AE 03: Bootstrap confidence intervals

Houses in Duke Forest

Dav King

Sep 12, 2022

```
library(tidyverse)
library(tidymodels)
library(openintro)
```

```
Warning: package 'openintro' was built under R version 4.1.3

Warning: package 'airports' was built under R version 4.1.3

Warning: package 'cherryblossom' was built under R version 4.1.3

Warning: package 'usdata' was built under R version 4.1.3
```

```
library(knitr)
```

Data

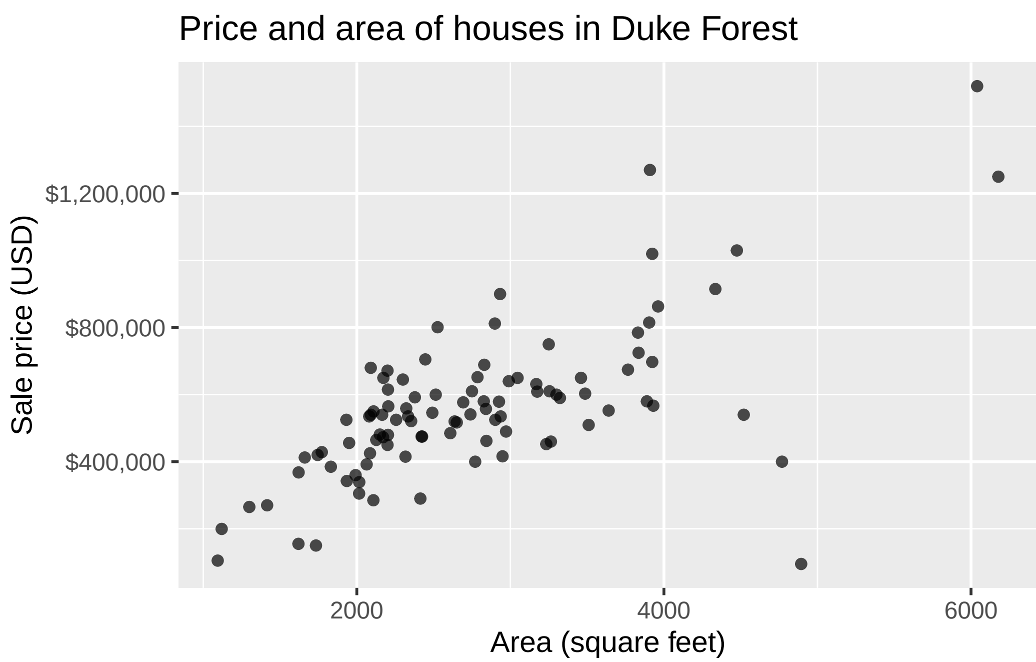
The data are on houses that were sold in the Duke Forest neighborhood of Durham, NC around November 2020. It was originally scraped from Zillow, and can be found in the `duke_forest` data set in the **openintro** R package.

```
glimpse(duke_forest)
```

```
Rows: 98
Columns: 13
$ address    <chr> "1 Learned Pl, Durham, NC 27705", "1616 Pinecrest Rd, Durha~
$ price      <dbl> 1520000, 1030000, 420000, 680000, 428500, 456000, 1270000, ~
$ bed        <dbl> 3, 5, 2, 4, 4, 3, 5, 4, 4, 3, 4, 4, 3, 5, 4, 5, 3, 4, 4, 3,~
$ bath       <dbl> 4.0, 4.0, 3.0, 3.0, 3.0, 3.0, 5.0, 3.0, 5.0, 2.0, 3.0, 3.0,~
$ area       <dbl> 6040, 4475, 1745, 2091, 1772, 1950, 3909, 2841, 3924, 2173,~
$ type       <chr> "Single Family", "Single Family", "Single Family", "Single ~
$ year_built <dbl> 1972, 1969, 1959, 1961, 2020, 2014, 1968, 1973, 1972, 1964,~
$ heating    <chr> "Other, Gas", "Forced air, Gas", "Forced air, Gas", "Heat p~
$ cooling    <fct> central, central, central, central, central, central, centr~
$ parking    <chr> "0 spaces", "Carport, Covered", "Garage - Attached, Covered~
$ lot        <dbl> 0.97, 1.38, 0.51, 0.84, 0.16, 0.45, 0.94, 0.79, 0.53, 0.73,~
$ hoa        <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
$ url        <chr> "https://www.zillow.com/homedetails/1-Learned-Pl-Durham-NC-~
```

Exploratory data analysis

```r
ggplot(duke_forest, aes(x = area, y = price)) +
  geom_point(alpha = 0.7) +
  labs(
    x = "Area (square feet)",
    y = "Sale price (USD)",
    title = "Price and area of houses in Duke Forest"
  ) +
  scale_y_continuous(labels = label_dollar())
```



Model

```
df_fit <- linear_reg() |>
  set_engine("lm") |>
  fit(price ~ area, data = duke_forest)

tidy(df_fit) |>
  kable(digits = 2)
```

term	estimate	std.error	statistic	p.value
(Intercept)	116652.33	53302.46	2.19	0.03
area	159.48	18.17	8.78	0.00

Bootstrap confidence interval

1. Calculate the observed fit (slope)

```
observed_fit <- duke_forest |>
  specify(price ~ area) |>
  fit()

observed_fit
```

```
# A tibble: 2 x 2
  term      estimate
  <chr>        <dbl>
1 intercept  116652.
2 area          159.
```

2 Take *n* bootstrap samples and fit models to each one.

Fill in the code, then set `eval: true` .

```
n = 100
set.seed(091222)

boot_fits <- duke_forest |>
  specify(price ~ area) |>
  generate(reps = n, type = "bootstrap") |>
  fit()

boot_fits
```

```
# A tibble: 200 x 3
# Groups:   replicate [100]
   replicate term      estimate
       <int> <chr>        <dbl>
 1         1 intercept  144850.
 2         1 area          149.
 3         2 intercept  187775.
 4         2 area          129.
 5         3 intercept  183626.
 6         3 area          135.
```

```
 7          4 intercept  135876.
 8          4 area          146.
 9          5 intercept   84386.
10          5 area          176.
# ... with 190 more rows
```

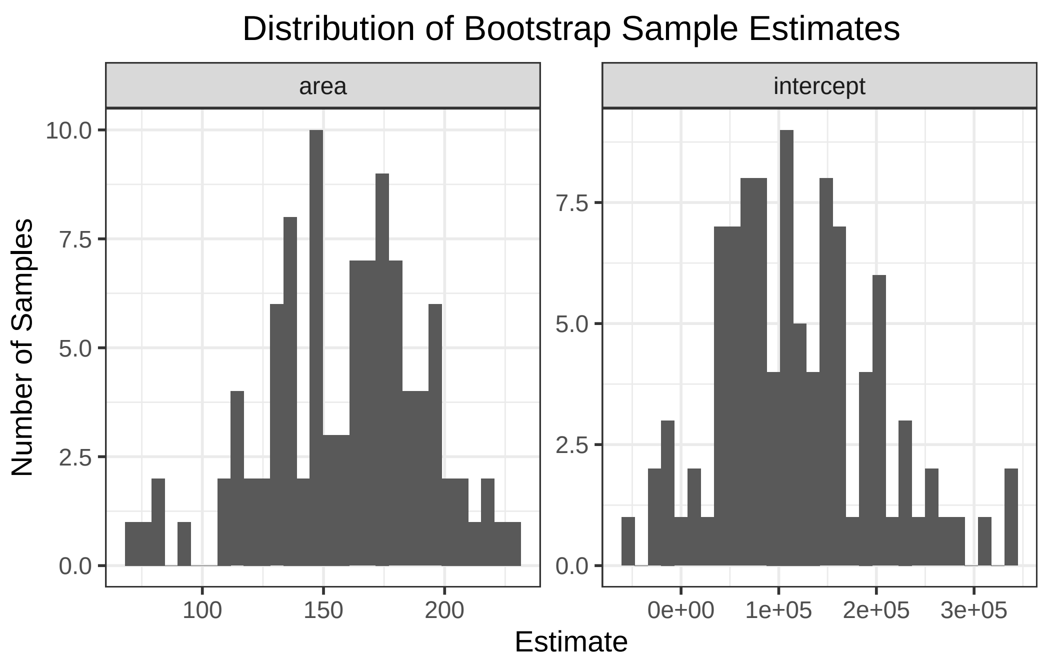
- Why do we set a seed before taking the bootstrap samples?

We set a seed so that the "random" values that R is taking here are replicated the exact same way every time that the .qmd file is run. In other words, we have a fixed set of random numbers, and the seed means that we will generate the exact same bootstrap samples every single time.

- Make a histogram of the bootstrap samples to visualize the bootstrap distribution.

```
boot_fits %>%
  group_by(replicate) %>%
  ggplot(aes(x = estimate)) +
  geom_histogram() +
  facet_wrap(~ term, scales = "free") +
  theme_bw() +
  labs(x = "Estimate", y = "Number of Samples",
       title = "Distribution of Bootstrap Sample Estimates") +
  theme(plot.title = element_text(hjust = 0.5))
```

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



3 Compute the 95% confidence interval as the middle 95% of the bootstrap distribution

Fill in the code, then set `eval: true` .

```
get_confidence_interval(
  boot_fits,
  point_estimate = observed_fit,
  level = .95,
  type = "percentile"
)
```

```
# A tibble: 2 x 3
  term      lower_ci upper_ci
  <chr>        <dbl>    <dbl>
1 area          83.1     218.
2 intercept -18031.   298414.
```

Changing confidence level

Modify the code from Step 3 to create a 90% confidence interval.

```
get_confidence_interval(
  boot_fits,
  point_estimate = observed_fit,
  level = .90,
  type = "percentile"
)
```

```
# A tibble: 2 x 3
  term      lower_ci upper_ci
  <chr>        <dbl>    <dbl>
1 area          107.     206.
2 intercept   -8665.  261168.
```

Modify the code from Step 3 to create a 99% confidence interval.

```
get_confidence_interval(
  boot_fits,
  point_estimate = observed_fit,
  level = .99,
  type = "percentile"
)
```

```
# A tibble: 2 x 3
  term      lower_ci upper_ci
  <chr>        <dbl>    <dbl>
1 area          73.1     225.
2 intercept -45140.   335041.
```

- Which confidence level produces the most accurate confidence interval (90%, 95%, 99%)? Explain

The most accurate confidence interval is at 99% - we are the most confident that the 99% interval contains our actual population parameter compared to the other two.

- Which confidence level produces the most precise confidence interval (90%, 95%, 99%)? Explain

The most precise confidence interval is the 90% confidence interval. At 90%, we have the most narrow range for our interval - there's a higher chance that the actual population parameter falls outside of this interval compared to the others, but this gives us a more precise estimate compared to the others because it holds a less inclusive range.

- If we want to be very certain that we capture the population parameter, should we use a wider or a narrower interval? What drawbacks are associated with using a wider interval?

We want to use a wider interval (a higher confidence level) in order to be more sure that we will capture the true population parameter. The wider the interval around our point estimate, the more likely we are to calculate an interval that contains the actual population parameter. However, this has a major drawback - with increased accuracy comes lower precision. The wider our interval gets, the less accurate of an estimate we have for what our population parameter actually is - even if we're more confident that it's contained within the interval in the first place.

- If we want to be very certain that we capture the population parameter, should we use a wider or a narrower interval? What drawbacks are associated with using a wider interval?

This is a duplicate question :)

! Important

To submit the AE:

- Render the document to produce the PDF with all of your work from today's class.
- Push all your work to your `ae-03-` repo on GitHub. (You do not submit AEs on Gradescope).