Hypothesis Testing

Camille Bergeron

2023-01-25

Redlined or Not Redlined

```
# reading in
redlined_tracts <- read_xlsx("RedlinedTracts2020.xlsx")

redlined_tracts <- redlined_tracts %>%
  rename(ct_red = `% of CT within Redlined Zone`) %>%
  mutate(prop50 = ifelse(ct_red > 50, "rl", "not")) %>%
  mutate(prop75 = ifelse(ct_red > 75, "rl", "not")) %>%
  mutate(prop90 = ifelse(ct_red > 90, "rl", "not"))

# ownership var
VARS <- tidycensus::load_variables(dataset = 'acs5', year = 2021, cache = T)

vars_tenure <- VARS %>%
  filter(grepl("B25003", name))

census_vars <- tidycensus::get_acs(geography = "tract",
                                   variable = c("B25003_001", "B25003_002"),
                                   output = "wide",
                                   state = "MA",
                                   county = "Suffolk",
                                   geometry = TRUE,
                                   year = 2021,
                                   cache_table = T,
                                   show_call = TRUE)
```

```
## Getting data from the 2017-2021 5-year ACS
```

```
## Downloading feature geometry from the Census website.  To cache shapefiles for use in future sessions
```

```
## Census API call: https://api.census.gov/data/2021/acs/acs5?get=B25003_001E%2CB25003_001M%2CB25003_00
```

```
## Warning: `funs()` was deprecated in dplyr 0.8.0.
## Please use a list of either functions or lambdas:
##
##   # Simple named list:
##   list(mean = mean, median = median)
##
##   # Auto named with `tibble::lst()`:
##   tibble::lst(mean, median)
##
##   # Using lambdas
```

```
##   list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))
```

```
##   |                                                                              |
```

```
tracts_ownership <- redlined_tracts %>%
  left_join(y = census_vars %>%
              select(!ends_with("M")) %>%
              rename(total_housing = B25003_001E) %>%
              rename(owner_occ = B25003_002E) %>%
              mutate(ownership_rate = owner_occ / total_housing) %>%
              select(GEOID, ownership_rate), by = c("GEOID20" = "GEOID"))
```

At 50% threshold

```
# hypothesis testing

# h0: m(nonredlined) - m(redlined) = 0
# hA: m(nonredlined) - m(redlined) > 0

t.test(tracts_ownership %>% filter(prop50 == "rl") %>% select(ownership_rate),
       tracts_ownership %>% filter(prop50 != "rl") %>% select(ownership_rate))
```

```
##
##  Welch Two Sample t-test
##
## data:  tracts_ownership %>% filter(prop50 == "rl") %>% select(ownership_rate) and tracts_ownership %
## t = -1.3723, df = 98.757, p-value = 0.1731
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.12383927  0.02257832
## sample estimates:
## mean of x mean of y
## 0.2894297 0.3400602
```

At 75% threshold

```
# hypothesis testing

# h0: m(nonredlined) - m(redlined) = 0
# hA: m(nonredlined) - m(redlined) > 0

t.test(tracts_ownership %>% filter(prop75 == "rl") %>% select(ownership_rate),
       tracts_ownership %>% filter(prop75 != "rl") %>% select(ownership_rate))
```

```
##
##  Welch Two Sample t-test
##
## data:  tracts_ownership %>% filter(prop75 == "rl") %>% select(ownership_rate) and tracts_ownership %
## t = -1.3456, df = 91.267, p-value = 0.1817
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.12357904  0.02376112
## sample estimates:
## mean of x mean of y
## 0.2825047 0.3324137
```

At 90% threshold

```
# hypothesis testing

# h0: m(nonredlined) - m(redlined) = 0
# hA: m(nonredlined) - m(redlined) > 0

t.test(tracts_ownership %>% filter(prop90 == "rl") %>% select(ownership_rate),
       tracts_ownership %>% filter(prop90 != "rl") %>% select(ownership_rate))
```

```
##
##  Welch Two Sample t-test
##
## data:  tracts_ownership %>% filter(prop90 == "rl") %>% select(ownership_rate) and tracts_ownership %
## t = -1.4157, df = 60.892, p-value = 0.1619
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.13520267  0.02311625
## sample estimates:
## mean of x mean of y
## 0.2736736 0.3297168
```