Statistics 506- Problem Set #6

Garrett Pinkston

Link to GitHub

github: https://github.com/garrettpinkston2015/Computational-Methods

Stratified Bootstrapping

If a sample has a categorical variable with small groups, bootstrapping can be tricky. Consider a situation where $n = 100$, but there is some categorical variable g where $g = 1$ has only 2 observations.

In a single bootstrap resample of that data, there is a

$$\binom{98}{100} \approx 13\%$$

chance that the bootstrap sample does not include either observation from $g = 1$. This implies that if we are attempting to obtain a bootstrap estimate in group $g = 1$, 13% of the bootstrapped samples will have no observations from that group and thus will be unable to produce an estimate.

A way around this is to carry out stratified bootstrap: Instead of taking a sample with replacement of the whole sample, take separate bootstrap resamples within each strata, then combine those resamples to generate the bootstrap sample.

```
library(DBI)
library(RSQLite)

# Specify the path to your SQLite database
db_path <- "/Users/garrettpinkston/Desktop/Michigan/STAT506/Data/lahman_1871-2022.sqlite"

# Connect to the database
lahman <- dbConnect(RSQLite::SQLite(), dbname = db_path)
```

Use the "lahman" data that we first introduced in SQL. In the statistical analysis of baseball statistics, one metric used to measure a player's performance is their **Range Factor**:

$$RF = 3\frac{PO + A}{InnOuts}$$

Here, "PO" is putouts, "A" is assists, and "InnOuts" is the number of outs they were on the field for.

1. Calculate the average RF for each team in the `Fielding` table. Then, since we don't have a closed form for the standard deviation of this statistic, carry out a stratified bootstrap *by team* to estimate it. Do this out three ways:

    1. Without any parallel processing

```
library(dplyr)
```

```
Attaching package: 'dplyr'

The following objects are masked from 'package:stats':

    filter, lag

The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union
```

```
# connect to database
db_path <- "/Users/garrettpinkston/Desktop/Michigan/STAT506/Data/lahman_1871-2022.sqlite"

lahman <- dbConnect(RSQLite::SQLite(), dbname = db_path)

fielding <- dbReadTable(lahman, "Fielding")

fielding <- fielding %>%
  filter(!is.na(PO), !is.na(A), !is.na(InnOuts), InnOuts > 0) %>%
  mutate(RF = 3 * (PO + A) / InnOuts)

team_avg_rf <- fielding %>%
  group_by(teamID) %>%
  summarise(avg_RF = mean(RF, na.rm = TRUE))
```

```r
stratified_bootstrap <- function(data, strata_col, n_boot) {
  results <- vector("list", n_boot)

  for (i in seq_len(n_boot)) {
    boot_sample <- data %>%
      group_by_at(strata_col) %>%
      group_modify(~ .x[sample(nrow(.x), replace = TRUE), ]) %>%
      ungroup()

    boot_avg_rf <- boot_sample %>%
      group_by(teamID) %>%
      summarise(avg_RF = mean(RF, na.rm = TRUE))

    results[[i]] <- boot_avg_rf
  }

  bind_rows(results, .id = "bootstrap_iteration")
}

n_boot <- 1000
boot_results <- stratified_bootstrap(fielding, "teamID", n_boot)

boot_sd <- boot_results %>%
  group_by(teamID) %>%
  summarise(sd_RF = sd(avg_RF, na.rm = TRUE))

final_results <- team_avg_rf %>%
  left_join(boot_sd, by = "teamID")

print(final_results)
```

```
# A tibble: 140 x 3
   teamID avg_RF   sd_RF
   <chr>   <dbl>   <dbl>
 1 ALT     0.387 0.0477
 2 ANA     0.415 0.0157
 3 ARI     0.366 0.00882
 4 ATL     0.388 0.00619
 5 BAL     0.393 0.00573
 6 BFN     0.451 0.0198
 7 BFP     0.464 0.0532
```

```
 8 BL1     0.442 0.0289
 9 BL2     0.401 0.0153
10 BL3     0.444 0.0427
# i 130 more rows
```

```
dbDisconnect(lahman)
```

parallel

```
library(dplyr)
library(DBI)
library(RSQLite)
library(parallel)

# connect to database
db_path <- "/Users/garrettpinkston/Desktop/Michigan/STAT506/Data/lahman_1871-2022.sqlite"
lahman <- dbConnect(SQLite(), dbname = db_path)

fielding <- dbReadTable(lahman, "Fielding") %>%
  filter(!is.na(PO), !is.na(A), !is.na(InnOuts), InnOuts > 0) %>%
  mutate(RF = 3 * (PO + A) / InnOuts)

team_avg_rf <- fielding %>%
  group_by(teamID) %>%
  summarise(avg_RF = mean(RF, na.rm = TRUE))

bootstrap_iter <- function(data) {
  library(dplyr)
  data %>%
    group_by(teamID) %>%
    group_modify(~ .x[sample(nrow(.x), replace = TRUE), ]) %>%
    summarise(avg_RF = mean(RF, na.rm = TRUE))
}

parallel_bootstrap <- function(data, n_boot) {
  n_cores <- detectCores() - 1  # Use all but one core
  cl <- makeCluster(n_cores)

  clusterExport(cl, c("data", "bootstrap_iter"), envir = environment())

  clusterEvalQ(cl, library(dplyr))
```

```
  results <- parLapply(cl, 1:n_boot, function(i) bootstrap_iter(data))
  stopCluster(cl)

  bind_rows(results, .id = "bootstrap_iteration")
}

n_boot <- 1000
boot_results <- parallel_bootstrap(fielding, n_boot)

boot_sd <- boot_results %>%
  group_by(teamID) %>%
  summarise(sd_RF = sd(avg_RF, na.rm = TRUE))

final_results <- team_avg_rf %>%
  left_join(boot_sd, by = "teamID")

print(final_results)
```

```
# A tibble: 140 x 3
   teamID avg_RF   sd_RF
   <chr>   <dbl>   <dbl>
 1 ALT     0.387 0.0502
 2 ANA     0.415 0.0158
 3 ARI     0.366 0.00906
 4 ATL     0.388 0.00613
 5 BAL     0.393 0.00526
 6 BFN     0.451 0.0203
 7 BFP     0.464 0.0517
 8 BL1     0.442 0.0291
 9 BL2     0.401 0.0164
10 BL3     0.444 0.0412
# i 130 more rows
```

```
dbDisconnect(lahman)
```

futures

```
library(dplyr)
library(DBI)
library(RSQLite)
library(future)
```

```r
library(furrr)

# connect to database
db_path <- "/Users/garrettpinkston/Desktop/Michigan/STAT506/Data/lahman_1871-2022.sqlite"
lahman <- dbConnect(SQLite(), dbname = db_path)

fielding <- dbReadTable(lahman, "Fielding") %>%
  filter(!is.na(PO), !is.na(A), !is.na(InnOuts), InnOuts > 0) %>%
  mutate(RF = 3 * (PO + A) / InnOuts)

team_avg_rf <- fielding %>%
  group_by(teamID) %>%
  summarise(avg_RF = mean(RF, na.rm = TRUE))

bootstrap_iter <- function(data) {
  data %>%
    group_by(teamID) %>%
    group_modify(~ .x[sample(nrow(.x), replace = TRUE), ]) %>%
    summarise(avg_RF = mean(RF, na.rm = TRUE))
}

plan(multisession)

n_boot <- 1000
boot_results <- future_map_dfr(1:n_boot, ~ bootstrap_iter(fielding), .options = furrr_options

boot_sd <- boot_results %>%
  group_by(teamID) %>%
  summarise(sd_RF = sd(avg_RF, na.rm = TRUE))

final_results <- team_avg_rf %>%
  left_join(boot_sd, by = "teamID")

print(final_results)
```

```
# A tibble: 140 x 3
   teamID avg_RF   sd_RF
   <chr>   <dbl>   <dbl>
 1 ALT     0.387 0.0499
 2 ANA     0.415 0.0153
 3 ARI     0.366 0.00874
 4 ATL     0.388 0.00610
```

```
 5 BAL     0.393 0.00556
 6 BFN     0.451 0.0204
 7 BFP     0.464 0.0516
 8 BL1     0.442 0.0291
 9 BL2     0.401 0.0160
10 BL3     0.444 0.0409
# i 130 more rows
```

```
dbDisconnect(lahman)
```