Model parameter inference and model selection

steven tjt

1 Confidence intervals for regression parameters

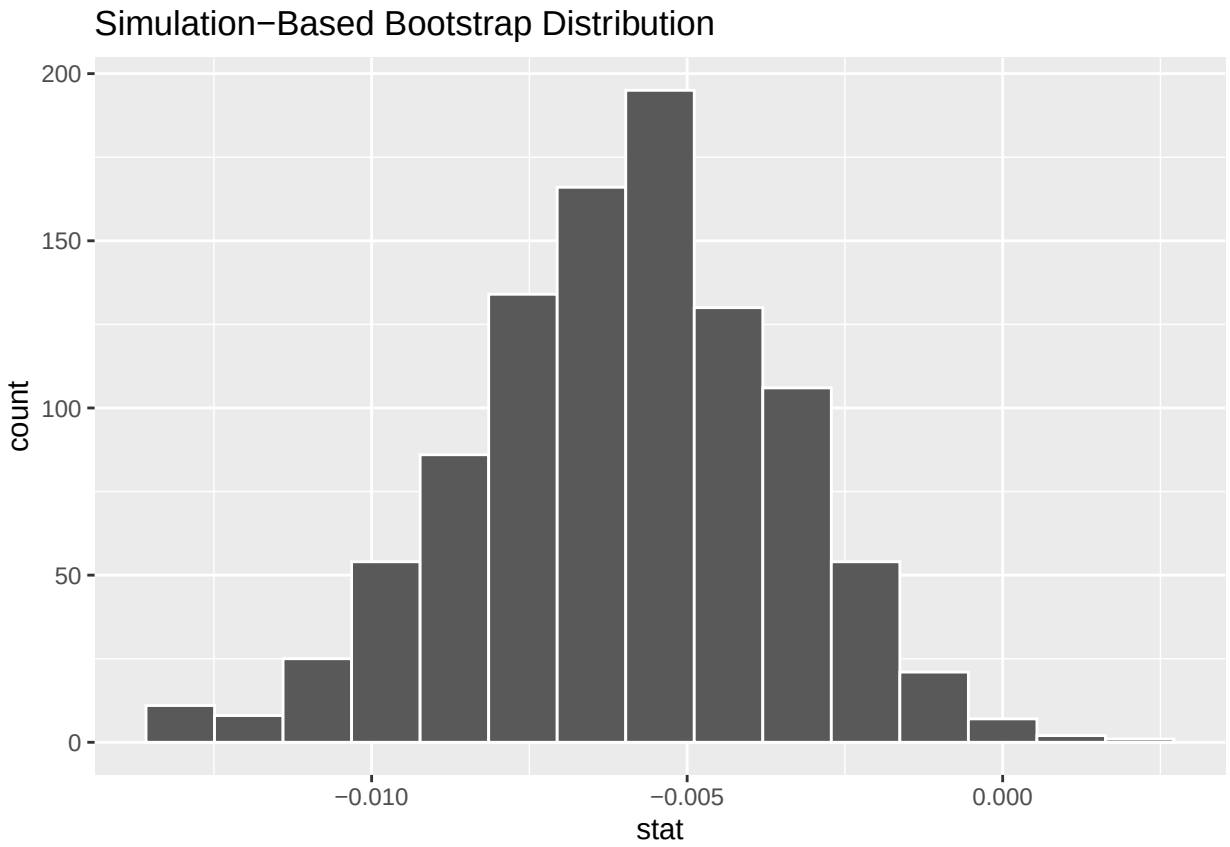
1.1 Bootstrap Confidence Intervals for the slope in Simple Linear Regression(SLR)

```
slr.model <- lm(score ~ age, data = evals)
coeff<-slr.model %>%
  coef()

coeff
```

```
 (Intercept)          age
 4.461932354 -0.005938225
```

The point estimate of the slope parameter here is hat = -0.006. The following code estimates the sampling distribution of b via the bootstrap method.

```
bootstrap_beta_distn <- evals %>%
  specify(score ~ age) %>%
  generate(reps = 1000, type = "bootstrap") %>%
  calculate(stat = "slope")

bootstrap_beta_distn %>%
  visualize()
```



Now we can use the get_ci function to calculate a 95% confidence interval. We can do this in two different ways. Remember that these denote a range of plausible values for an unknown true population slope parameter regressing teaching score on age.

```
percentile_beta_ci <- bootstrap_beta_distn %>%
  get_ci(level = 0.95, type = "percentile")
```

```
se_beta_ci <- bootstrap_beta_distn %>%
  get_ci(level = 0.95, type = "se", point_estimate = coeff[2])
```

Using the 2.5% and 97.5% percentiles of the simulated bootstrap sampling distribution the 95% confidence interval is (-0.011, -0.001) and the 95% confidence interval using the standard deviation of the sampling distribution (i.e. estimated standard error of b) is (-0.011, -0.001). With the bootstrap distribution being close to symmetric, it makes sense that the two resulting confidence intervals are similar.

1.2 Confidence Intervals for the parameters in Multiple Regression

Let's continue with the teaching evaluations data by fitting the multiple regression model with one numerical and one categorical explanatory variable that we first saw in Week 7. In this model: * y: response variable of instructor evaluation score * x1: numerical explanatory variable of age x2: categorical explanatory variable of gender

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

evals_multiple
```

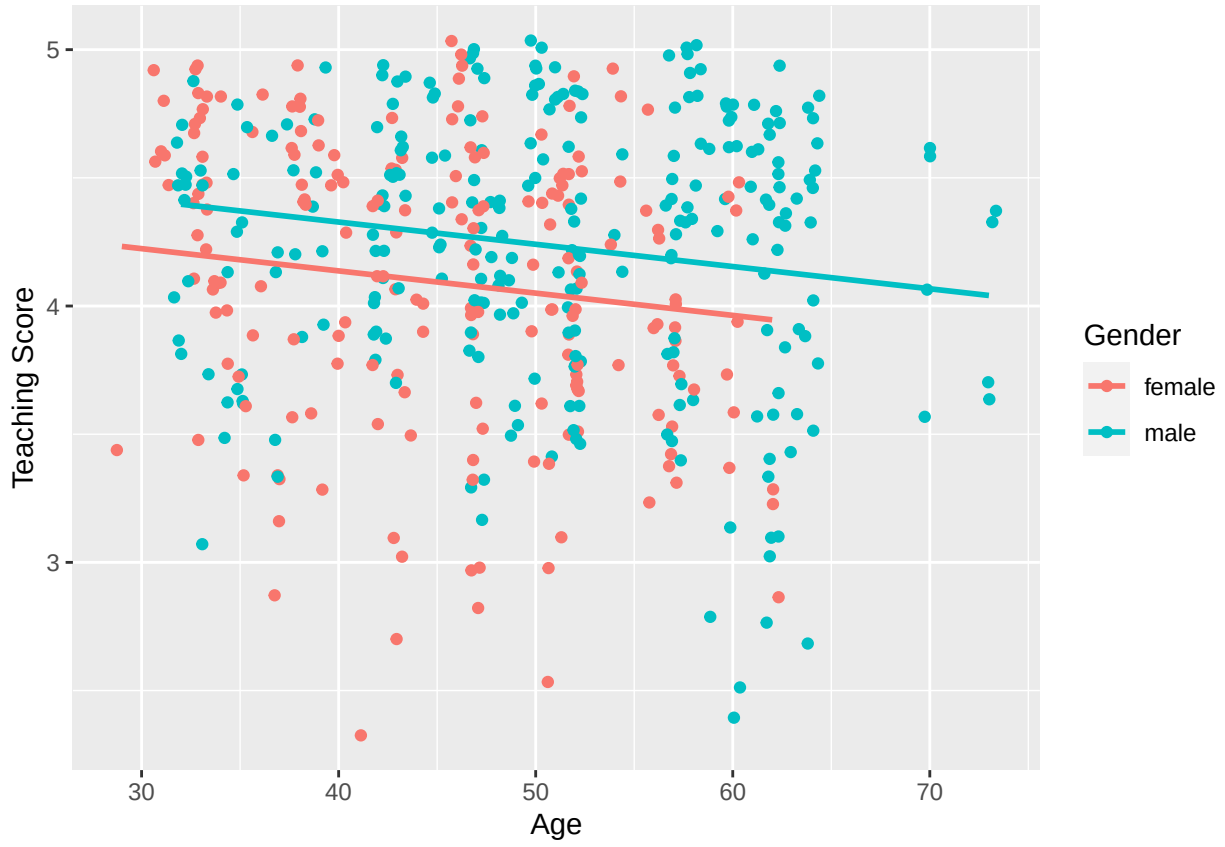
```
# A tibble: 463 x 3
   score gender   age
   <dbl> <fct>  <int>
 1   4.7 female    36
 2   4.1 female    36
 3   3.9 female    36
 4   4.8 female    36
 5   4.6 male      59
 6   4.3 male      59
 7   2.8 male      59
 8   4.1 male      51
 9   3.4 male      51
10   4.5 female    40
# ... with 453 more rows
```

1.Model 1: Parallel lines model (no interaction term) - both male and female professors have the same slope describing the associated effect of age on teaching score

```
ggplot(evals_multiple, aes(x = age, y = score, color = gender)) +
  geom_jitter() +
  labs(x = "Age", y = "Teaching Score", color = "Gender") +
  geom_smooth(method = "lm", se = FALSE)
```

Model 2: Interaction model - allowing for male and female professors to have different slopes describing the associated effect of age on teaching score

```r
# Now, let's superimpose our parallel regression lines onto the scatterplot of teaching score against a
coeff <- par.model %>%
  coef() %>%
  as.numeric()

slopes <- evals_multiple %>%
  group_by(gender) %>%
  summarise(min = min(age), max = max(age)) %>%
  mutate(intercept = coeff[1]) %>%
  mutate(intercept = ifelse(gender == "male", intercept + coeff[3], intercept)) %>%
  gather(point, age, -c(gender, intercept)) %>%
  mutate(y_hat = intercept + age * coeff[2])

ggplot(evals_multiple, aes(x = age, y = score, col = gender)) +
  geom_jitter() +
  labs(x = "Age", y = "Teaching Score", color = "Gender") +
  geom_line(data = slopes, aes(y = y_hat), size = 1)
```



Refresher: Regression tables

Let's also recall the regression models. First, the regression model with no interaction effect: note the use of + in the formula.

```r
par.model <- lm(score ~ age + gender, data = evals_multiple)

get_regression_table(par.model) %>%
```

```
  knitr::kable(
                digits = 3,
                caption = "Model 1: Regression model with no interaction effect included.",
                booktabs = TRUE
    )
```

Table 1: Model 1: Regression model with no interaction effect included.

term	estimate	std_error	statistic	p_value	lower_ci	upper_ci
intercept	4.484	0.125	35.792	0.000	4.238	4.730
age	-0.009	0.003	-3.280	0.001	-0.014	-0.003
gender: male	0.191	0.052	3.632	0.000	0.087	0.294

Second, the regression model with an interaction effect: note the use of * in the formula.

```
par.model <- lm(score ~ age*gender, data = evals_multiple)

get_regression_table(par.model) %>%
  knitr::kable(
                digits = 3,
                caption = "Model 2: Regression model with interaction effect included.",
                booktabs = TRUE
    )
```

Table 2: Model 2: Regression model with interaction effect included.

term	estimate	std_error	statistic	p_value	lower_ci	upper_ci
intercept	4.883	0.205	23.795	0.000	4.480	5.286
age	-0.018	0.004	-3.919	0.000	-0.026	-0.009
gender: male	-0.446	0.265	-1.681	0.094	-0.968	0.076
age:gendermale	0.014	0.006	2.446	0.015	0.003	0.024

2 Inference using Confidence Intervals

Let's use the confidence interval based on theoretical results for the slope parameter in the SLR model applied to the teacher evaluation scores with age as the the single explanatory variable and the instructors' evaluation scores as the outcome variable.

```
get_regression_table(slr.model) %>%
  knitr::kable(
                digits = 3,
                caption = "Estimates from the SLR model of `score` on `age`.",
                booktabs = TRUE
              )
```

Table 3: Estimates from the SLR model of `score` on `age`.

term	estimate	std_error	statistic	p_value	lower_ci	upper_ci
intercept	4.462	0.127	35.195	0.000	4.213	4.711
age	-0.006	0.003	-2.311	0.021	-0.011	-0.001

2.1 Multiple regression

Consider, again, the fitted interaction model for score with age and gender as the two explanatory variables.

```
int.model <- lm(score ~ age * gender, data = evals_multiple)
get_regression_table(int.model)
```

```
# 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
```

3 Variable selection using confidence intervals

Recall that as well as age and gender, there is also a potential explanatory variable bty_avg in the evals data, i.e. the numerical variable of the average beauty score from a panel of six students' scores between 1 and 10. We can fit the multiple regression model with the two continuous explanatory variables age and bty_avg as follows:

```
mlr.model <- lm(score ~ age + bty_avg, data = evals)
mlr.model
```

```
Call:
lm(formula = score ~ age + bty_avg, data = evals)

Coefficients:
(Intercept)          age      bty_avg
   4.054732    -0.003059     0.060656
```

4 Model comparisons using objective criteria

To illustrate this, let's return to the evals data and the MLR on the teaching evaluation score score with the two continuous explanatory variables age and bty_avg and compare this with the SLR model with just bty_avg. To access these measures for model comparisons we can use the glance function in the broom package

```
model.comp.values.slr.age <- glance(lm(score ~ age, data = evals))
model.comp.values.slr.age
```

```
# A tibble: 1 x 12
  r.squared adj.r.squared sigma statistic p.value    df logLik   AIC   BIC
      <dbl>         <dbl> <dbl>     <dbl>   <dbl> <dbl>  <dbl> <dbl> <dbl>
1    0.0115       0.00931 0.541      5.34  0.0213     1  -372.  750.  762.
# ... with 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

```
model.comp.values.slr.bty_avg <- glance(lm(score ~ bty_avg, data = evals))
model.comp.values.slr.bty_avg
```

```
# A tibble: 1 x 12
  r.squared adj.r.squared sigma statistic   p.value    df logLik   AIC   BIC
      <dbl>         <dbl> <dbl>     <dbl>     <dbl> <dbl>  <dbl> <dbl> <dbl>
1    0.0350        0.0329 0.535      16.7 0.0000508     1  -366.  738.  751.
# ... with 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

```
model.comp.values.mlr <- glance(lm(score ~ age + bty_avg, data = evals))
model.comp.values.mlr
```

```
# A tibble: 1 x 12
  r.squared adj.r.squared sigma statistic  p.value    df logLik   AIC   BIC
      <dbl>         <dbl> <dbl>     <dbl>    <dbl> <dbl>  <dbl> <dbl> <dbl>
1    0.0378        0.0336 0.535      9.03 0.000142     2  -366.  739.  756.
# ... with 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

Note that R^2_adj , AIC and BIC are contained in columns 2, 9 and 10 respectively. To access just these values and combine them in a single table we use:

```
Models <- c('SLR(age)','SLR(bty_avg)','MLR')
bind_rows(model.comp.values.slr.age, model.comp.values.slr.bty_avg,
          model.comp.values.mlr, .id = "Model") %>%
          select(Model, adj.r.squared, AIC, BIC) %>%
          mutate(Model = Models) %>%
          knitr::kable(
                  digits = 2,
                  caption = "Model comparison values for different models.",
              )
```

Table 4: Model comparison values for different models.

Model	adj.r.squared	AIC	BIC
SLR(age)	0.01	749.62	762.03
SLR(bty_avg)	0.03	738.44	750.86
MLR	0.03	739.12	755.67