Math 300 Lesson 36 Notes

Regression Inference via Simulation

YOUR NAME HERE

July, 2022

Contents

Objectives

1. Construct and interpret bootstrap confidence intervals, both percentile and standard error method, for regression parameters such as slope and correlation coefficient.
2. Conduct and interpret a permutation test on the regression slope parameter.

Reading

Chapter 10.4

Lesson

Work through the learning check LC 10.2.

- We don't have to worry about normality and equality of variance for this lesson. Those assumptions are only required to conduct inference using theory-based methods. Simulation-based methods do not require these assumptions.
- The confidence intervals are similar to what we have done before. The permutation test requires careful thought about the procedure.

Libraries

```
library(tidyverse)
library(infer)
library(moderndive)
```

Review

Let's review the permutation test from the reading.

Data and model

Let's get the data and model again.

```
evals_ch5 <- evals %>%
  select(ID, score, bty_avg, age)
glimpse(evals_ch5)
```

```
## Rows: 463
## Columns: 4
## $ ID      <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18,~
## $ score   <dbl> 4.7, 4.1, 3.9, 4.8, 4.6, 4.3, 2.8, 4.1, 3.4, 4.5, 3.8, 4.5, 4.~
## $ bty_avg <dbl> 5.000, 5.000, 5.000, 5.000, 3.000, 3.000, 3.000, 3.333, 3.333,~
## $ age     <int> 36, 36, 36, 36, 59, 59, 59, 51, 51, 40, 40, 40, 40, 40, 40, 40~
```

```
score_model <- lm(score ~ bty_avg, data = evals_ch5)
```

And the regression summary.

```
get_regression_table(score_model)
```

```
## # A tibble: 2 x 7
##   term      estimate std_error statistic p_value lower_ci upper_ci
##   <chr>        <dbl>     <dbl>     <dbl>   <dbl>    <dbl>    <dbl>
## 1 intercept    3.88      0.076     51.0        0    3.73     4.03
## 2 bty_avg      0.067     0.016      4.09       0    0.035    0.099
```

This table is for reference.

Permutation Test

The hypothesis test is

$$H_O : \beta_1 = 0$$

$$H_A : \beta_1 \neq 0$$

Under the null, there is no linear relationship between teaching scores and beauty scores. Thus under the null hypothesis, we could just repeatedly shuffle the beauty scores and calculate the slope. This would give us an estimate of the sampling distribution of the slope under the null hypothesis.

Here is the code from the `infer` package:

```
set.seed(50)
# Complete code
# null_distn_slope <- evals %>%
#   specify(______ ~ _______) %>%
#   hypothesize(null = "________") %>%
#   generate(reps = 1000, type = "_________") %>%
#   calculate(stat = "_________")
```

```
# Complete code
#visualize(__________________)
```

The distribution is centered at zero since this is the assumed value under the null distribution.

To find the p-value, we need the observed slope.

```
# Complete code
# (observed_slope <- evals %>%
#   specify(_______ ~ _______) %>%
#   calculate(stat = "________"))
```

```
# Complete code
# visualize(_____________) +
#   shade_p_value(obs_stat = _____________, direction = "____________")
```

The p-value is going to be small.

```
# Complete code
# null_distn_slope %>%
#   get_p_value(obs_stat = ________________, direction = "_________________")
```

Based on the data we reject the null hypothesis of no linear relationship between beauty score and teaching score.

Learning Check 10.2 (Objectives 1 and 2)

(LC 10.2) Repeat the inference but this time for the correlation coefficient instead of the slope. Note the implementation of `stat = "correlation"` in the `calculate()` function of the `infer` package.

Confidence Intervals

```
# Complete code
set.seed(839)
# bootstrap_distn_corr <- evals_ch5 %>%
#   specify(formula = ___________ ~ __________) %>%
#   generate(reps = 1000, type = "___________") %>%
#   calculate(stat = "_______________")
```

```
# Complete code
# percentile_ci <- bootstrap_distn_corr %>%
#   get_confidence_interval(type = "_________________", level = _______________)
# percentile_ci
```

```
# Complete code
# visualize(_______________) +
#   theme_classic()
```

Can we use standard error method?

```
# Complete code
# (observed_corr <- evals %>%
#   specify(___________ ~ ______________) %>%
#   calculate(stat = "_____________"))
```

```
# Complete code
# (std_err_ci<-bootstrap_distn_corr %>%
#   get_confidence_interval(point_estimate = _________________, type = "_________", level = 0.95))
```

Conclusion:

Hypothesis Test

The hypothesis test is

$$H_O : \rho = 0$$

$$H_A : \rho \neq 0$$

```
# Complete code
set.seed(2001)
# null_distn_corr <- evals %>%
#   specify(___________ ~ ______________) %>%
#   hypothesize(null = "______________") %>%
#   generate(reps = 1000, type = "______________") %>%
#   calculate(stat = "_________________")
```

```
# Complete code
# visualize(_________________) +
#   shade_p_value(obs_stat = observed_corr, direction = "____________")
# Complete code
# _____________________ %>%
#   get_p_value(obs_stat = __________________, direction = "_____________")
```

Conclusion:

Documenting software

- File creation date: 2022-07-15
- R version 4.1.3 (2022-03-10)
- `tidyverse` package version: 1.3.1
- `moderndive` package version: 0.5.4
- `infer` package version: 1.0.2