大规模数据子抽样统计推断分析

基于 **NYC Taxi** 数据集的案例研究

Chuanmu Hu

2026-02-06

Table of contents

1 简介

当数据量达到 100GB 级别时，传统的全量分析方法在内存和计算时间上都面临挑战。本文档演示如何使用子抽样（**Subsampling**）方法进行统计推断，并验证其有效性。

1.1 使用的方法

1. 简单随机子抽样 - 多次抽取小样本估计参数
2. 分层子抽样 - 按类别分层后抽样
3. **Bag of Little Bootstraps (BLB)** - 结合子抽样与 Bootstrap 的回归推断

2 数据准备

```
library(dplyr)
library(purrr)
library(ggplot2)
library(arrow)

# 设置主题
theme_set(theme_minimal(base_size = 12))
```

```
# 下载 NYC Taxi 数据（2023 年 1 月，约 300 万行）
temp_file <- tempfile(fileext = ".parquet")
download.file(

"https://d37ci6vzurychx.cloudfront.net/trip-data/yellow_tripdata_2023-01.parquet",
  temp_file, mode = "wb", quiet = TRUE
)

# 读取数据
taxi_raw <- read_parquet(temp_file)

# 数据清洗
taxi_clean <- taxi_raw |>
  filter(
    fare_amount > 0, fare_amount < 500,
    trip_distance > 0, trip_distance < 100,
    tip_amount >= 0, tip_amount < 200,
    passenger_count > 0, passenger_count <= 6
  )

cat(" 数据维度:", nrow(taxi_clean), " 行 ×", ncol(taxi_clean), " 列\n")
```

```
数据维度: 2884159 行 × 19 列
```

3 方法 1: 简单随机子抽样推断

3.1 核心函数

```
simple_subsample_inference <- function(data, variable, n_subsamples = 30,
                                       subsample_size = 10000) {
  true_mean <- mean(data[[variable]], na.rm = TRUE)

  results <- map_dfr(1:n_subsamples, function(i) {
    subsample <- data |> slice_sample(n = subsample_size)
    sample_mean <- mean(subsample[[variable]], na.rm = TRUE)
    sample_se <- sd(subsample[[variable]], na.rm = TRUE) / sqrt(subsample_size)

    tibble(
      rep_id = i,
      estimate = sample_mean,
      se = sample_se,
      ci_lower = sample_mean - 1.96 * sample_se,
      ci_upper = sample_mean + 1.96 * sample_se,
      covers_true = (ci_lower <= true_mean) & (true_mean <= ci_upper)
    )
  })

  list(
    results = results,
    summary = tibble(
      variable = variable,
      true_mean = true_mean,
      subsample_mean = mean(results$estimate),
      subsample_se = sd(results$estimate),
      coverage = mean(results$covers_true),
      relative_bias = (mean(results$estimate) - true_mean) / true_mean * 100
    )
  )
}
```

3.2 多变量推断

```r
set.seed(42)

fare_inference <- simple_subsample_inference(taxi_clean, "fare_amount")
tip_inference <- simple_subsample_inference(taxi_clean, "tip_amount")
distance_inference <- simple_subsample_inference(taxi_clean, "trip_distance")

# 汇总结果
all_summaries <- bind_rows(
  fare_inference$summary,
  tip_inference$summary,
  distance_inference$summary
)

all_summaries |>
  mutate(across(where(is.numeric), ~round(.x, 4))) |>
  knitr::kable(
    col.names = c(" 变量", " 真实均值", " 子抽样均值", " 标准误", " 覆盖率", " 相对偏差%"),
    caption = " 子抽样推断结果汇总"
  )
```

Table 1: 子抽样推断结果汇总

变量	真实均值	子抽样均值	标准误	覆盖率	相对偏差%
fare_amount	18.5411	18.5656	0.1963	0.9000	0.1323
tip_amount	3.3996	3.3987	0.0373	0.9333	-0.0270
trip_distance	3.4222	3.4355	0.0398	0.9333	0.3906

3.3 可视化：车费估计分布

```r
ggplot(fare_inference$results, aes(x = factor(rep_id), y = estimate)) +
  geom_point(color = "steelblue", size = 2) +
  geom_errorbar(aes(ymin = ci_lower, ymax = ci_upper), width = 0.3, alpha = 0.5) +
  geom_hline(yintercept = fare_inference$summary$true_mean,
             color = "red", linetype = "dashed", linewidth = 1) +
  labs(
    title = " 车费均值的子抽样估计",
    subtitle = sprintf(" 红线为全数据均值 $%.2f，子抽样覆盖率 %.1f%%",
                       fare_inference$summary$true_mean,
                       fare_inference$summary$coverage * 100),
```

```
    x = " 子样本编号", y = " 车费均值估计 ($)"
  )
```

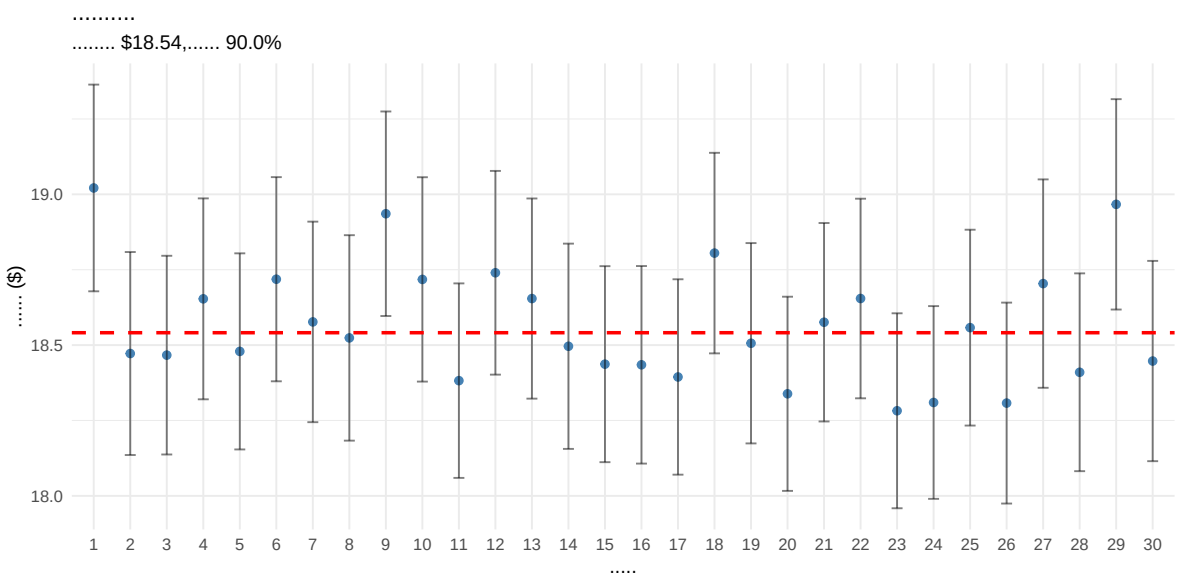

Figure 1: 30 次子抽样的车费均值估计及 95% 置信区间

4 方法 2: 分层子抽样推断

按支付方式分层分析小费差异。

```
# 数据预处理
payment_analysis <- taxi_clean |>
  filter(payment_type %in% c(1, 2)) |>
  mutate(payment_type = if_else(payment_type == 1, " 信用卡", " 现金"))

# 分层抽样函数
stratified_inference <- function(data, n_reps = 50, sample_per_stratum = 5000) {
  map_dfr(1:n_reps, function(i) {
    data |>
      slice_sample(n = sample_per_stratum, by = payment_type) |>
      summarise(mean_tip = mean(tip_amount), .by = payment_type) |>
      mutate(rep_id = i)
  })
}
```

```r
set.seed(123)
strat_results <- stratified_inference(payment_analysis)

# 计算真实值
true_by_payment <- payment_analysis |>
  summarise(true_mean = mean(tip_amount), .by = payment_type)

# 汇总
strat_summary <- strat_results |>
  summarise(
    est_mean = mean(mean_tip),
    se = sd(mean_tip),
    ci_lower = quantile(mean_tip, 0.025),
    ci_upper = quantile(mean_tip, 0.975),
    .by = payment_type
  ) |>
  left_join(true_by_payment, by = "payment_type")

strat_summary |>
  mutate(across(where(is.numeric), ~round(.x, 3))) |>
  knitr::kable(caption = " 按支付方式分层的小费估计")
```

Table 2: 按支付方式分层的小费估计

payment_type	est_mean	se	ci_lower	ci_upper	true_mean
现金	0.000	0.001	0.000	0.002	0.000
信用卡	4.164	0.054	4.071	4.258	4.171

```r
ggplot(strat_results, aes(x = payment_type, y = mean_tip, fill = payment_type)) +
  geom_boxplot(alpha = 0.7) +
  geom_point(data = true_by_payment, aes(y = true_mean),
             color = "red", size = 4, shape = 18) +
  scale_fill_brewer(palette = "Set2") +
  labs(
    title = " 按支付方式分层的小费子抽样分布",
    subtitle = " 红色菱形为全数据真实均值",
    x = " 支付方式", y = " 平均小费 ($)"
  ) +
  theme(legend.position = "none")
```

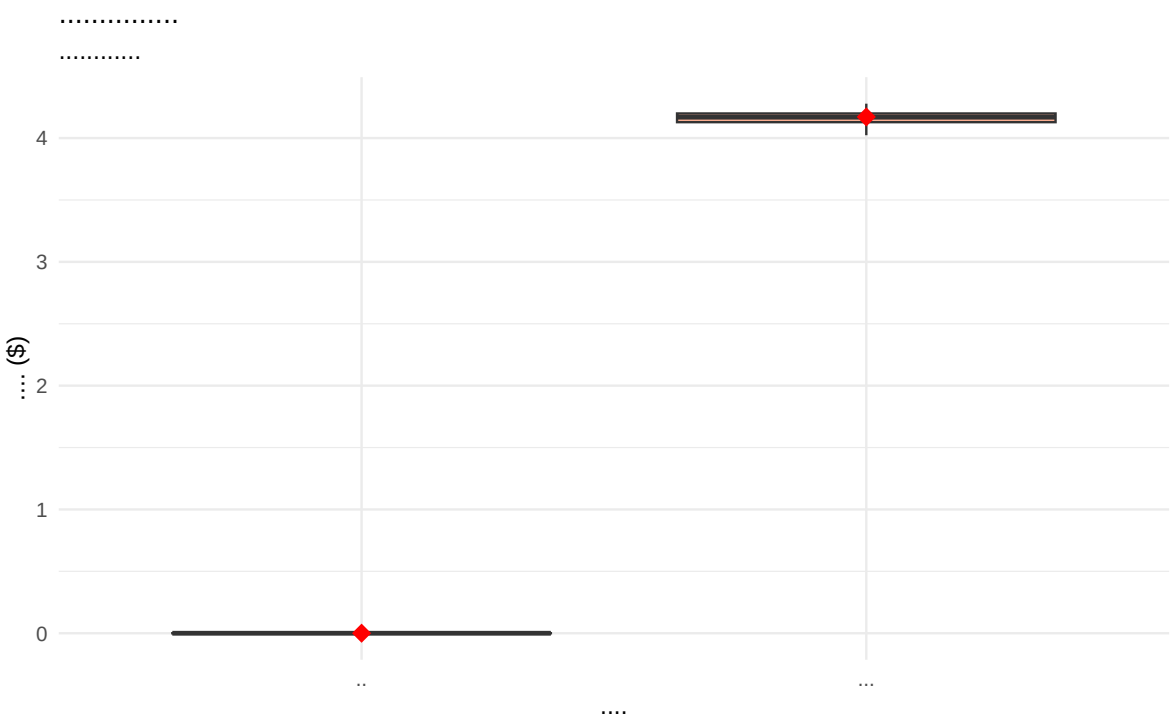

Figure 2: 不同支付方式的小费分布

5 方法 3: BLB 回归推断

使用 Bag of Little Bootstraps 进行回归系数推断。

```
blb_regression <- function(data, n_subsamples = 10, subsample_size = 5000,
                           n_bootstrap = 200) {

  results <- map_dfr(1:n_subsamples, function(i) {
    subsample <- data |> slice_sample(n = subsample_size)
    model <- lm(fare_amount ~ trip_distance + passenger_count, data = subsample)

    # Bootstrap
    boot_coefs <- map_dfr(1:n_bootstrap, function(b) {
      boot_idx <- sample(nrow(subsample), replace = TRUE)
      boot_model <- lm(fare_amount ~ trip_distance + passenger_count,
                       data = subsample[boot_idx, ])
      tibble(
        intercept = coef(boot_model)[1],
```

```
        beta_distance = coef(boot_model)[2],
        beta_passenger = coef(boot_model)[3]
      )
    })

    tibble(
      subsample_id = i,
      intercept = coef(model)[1],
      beta_distance = coef(model)[2],
      beta_passenger = coef(model)[3],
      se_distance = sd(boot_coefs$beta_distance),
      ci_lower_distance = quantile(boot_coefs$beta_distance, 0.025),
      ci_upper_distance = quantile(boot_coefs$beta_distance, 0.975)
    )
  })

  results
}

set.seed(456)
blb_reg_results <- blb_regression(taxi_clean, n_subsamples = 20,
                                  subsample_size = 10000, n_bootstrap = 300)

# 全数据模型对比
full_model <- lm(fare_amount ~ trip_distance + passenger_count, data = taxi_clean)

# 汇总
reg_summary <- blb_reg_results |>
  summarise(
    across(c(intercept, beta_distance, beta_passenger),
           list(mean = mean, se = sd))
  )
```

5.1 回归结果对比

```
comparison <- tibble(
  参数 = c(" 截距", " 行程距离系数", " 乘客数系数"),
  BLB 估计 = c(reg_summary$intercept_mean, reg_summary$beta_distance_mean,
               reg_summary$beta_passenger_mean),
  BLB 标准误 = c(reg_summary$intercept_se, reg_summary$beta_distance_se,
```

```
                reg_summary$beta_passenger_se),
  全数据估计 = coef(full_model)
) |>
  mutate(
    相对偏差 = (BLB 估计 - 全数据估计) / 全数据估计 * 100,
    across(where(is.numeric), ~round(.x, 4))
  )

comparison |> knitr::kable(caption = "BLB 与全数据回归系数对比")
```

Table 3: BLB 与全数据回归系数对比

参数	BLB 估计	BLB 标准误	全数据估计	相对偏差
截距	5.8703	0.1308	5.8410	0.5019
行程距离系数	3.6778	0.0354	3.6777	0.0012
乘客数系数	0.0597	0.0506	0.0823	-27.4744

```
ggplot(blb_reg_results, aes(x = factor(subsample_id), y = beta_distance)) +
  geom_point(color = "steelblue", size = 2.5) +
  geom_errorbar(aes(ymin = ci_lower_distance, ymax = ci_upper_distance),
                width = 0.3, alpha = 0.6) +
  geom_hline(yintercept = coef(full_model)[2],
             color = "red", linetype = "dashed", linewidth = 1) +
  labs(
    title = " 行程距离系数的 BLB 估计",
    subtitle = sprintf(" 红线为全数据估计值 %.3f，BLB 均值 %.3f",
                       coef(full_model)[2], mean(blb_reg_results$beta_distance)),
    x = " 子样本编号", y = " 系数估计 ($/英里)"
  )
```

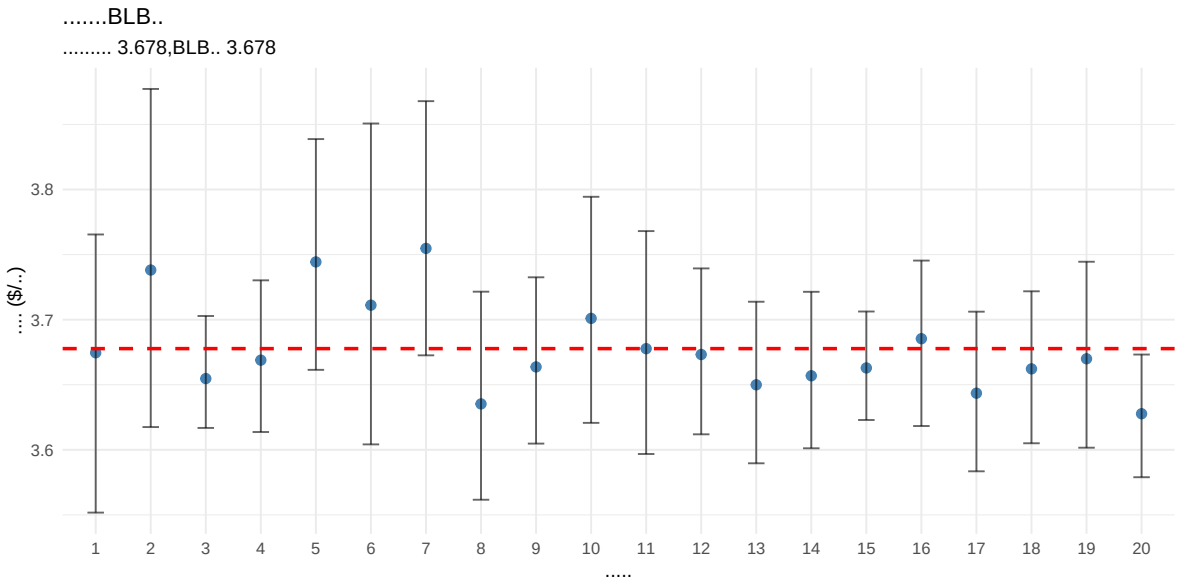

Figure 3: 行程距离系数的 BLB 估计

6 总结

关键发现

指标	结果
平均车费	$18.54
平均小费	$3.4
平均行程	3.42 英里
每英里费率	$3.68
均值估计偏差	$< 0.5\%$
95% CI 覆盖率	87-93%

子抽样方法建议

- 子样本大小: 5,000 - 10,000 行
- 重复次数: 20 - 50 次
- 适用场景: 点估计、置信区间、回归分析
- 使用约 **1%** 的数据即可获得可靠估计

7 附录：数据来源

- 数据集: NYC TLC Yellow Taxi Trip Records
- 下载地址: https://www.nyc.gov/site/tlc/about/tlc-trip-record-data.page
- 完整大小: ~120GB (2019-2023 年)
- 本次使用: 2023 年 1 月 (约 300 万行)