截尾和缩尾

黄嘉平

2025-03-06

1 关于截尾和缩尾

在一部份学术论文和很多学位论文中，作者在进行计量分析前会对数据进行截尾或缩尾操作。下面我们讨论该操作给分析结果带来的影响，以及是否应该在应用计量研究中进行这样的预处理。

截尾（trimming）和**缩尾（Winsorization）**都是通过剔除或更改极值的方式获得稳健统计量的方法，后者的名称源于提出该方法的统计学家 Charles P. Winsor。对于样本量为 N 的单变量样本，取 $k < N/2$，则

- （对称）截尾（trimming）：删除样本中最小和最大的 k 个观测值。
- （对称）缩尾（Winsorization）：将样本中最小的 k 个观测值更改为第 $k+1$ 小的观测值 $X_{k+1:N}$，同时将最大的 k 个观测值更改为第 $k+1$ 大的观测值 $X_{k:N}$。

经过这样操作后再计算的统计量称为 trimmed 或 Winsorized 统计量，例如 trimmed mean，Winsorized variance 等。我们知道样本均值和样本方差极易受异常值的影响，截尾和缩尾可以一定程度上减小该影响。

DescTools 程序包提供的 `Trim()` 和 `Winsorize()` 命令可以完成截尾和缩尾的操作。前者仅支持对称操作，后者则可进行非对称操作。

```
library(DescTools)
set.seed(111)
x <- round(rnorm(10) * 10, 0)
x
```

```
##  [1]   2  -3  -3 -23  -2   1 -15 -10  -9  -5
```

```
Trim(x, trim = 0.2) # 在 20% 和 80% 处截尾
```

```
## [1]  -3  -3  -2 -10  -9  -5
## attr(,"trim")
## [1] 4 7 6 1
```

```
Trim(x, trim = 1) # 上下两端各删除 1 个观测值
```

```
## [1]  -3  -3  -2   1 -15 -10  -9  -5
## attr(,"trim")
```

```
## [1] 4 1
```

```
Winsorize(x) # 默认在 5% 和 95% 处缩尾
```

```
##  [1]   1.55  -3.00  -3.00 -19.40  -2.00   1.00 -15.00 -10.00  -9.00  -5.00
```

```
Winsorize(x, val = c(Small(x, 3)[3], Large(x, 3)[1])) # 在第 3 小和第 3 大观测值处缩尾
```

```
##  [1]  -2  -3  -3 -10  -2  -2 -10 -10  -9  -5
```

```
Winsorize(x, val = quantile(x, probs = c(0, 0.9))) # 仅在上端 90% 处缩尾
```

```
##  [1]   1.1  -3.0  -3.0 -23.0  -2.0   1.0 -15.0 -10.0  -9.0  -5.0
```

这里需要注意 `Winsorize()` 函数中利用了 `quantile()` 函数计算样本分位点，而 `quantile()` 函数共提供了 9 种不同的计算方法可供选择。具体可参考帮助文档 `help(quantile)`。

2 截尾和缩尾对回归分析的影响

下面我们参考 Lien & Balakrishnan (2005) 探索截尾和缩尾对回归分析的影响。

模型：

$$Y_i = 5 + 2X_i + u_i, \quad X_i \sim N(0,1), \quad u_i \sim N(0,1)$$

数据预处理方式：针对解释变量 X，考虑下面几种数据预处理方式

1. 不处理
2. 对上下 5% 的观测值进行截尾
3. 对上下 10% 的观测值进行截尾
4. 对上下 20% 的观测值进行截尾
5. 对上下 5% 的观测值进行缩尾
6. 对上下 10% 的观测值进行缩尾
7. 对上下 20% 的观测值进行缩尾

目标统计量：我们对比不同预处理方式下几个核心统计量的变化

1. 常数项 β_0 和斜率 β_1 的 OLS 估计值
2. 常数项和斜率的标准误
3. 残差标准误
4. 决定系数 R^2

```
## 生成数据
n <- 10000 # 样本量
set.seed(555)
sim_data_0 <- tibble(
  x = rnorm(n),
  u = rnorm(n),
  y = 5 + 2 * x + u
)

## 预处理数据
sim_data_t1 <- filter(sim_data_0, x %in% Trim(x, 0.05))
sim_data_t2 <- filter(sim_data_0, x %in% Trim(x, 0.1))
sim_data_t3 <- filter(sim_data_0, x %in% Trim(x, 0.2))
sim_data_w1 <- mutate(sim_data_0, x = Winsorize(x, quantile(x, probs = c(0.05, 0.95))))
sim_data_w2 <- mutate(sim_data_0, x = Winsorize(x, quantile(x, probs = c(0.1, 0.9))))
sim_data_w3 <- mutate(sim_data_0, x = Winsorize(x, quantile(x, probs = c(0.2, 0.8))))

sim_data <- list(
  sim_data_0,
  sim_data_t1,
  sim_data_t2,
  sim_data_t3,
  sim_data_w1,
  sim_data_w2,
  sim_data_w3
) # 将七种样本保存在同一个 list 下以备后用

## 回归
model <- sim_data |>
  map(\(df) lm(y ~ x, data = df))
# 利用 map() 函数对 sim_data 中的每一项数据进行同样的操作

## 提取目标统计量
model |>
  map(coef) |>
  map_dbl(1) # 常数项的估计值
```

```
## [1] 4.985640 4.988002 4.993808 4.985582 4.994855 4.999812 5.006978
```

```
model |>
  map(coef) |>
  map_dbl(2) # 斜率的估计值
```

```
## [1] 2.012839 2.023068 2.021259 2.029724 2.177826 2.368717 2.924275
```

```
model |>
  map(vcov) |>
  map(diag) |>
  map(sqrt) |>
  map_dbl(1) # 常数项的标准误
```

```
## [1] 0.01007674 0.01063260 0.01129488 0.01290955 0.01052238 0.01107067 0.01221934
```

```
model |>
  map(vcov) |>
  map(diag) |>
  map(sqrt) |>
  map_dbl(2) # 斜率的标准误
```

```
## [1] 0.01013773 0.01350530 0.01707406 0.02784519 0.01158571 0.01346166 0.01902200
```

```
model |>
  map(summary) |>
  map_dbl("sigma") # 残差标准误
```

```
## [1] 1.0074603 1.0082788 1.0096313 0.9988219 1.0518958 1.1066196 1.2212561
```

```
model |>
  map(summary) |>
  map_dbl("r.squared") # R 方
```

```
## [1] 0.7976924 0.7137810 0.6366571 0.4697393 0.7794527 0.7559083 0.7027172
```

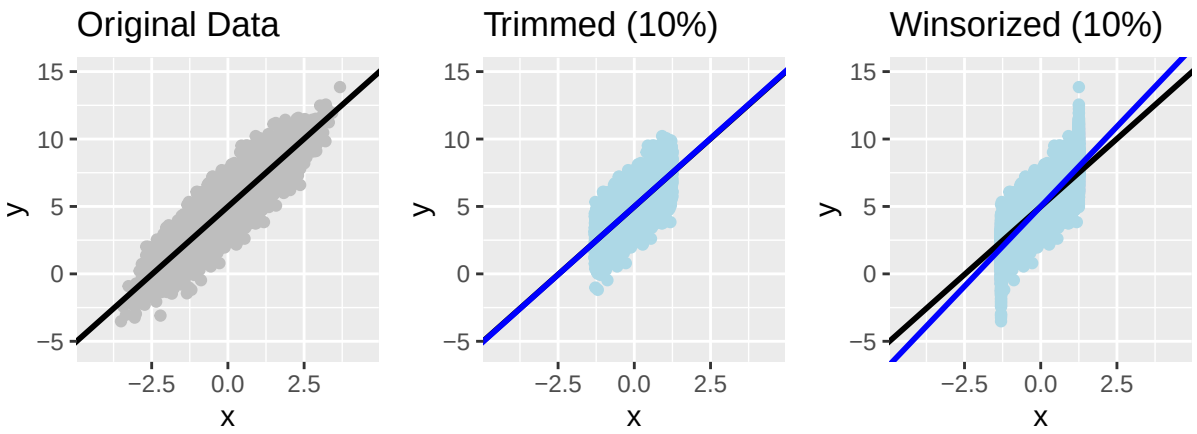
```
## 可视化
library(patchwork)
p1 <- sim_data_0 |>
  ggplot(aes(x, y)) +
  geom_point(color = "gray") +
  geom_abline(
    intercept = coef(model[[1]])[1],
    slope = coef(model[[1]])[2],
    linewidth = 1
  ) +
  coord_cartesian(
    xlim = c(-4.5, 4.5),
    ylim = c(-5.5, 15)
  ) +
  labs(title = "Original Data")

p2 <- sim_data_t2 |>
```

```
  ggplot(aes(x, y)) +
  geom_point(color = "lightblue") +
  geom_abline(
    intercept = coef(model[[1]])[1],
    slope = coef(model[[1]])[2],
    color = "black",
    linewidth = 1
  ) +
  geom_abline(
    intercept = coef(model[[3]])[1],
    slope = coef(model[[3]])[2],
    color = "blue",
    linewidth = 1
  ) +
  coord_cartesian(
    xlim = c(-4.5, 4.5),
    ylim = c(-5.5, 15)
  ) +
  labs(title = "Trimmed (10%)")

p3 <- sim_data_w2 |>
  ggplot(aes(x, y)) +
  geom_point(color = "lightblue") +
  geom_abline(
    intercept = coef(model[[1]])[1],
    slope = coef(model[[1]])[2],
    color = "black",
    linewidth = 1
  ) +
  geom_abline(
    intercept = coef(model[[6]])[1],
    slope = coef(model[[6]])[2],
    color = "blue",
    linewidth = 1
  ) +
  coord_cartesian(
    xlim = c(-4.5, 4.5),
    ylim = c(-5.5, 15)
  ) +
  labs(title = "Winsorized (10%)")

p1 + p2 + p3
```



从上面的结果可以看出，

- 截尾并不影响常数项和斜率的估计值；缩尾不影响常数项的估计值，但会增加斜率的估计值。
- 截尾和缩尾都会增加回归系数的标准误。
- 截尾不影响残差标准误，缩尾会增加残差标准误。
- 截尾和缩尾都会降低决定系数 R^2，截尾更加明显。

由此可见，截尾和缩尾虽然有可能减少异常值的影响（这或许是实证研究中进行这类操作的理由），但它们并不是建立在判断的基础上进行数据预处理。如果数据中不存在异常值，这类操作会给回归结果增加系统性偏误。

3 课后练习

改变文中的设定（例如调整真实模型中的系数值，更改 X 的分布，增加解释变量等），进一步验证文中观察到的现象是否是具有普遍性。

参考文献

Lien, D., & Balakrishnan, N. (2005). On Regression Analysis with Data Cleaning via Trimming, Winsorization, and Dichotomization. *Communications in Statistics – Simulation and Computation*, 34(4): 839-849. DOI: 10.1080/03610910500307695