald Trump.

In this lab, we'll use a subset of these variables to predict hate crimes in the US. There are two possible outcome variables here: (1) pre-election data from the FBI, and (2) post-election data from the Southern Poverty Law Center. We'll focus on the pre-election data in this lab.

The Data

This data is included in the `fivethirtyeight` package in the `hate_crimes` data frame, which we'll refer to as the "Hate crimes" dataset. You can use `?hate_crimes` to read more about it and the variables.

You'll need to load these packages to do this lab:

```r
library(fivethirtyeight) # new to you!
library(moderndive)
library(skimr)
library(tidyverse)
library(GGally) # new to you!
```

We'll use `hate_crimes` to demonstrate multiple regression with:

1. A numerical outcome variable y, in this case average annual hate crimes per 100,000 population, FBI, 2010-2015 (`avg_hatecrimes_per_100k_fbi`)
2. Three possible explanatory variables:
    1. A first numerical explanatory variable x_1: percent of adults in each state 25 and older with at least a high school degree (2009) (`share_pop_hs`)
    2. A second numerical explanatory variable x_2: each state's income inequality (as measured by the Gini index, 2015) (`gini_index`)
    3. A third numerical explanatory variable x_3: each state's percent population that voted for Donald Trump (`share_vote_trump`). At a later stage, we'll convert this variable to a factor.

EDA, Part I

Recall that a new exploratory data analysis involves three things:

- Looking at the raw values and the structure of the data.
- Computing summary statistics of the variables of interest.
- Creating informative visualizations.

General functions we use below- add narrative to interpret each!:

- `dplyr::glimpse()`
- `skimr::skim()`
- `ggplot2::ggplot()`
    - `geom_histogram()` or `geom_density()` for numeric continuous variables
    - `geom_bar()` or `geom_col()` for categorical variables

At this stage, you may also find your want to use `filter`, `mutate`, `arrange`, `select`, or `count`. Let your questions lead you! Feel free to add onto the EDA that follows.

Look at the raw values

- How many states are here? Are they all "states"?

```r
glimpse(hate_crimes)
```

- How many rows do we have per state? Is there ever more than 1 row per state?

```r
hate_crimes %>%
  count(state, sort = TRUE)
```

Compute summary statistics

Let's select just the variables we need first.

```r
hate_demo <- hate_crimes %>%
  select(state, avg_hatecrimes_per_100k_fbi, share_pop_hs, gini_index,
         share_vote_trump)
```

Following the narrative in ModernDive, write a few sentences describing the output here.

```r
skim(hate_demo)
```

Create informative visualizations

First let's look at the outcome variable:

```r
# Density of hate crimes (DV):
ggplot(hate_demo, aes(x = avg_hatecrimes_per_100k_fbi)) +
  geom_density() +
  labs(x = "", title = "Hate Crimes")
```

Next we'll look at our three explanatory variables as continuous:

```r
# Histogram of share_pop_hs (IV):
ggplot(hate_demo, aes(x = share_pop_hs)) +
  geom_density() +
  labs(x = "", title = "HS")

# Histogram of gini (IV):
ggplot(hate_demo, aes(x = gini_index)) +
  geom_density() +
  labs(x = "", title = "Gini")

# Histogram of trump (IV):
ggplot(hate_demo, aes(x = share_vote_trump)) +
  geom_density() +
  labs(x = "", title = "Trump")
```

Let's make `share_vote_trump` a categorical variable:

```r
hate_demo <- hate_demo %>%
  mutate(
    cat_trump = case_when(
      share_vote_trump < .5 ~ "less than half",
      TRUE ~ "more than half"
      )) %>%
  mutate(cat_trump = as.factor(cat_trump)) %>%
  select(-share_vote_trump)
```

Following the narrative in ModernDive, write a few sentences describing the output here.

EDA, Part II

Part I of this EDA was univariate in nature in that we only considered one variable at a time. The goal of modeling, however, is to explore relationships between variables. Specifically, we care about bivariate relationships between pairs of variables. But with 1 outcome and 3 explanatory variables, that means we have $3 \times 2 = 6$ correlations to compute.

For simple regression, we calculated correlation coefficients between the outcome and explanatory variables. For multiple regression, your EDA should involve multiple correlation coefficients. We'll use the `cor()` function to do this. You'll want to first `select` only numeric variables first.

Use this code as an example:

```
data %>%
  select(-my_char_var, -my_factor_var) %>%
  cor()
```

To produce this output:

```
                            avg_hatecrimes_per_100k_fbi share_pop_hs
avg_hatecrimes_per_100k_fbi                           1           NA
share_pop_hs                                         NA    1.0000000
gini_index                                           NA   -0.5920518
                            gini_index
avg_hatecrimes_per_100k_fbi         NA
share_pop_hs                -0.5920518
gini_index                   1.0000000
```

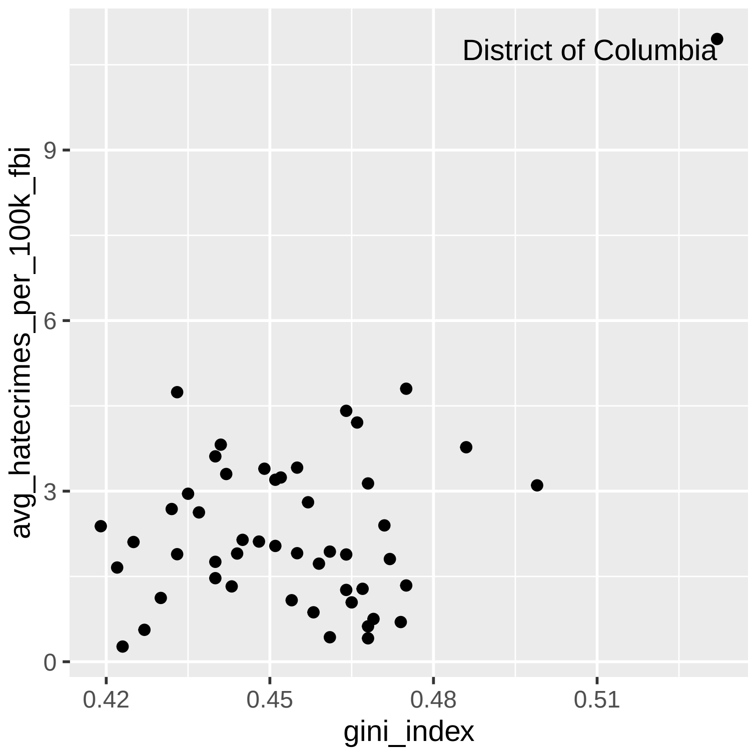
Lots of `NA` correlations though! Try this code instead:

```
data %>%
  select(-my_char_var, -my_factor_var) %>%
  cor(., use = "pairwise.complete.obs")
```

You could do the same thing in the `corrr` package, using the `correlate` function:

```
library(corrr)
data %>%
  select(-my_char_var, -my_factor_var) %>%
  correlate()
```

We also want to create scatterplots to see the association between each pair of variables in the model (both between the explanatory and the outcome, but also between all explanatory variables with each other). Let's start with the `gini_index`:



That's a lot of plots! However, we can actually do all of these comparisons with one function! We'll use `GGally::ggpairs()` to create a pairwise comparison of multivariate data. This includes what is known as a "Generalized Pairs Plot" which is an improved version of a scatterplot matrix. This function provides two different comparisons of each pair of columns, and displays either the density (continuous numeric) or count (factors) of the respective variable along the diagonal. You can read more about the function and package here.

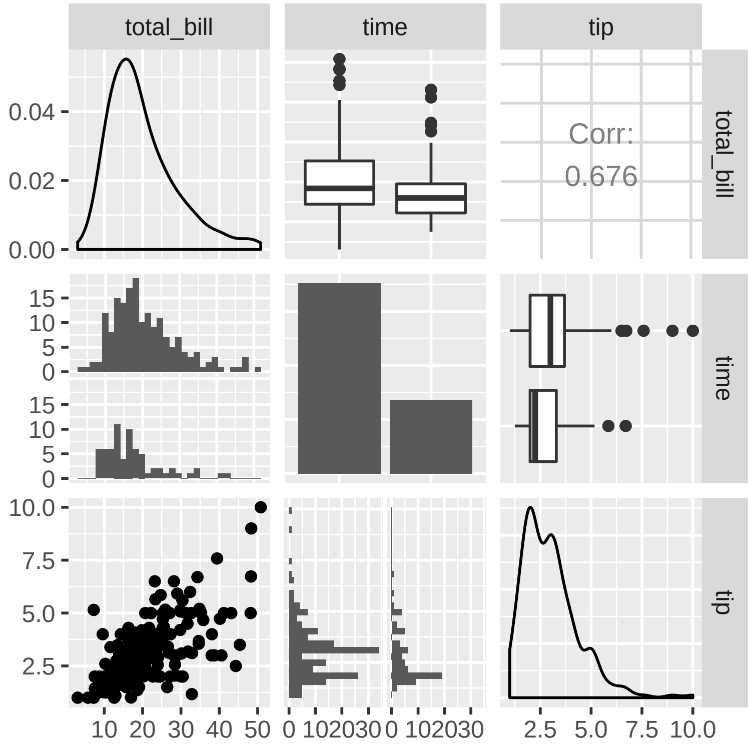
There are three pieces to the output: `lower`, `upper`, and `diag`. Read more about the sections of the matrix here.

Here is how you can use the function:

```r
data %>%
  select(-my_char_var) %>%
  ggpairs()
```

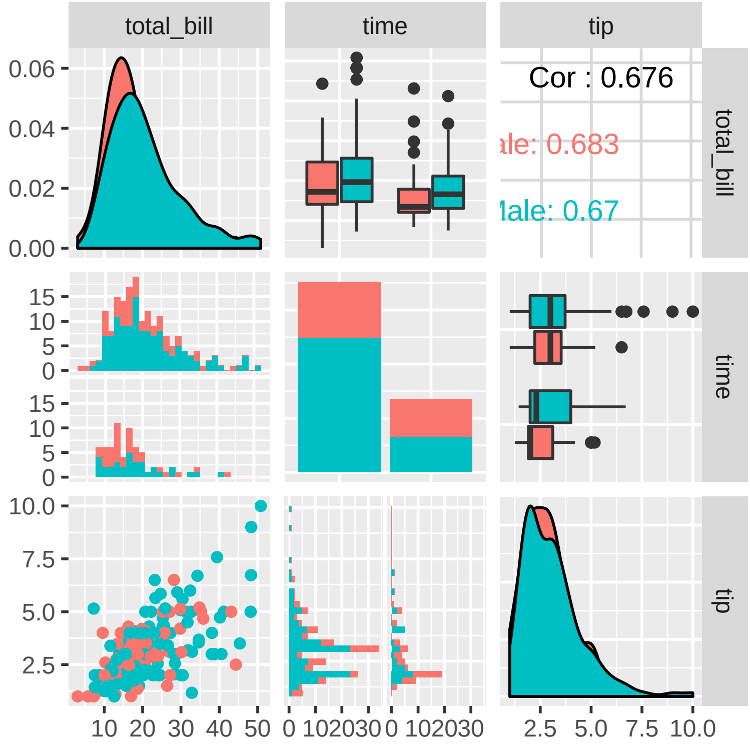
And here is some demo output of how to use it using a dataset called `tips`:

```r
data(tips, package = "reshape")
tips %>%
  select(total_bill, time, tip) %>%
  ggpairs()
```



And it builds from `ggplot2`, so you can add aesthetic mappings for color, etc. Hurray!

```
tips %>%
  ggpairs(., aes(color = sex),
          columns = c("total_bill", "time", "tip"))
```



Calculate all the correlation coefficients. Make a ggpairs plot between all explanatory and outcome variables. Following this narrative in ModernDive, write a few sentences describing the output here.

Multiple regression models

Do the following:

- Fit a multiple regression model and get the regression table. You'll be assigned **one** of the following models:
    1. Two numerical predictors with a + (gini_index and share_pop_hs)
    2. One numerical / one categorical with parallel slopes (gini_index and cat_trump)
    3. One numerical / one categorical interaction model (gini_index and cat_trump)
    4. Two numerical predictors with a * (gini_index and share_pop_hs)
- Sketch out the *modeling equation* for your model (not in your R Markdown)
    - Parallel slopes example here
    - Interaction model here
- Interpret the output from the regression table (in complete sentences, but you may use bullet points to organize)
    - Parallel slopes example here
    - Interaction model here
- Compare the coefficients from your multiple regression model to the "simple" correlation coefficients for each explanatory variable. Recall that in a simple linear regression:

$$b_{x_1} = r_{x_1\ y} \frac{s_y}{s_{x_1}}$$

- For those with the two numerical predictors, you may want to look into making a 3D scatterplot.
    - The numerical outcome variable y avg_hatecrimes_per_100k_fbi goes on the z-axis (vertical axis)
    - The two numerical explanatory variables form the "floor" axes. In this case
        - The first numerical explanatory variable x_1 share_vote_hs is on of the floor axes.
        - The second numerical explanatory variable x_2 gini_index is on the other floor axis.

```
library(plotly)
dim_scatter <- plot_ly(hate_demo,
                       x = ~share_pop_hs,
                       y = ~gini_index,
                       z = ~avg_hatecrimes_per_100k_fbi) %>%
  add_markers() %>%
  layout(scene = list(xaxis = list(title = 'HS'),
                     yaxis = list(title = 'Gini'),
                     zaxis = list(title = 'Hate Crimes')))
dim_scatter
```