Math 300 NTI Lesson 16

Multiple Regression - Related Topics

Professor Bradley Warner

June, 2022

Contents

Objectives

1. Find and interpret R-squared.
2. Use R-squared to select the best model.
3. Explain Simpson's paradox and confounding variables.

Reading

Chapter 6.3

Lesson

There are no learning checks for this lesson, so work through the code and ideas in the reading. Keep in mind there is no new additional material next lesson, so you'll have time next lesson to finish this document.

- The book discusses Occam's razor, which states that given the choice between a complex model and simple model pick the simple. This assumes similar performance. One way to measure performance is with R-squared. This is an internal measure of performance, it uses the data that was used to build the model. In a machine learning class, students can learn other ways to pick a model.
- R-squared always increases when we add a variable. Adjusted R-squared puts in a penalty for adding a variable. This penalty is just a heuristic. Analysts often use adjusted R-squared instead of R-squared.
- We preface our interpretation with the statement, "taking into account all the other explanatory variables in our model" in this section. This means we have to treat the other variables as at a constant value even though collinearity in practice may not allow this. It is only from an interpretation point of view that we use that statement.

- We must be aware of a phenomenon known as Simpson's Paradox, whereby overall trends that exist in aggregate either disappear or reverse when the data are broken down into groups. We will discuss this today or next lesson.

Setup

```
library(tidyverse)
library(moderndive)
library(skimr)
library(ISLR)
```

Recreate the analysis done in the book.

```
model_2_interaction <- lm(average_sat_math ~ perc_disadvan * size,
                          data = MA_schools)
get_regression_table(model_2_interaction)
```

```
## # A tibble: 6 x 7
##   term                     estimate std_error statistic p_value lower_ci upper_ci
##   <chr>                       <dbl>     <dbl>     <dbl>   <dbl>    <dbl>    <dbl>
## 1 intercept                 594.       13.3      44.7     0      568.     620.
## 2 perc_disadvan              -2.93      0.294    -9.96     0       -3.51    -2.35
## 3 size: medium              -17.8      15.8      -1.12     0.263  -48.9     13.4
## 4 size: large               -13.3      13.8      -0.962    0.337  -40.5     13.9
## 5 perc_disadvan:sizemedi~     0.146     0.371     0.393    0.694   -0.585    0.877
## 6 perc_disadvan:sizelarge     0.189     0.323     0.586    0.559   -0.446    0.824
```

```
model_2_parallel_slopes <- lm(average_sat_math ~ perc_disadvan + size,
                              data = MA_schools)
get_regression_table(model_2_parallel_slopes)
```

```
## # A tibble: 4 x 7
##   term          estimate std_error statistic p_value lower_ci upper_ci
##   <chr>            <dbl>     <dbl>     <dbl>   <dbl>    <dbl>    <dbl>
## 1 intercept       588.       7.61     77.3     0       573.     603.
## 2 perc_disadvan    -2.78     0.106   -26.1     0        -2.99    -2.57
## 3 size: medium    -11.9      7.54     -1.58    0.115   -26.7      2.91
## 4 size: large      -6.36     6.92     -0.919   0.359   -20.0      7.26
```

In a future lesson we will use the p-value and confidence intervals to determine the statistical importance of an explanatory variable.

```
get_regression_summaries(model_2_interaction)
```

```
## # A tibble: 1 x 9
##   r_squared adj_r_squared   mse  rmse sigma statistic p_value    df  nobs
##       <dbl>         <dbl> <dbl> <dbl> <dbl>     <dbl>   <dbl> <dbl> <dbl>
## 1     0.699         0.694 1107.  33.3  33.6      151.       0     5   332
```

```
get_regression_summaries(model_2_parallel_slopes)
```

```
## # A tibble: 1 x 9
##   r_squared adj_r_squared   mse  rmse sigma statistic p_value    df  nobs
##       <dbl>         <dbl> <dbl> <dbl> <dbl>     <dbl>   <dbl> <dbl> <dbl>
## 1     0.699         0.696 1109.  33.3  33.5      254.       0     3   332
```

R-squared

- Use R-squared to determine the model for the UT Austin teacher evaluation problem. (Objective 1 and 2)

Get the data

```
evals_ch6 <- evals %>%
  select(ID, score, age, gender)
```

Let's look at 5 random rows of data.

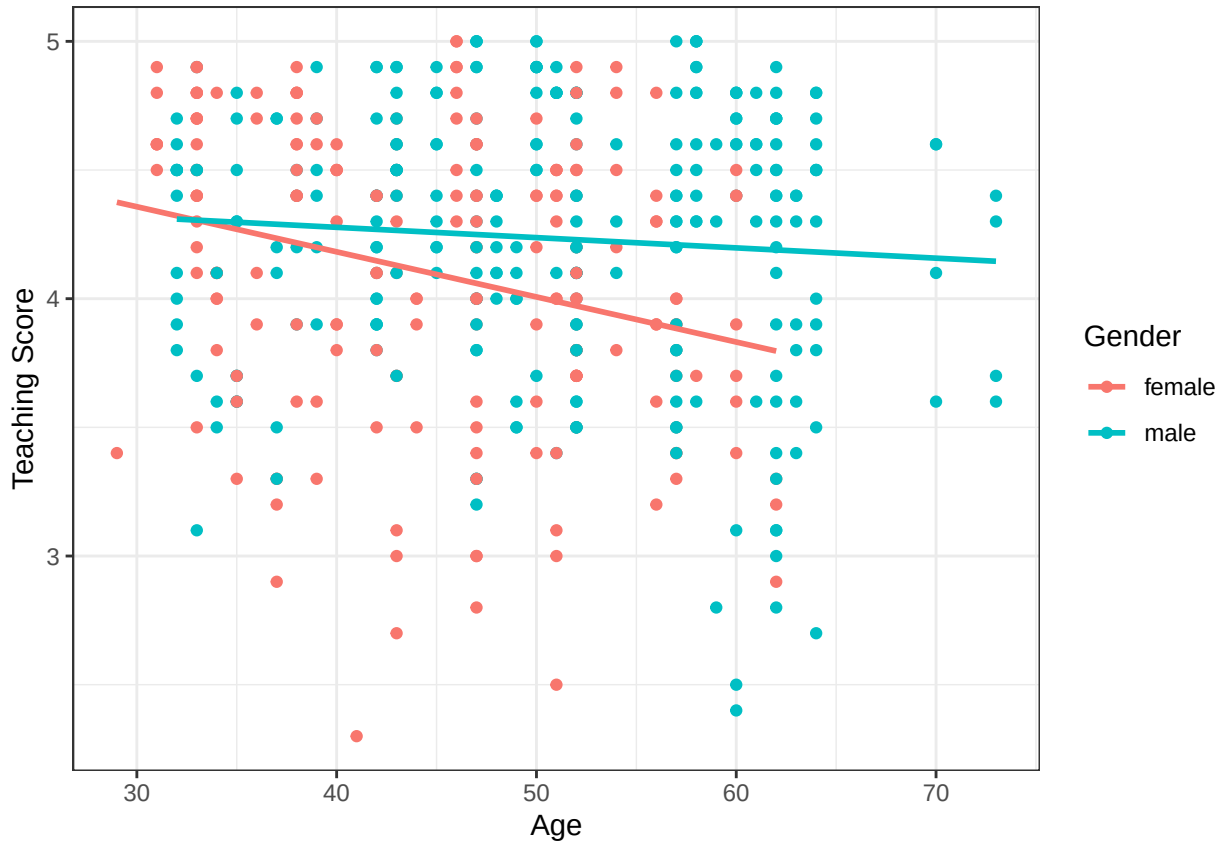
```
set.seed(941)
evals_ch6 %>%
  sample_n(size = 5)
```

```
## # A tibble: 5 x 4
##      ID score   age gender
##   <int> <dbl> <int> <fct>
## 1    61   3.7    35 male
## 2    15   3.9    40 female
## 3   309   3.6    35 male
## 4   274   4.2    57 male
## 5   256   4.1    52 male
```

- Interaction Model

In this model we allow a different slope and intercept for each gender.

```
ggplot(evals_ch6, aes(x = age, y = score, color = gender)) +
  geom_point() +
  # I suggest adding jitter (geom_point(position=))
  labs(x = "Age", y = "Teaching Score", color = "Gender") +
  geom_smooth(method = "lm", se = FALSE) +
  theme_bw()
```



To get the model in R, we use the * which is not multiplication but an interaction term in the model formula.

```
# Fit regression model:
score_model_interaction <- lm(score ~ age * gender, data = evals_ch6)
```

```
# Get regression table:
get_regression_table(score_model_interaction)
```

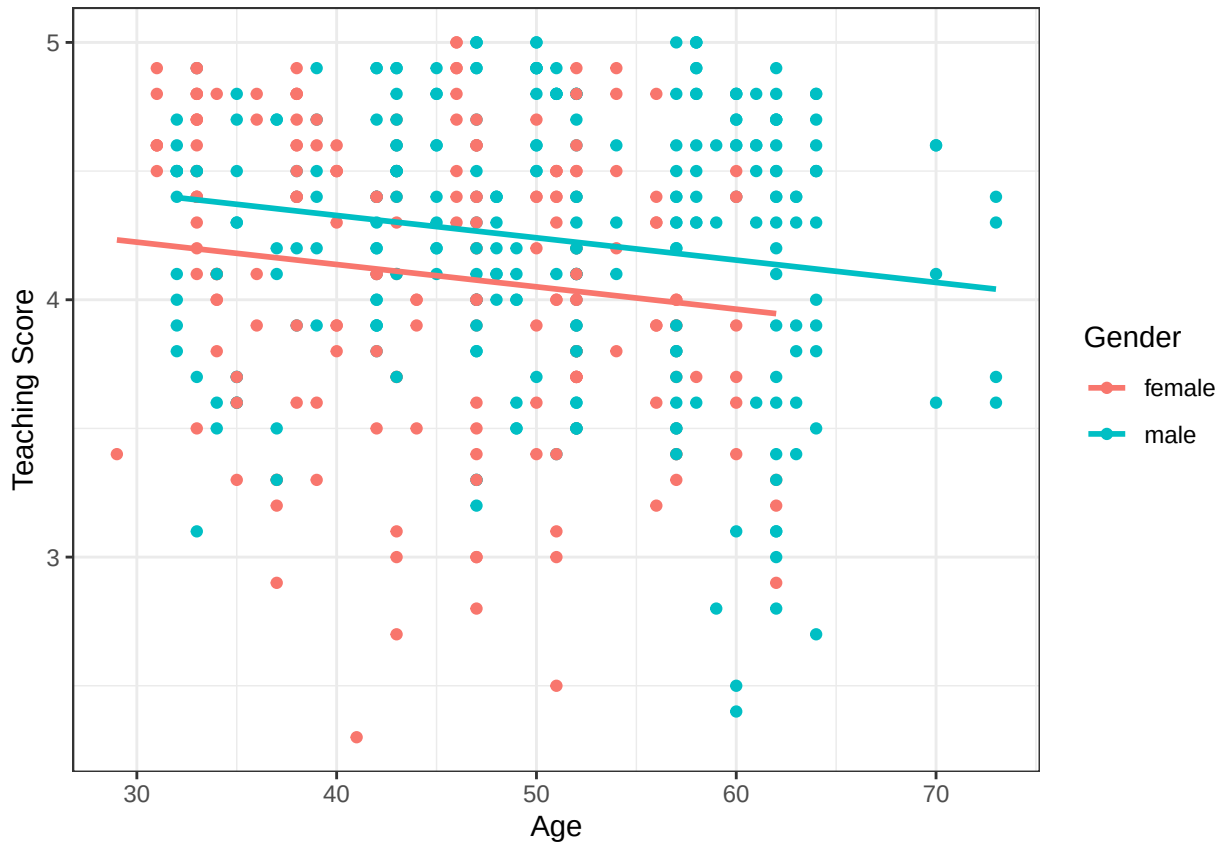
```
## # A tibble: 4 x 7
##   term           estimate std_error statistic p_value lower_ci upper_ci
##   <chr>             <dbl>     <dbl>     <dbl>   <dbl>    <dbl>    <dbl>
## 1 intercept          4.88      0.205     23.8    0         4.48     5.29
## 2 age               -0.018     0.004     -3.92   0        -0.026   -0.009
## 3 gender: male      -0.446     0.265     -1.68   0.094    -0.968    0.076
## 4 age:gendermale     0.014     0.006      2.45   0.015     0.003    0.024
```

- Parallel Slopes Model

We will use the same data, but just build a different model.

```
ggplot(evals_ch6, aes(x = age, y = score, color = gender)) +
  geom_point() +
```

```
  labs(x = "Age", y = "Teaching Score", color = "Gender") +
  geom_parallel_slopes(se = FALSE) +
  theme_bw()
```



- Notice that the line for females stops at the extremes of the observed data.

```
# Fit regression model:
score_model_parallel_slopes <- lm(score ~ age + gender, data = evals_ch6)
```

```
# Get regression table:
get_regression_table(score_model_parallel_slopes)
```

```
## # A tibble: 3 x 7
##   term          estimate std_error statistic p_value lower_ci upper_ci
##   <chr>            <dbl>     <dbl>     <dbl>   <dbl>    <dbl>    <dbl>
## 1 intercept        4.48      0.125     35.8    0         4.24     4.73
## 2 age             -0.009     0.003     -3.28   0.001    -0.014   -0.003
## 3 gender: male     0.191     0.052      3.63   0         0.087    0.294
```

Now let's use R-squared to pick the model. We will use `rbind()` and `tidyverse` commands to put the results in a nice form.

```
get_regression_summaries(score_model_interaction) %>%
  rbind(get_regression_summaries(score_model_parallel_slopes)) %>%
  mutate(model=c("Interaction","Parallel Slopes")) %>%
  select(model,r_squared,adj_r_squared)
```

```
## # A tibble: 2 x 3
##   model           r_squared adj_r_squared
##   <chr>               <dbl>         <dbl>
## 1 Interaction         0.051         0.045
## 2 Parallel Slopes     0.039         0.035
```

Neither model is great since the R-squared is so small; we are only explaining 5 percent of the variation in the teacher score with our model. However, the more complex interaction model is better than the parallel slope model.

Simpson's paradox

It is key in modeling to account for lurking variables. This site Simpson's paradox.

Here is a nice example from the `palmerpenguins` data package.

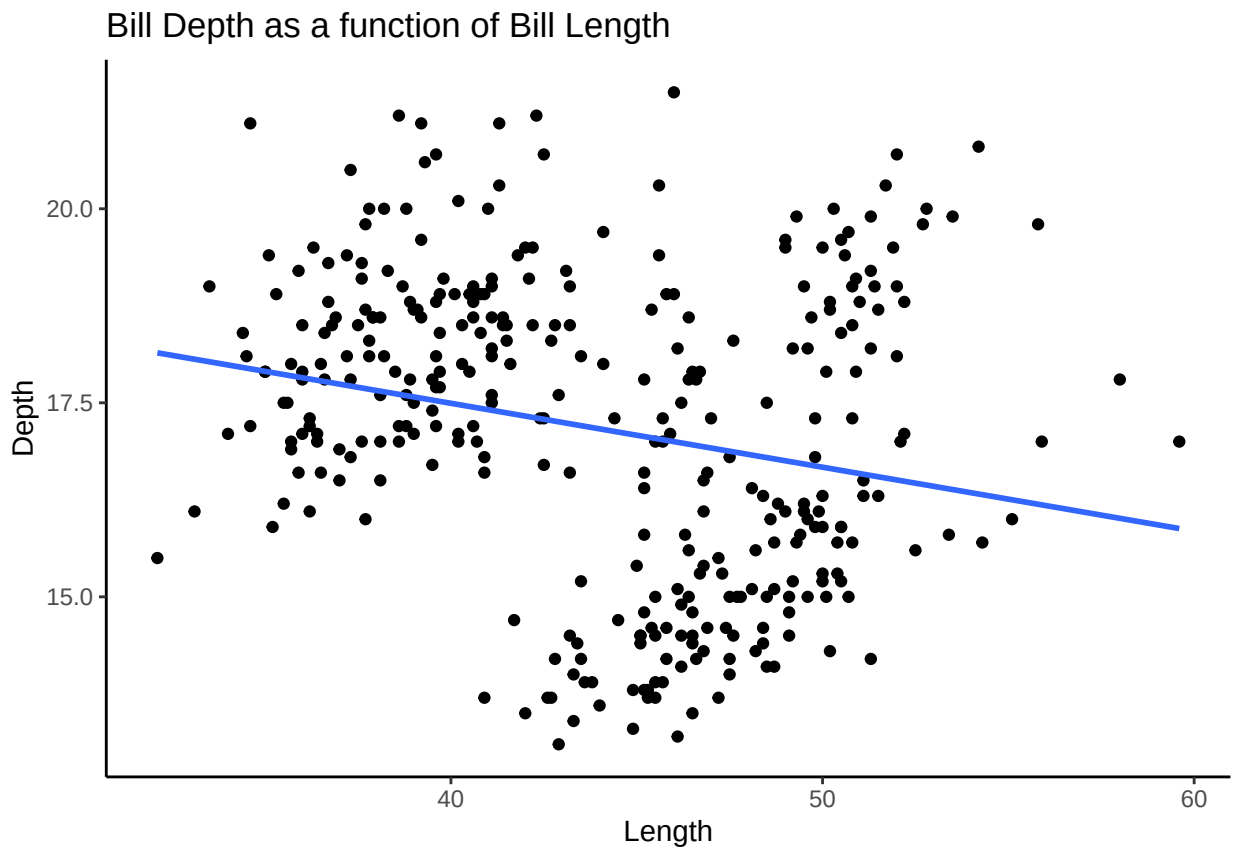
```
library(palmerpenguins)
```

```
penguin_df<-
  palmerpenguins::penguins %>%
  na.omit()
```

```
head(penguin_df)
```

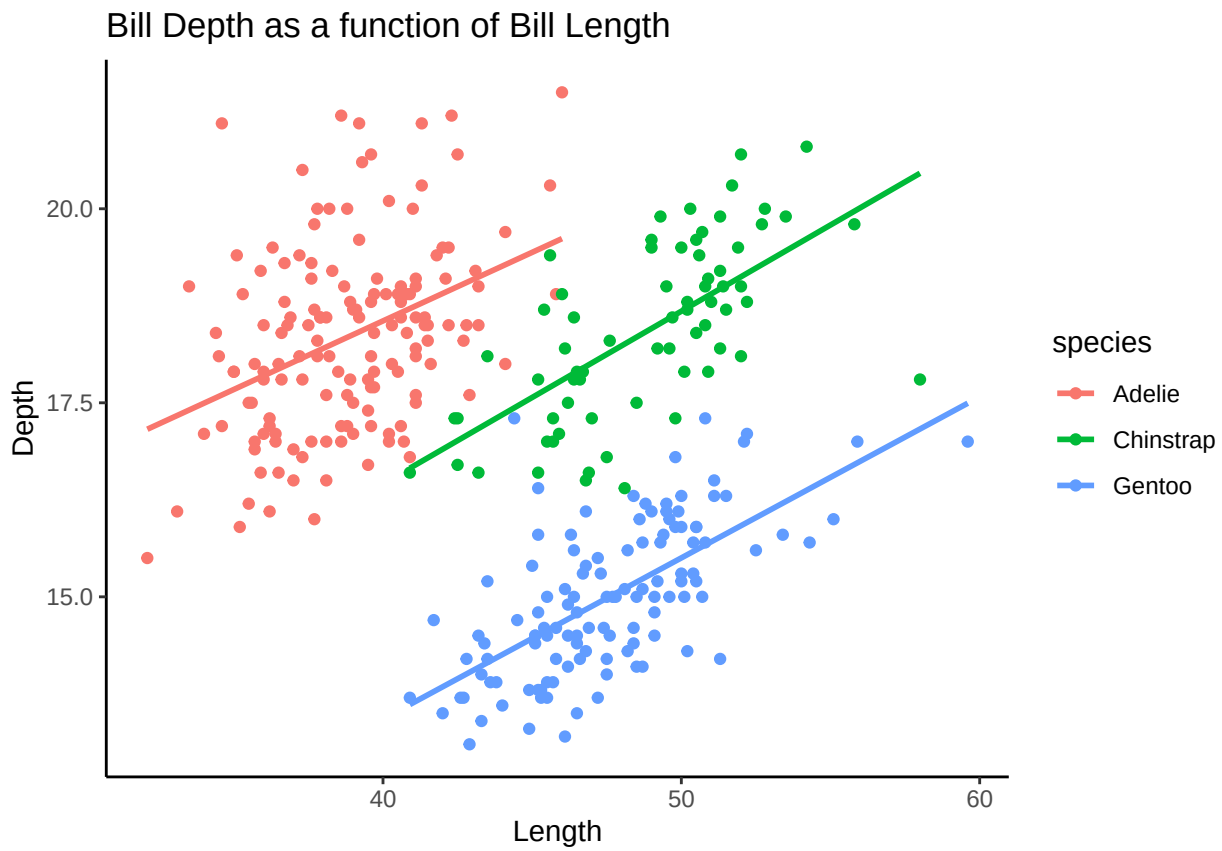
```
## # A tibble: 6 x 8
##   species island bill_length_mm bill_depth_mm flipper_length_~ body_mass_g sex
##   <fct>   <fct>           <dbl>         <dbl>            <int>       <int> <fct>
## 1 Adelie  Torge~           39.1          18.7              181        3750 male
## 2 Adelie  Torge~           39.5          17.4              186        3800 fema~
## 3 Adelie  Torge~           40.3          18                195        3250 fema~
## 4 Adelie  Torge~           36.7          19.3              193        3450 fema~
## 5 Adelie  Torge~           39.3          20.6              190        3650 male
## 6 Adelie  Torge~           38.9          17.8              181        3625 fema~
## # ... with 1 more variable: year <int>
```

```
penguin_df %>%
  ggplot(aes(x=bill_length_mm, y=bill_depth_mm)) +
  geom_point() +
  labs(x="Length", y="Depth", title="Bill Depth as a function of Bill Length") +
  theme_classic() +
  geom_smooth(method = "lm", se = FALSE)
```



From this we might as well conclude that the longer the bill, the less deep it is. However, if you drill down from the population level to the species level we see the opposite result.

```
penguin_df %>%
  ggplot(aes(x=bill_length_mm, y=bill_depth_mm,
             color=species)) +
  geom_point() +
  labs(x="Length", y="Depth", title="Bill Depth as a function of Bill Length") +
  theme_classic() +
  geom_smooth(method = "lm", se = FALSE)
```



Explain these results in terms of a confounding variable.

Documenting software

- File creation date: 2022-06-27
- R version 4.1.3 (2022-03-10)
- `tidyverse` package version: 1.3.1
- `skimr` package version: 2.1.4
- `palmerpenguins` package version: 0.1.0

- `moderndive` package version: 0.5.4