Robust test Q2

Zhong

2025-06-06

#The Coleman data set contains information on 20 schools from the Mid-Atlantic and New England states, drawn from a population study. This data set consists of measurements on six different variables, one of which will be treated as response. They can be described as follows:

#X1: staff salaries per pupil #X2: percent of white-collar fathers #X3: socio-economic status composite (deviation means for family size, family intactness, father’s education, mother’s education, and home items) #X4: mean teacher’s verbal test score #X5: mean mother’s educational level (one unit is equal to two school years) #Y: verbal mean test score (all sixth graders) #This data is available in the robustbase package: data(coleman).

#(a) Estimate (β0,β1,β2,β3,β4) by each of the following methods and report the fitted coefficients with their estimated standard errors:

#Ordinary Least Squares (OLS) #Huber M-estimator with tuning constant c=1.345 #Tukey bisquare M-estimator with tuning constant c=4.685

#(Hint: first attach the relevant packages with library(robustbase) to load the data and library(MASS) for rlm().)

# step 1 input data

library(robustbase)#先加载这个数据包才能读取数据

## Warning: package 'robustbase' was built under R version 4.4.3

data("coleman") #记得加“”  
str(coleman)

## 'data.frame': 20 obs. of 6 variables:  
## $ salaryP : num 3.83 2.89 2.86 2.92 3.06 2.07 2.52 2.45 3.13 2.44 ...  
## $ fatherWc : num 28.9 20.1 69 65.4 29.6 ...  
## $ sstatus : num 7.2 -11.71 12.32 14.28 6.31 ...  
## $ teacherSc: num 26.6 24.4 25.7 25.7 25.4 ...  
## $ motherLev: num 6.19 5.17 7.04 7.1 6.15 6.41 6.86 5.78 6.51 5.57 ...  
## $ Y : num 37 26.5 36.5 40.7 37.1 ...

head(coleman)

## salaryP fatherWc sstatus teacherSc motherLev Y  
## 1 3.83 28.87 7.20 26.6 6.19 37.01  
## 2 2.89 20.10 -11.71 24.4 5.17 26.51  
## 3 2.86 69.05 12.32 25.7 7.04 36.51  
## 4 2.92 65.40 14.28 25.7 7.10 40.70  
## 5 3.06 29.59 6.31 25.4 6.15 37.10  
## 6 2.07 44.82 6.16 21.6 6.41 33.90

colnames(coleman)[1:5] <- c("X1","X2","X3","X4","X5")#重新命名变量，记得加""！

#step 2 三种回归模型 #Ordinary Least Squares (OLS) #Huber M-estimator with tuning constant c=1.345 #Tukey bisquare M-estimator with tuning constant c=4.685

library("MASS")  
#OLS  
ols\_model <- lm(Y~X1+X2+X3+X4+X5,coleman)  
summary(ols\_model)

##   
## Call:  
## lm(formula = Y ~ X1 + X2 + X3 + X4 + X5, data = coleman)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -3.9497 -0.6174 0.0623 0.7343 5.0018   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 19.94857 13.62755 1.464 0.1653   
## X1 -1.79333 1.23340 -1.454 0.1680   
## X2 0.04360 0.05326 0.819 0.4267   
## X3 0.55576 0.09296 5.979 3.38e-05 \*\*\*  
## X4 1.11017 0.43377 2.559 0.0227 \*   
## X5 -1.81092 2.02739 -0.893 0.3868   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 2.074 on 14 degrees of freedom  
## Multiple R-squared: 0.9063, Adjusted R-squared: 0.8728   
## F-statistic: 27.08 on 5 and 14 DF, p-value: 9.927e-07

#Huber ME  
huber\_model <- rlm(Y~X1+X2+X3+X4+X5,data=coleman,  
 psi=psi.huber,  
 k=1.345)  
summary (huber\_model)

##   
## Call: rlm(formula = Y ~ X1 + X2 + X3 + X4 + X5, data = coleman, psi = psi.huber,   
## k = 1.345)  
## Residuals:  
## Min 1Q Median 3Q Max   
## -4.2059 -0.3886 -0.1092 0.4231 6.7054   
##   
## Coefficients:  
## Value Std. Error t value  
## (Intercept) 27.3497 7.6808 3.5608  
## X1 -1.6207 0.6952 -2.3314  
## X2 0.0752 0.0300 2.5045  
## X3 0.6401 0.0524 12.2182  
## X4 1.1557 0.2445 4.7271  
## X5 -3.5195 1.1427 -3.0801  
##   
## Residual standard error: 0.7461 on 14 degrees of freedom

#Tukey ME  
tukey\_model <- rlm(Y~X1+X2+X3+X4+X5,data=coleman,  
 psi=psi.bisquare,  
 k=4.658)

## Warning in rlm.default(x, y, weights, method = method, wt.method = wt.method, :  
## some of ... do not match

summary(tukey\_model)

##   
## Call: rlm(formula = Y ~ X1 + X2 + X3 + X4 + X5, data = coleman, psi = psi.bisquare,   
## k = 4.658)  
## Residuals:  
## Min 1Q Median 3Q Max   
## -4.27163 -0.44672 -0.00388 0.49205 7.14949   
##   
## Coefficients:  
## Value Std. Error t value  
## (Intercept) 29.3415 6.0570 4.8443  
## X1 -1.6328 0.5482 -2.9785  
## X2 0.0823 0.0237 3.4781  
## X3 0.6653 0.0413 16.1036  
## X4 1.1743 0.1928 6.0912  
## X5 -3.9705 0.9011 -4.4063  
##   
## Residual standard error: 0.6964 on 14 degrees of freedom

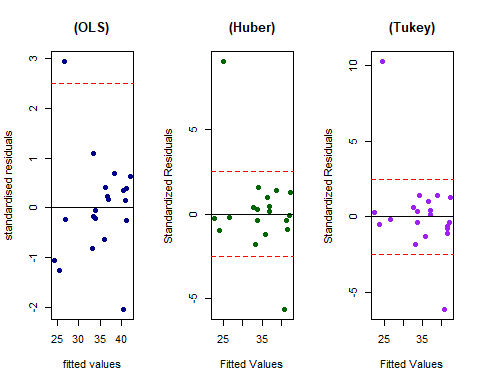
# step 3 整理数据生成表格

#提取系数和标准误差  
ols\_coe<-summary(ols\_model)$coefficients  
huber\_coe<-summary(huber\_model)$coefficients  
tukey\_coe<-summary(tukey\_model)$coefficients  
#构造表格  
coe\_names <- rownames(ols\_coe)#rownames() 提取的是 每一行所代表的对象名  
#构造结果表格  
results\_2 <- data.frame(  
 Term=coe\_names,  
 OLS\_estimate=round(ols\_coe[,"Estimate"],4),#round保留4位小数  
 OLS\_se=round(ols\_coe[,"Std. Error"],4),  
   
 Huber\_estimate=round(huber\_coe[,"Value"],4),#没有列名叫 "Estimate"，而是叫 "Value"  
 Huber\_se=round(huber\_coe[,"Std. Error"],4),  
   
 Tukey\_estimate=round(tukey\_coe[,"Value"],4),  
 Tukey\_se=round(tukey\_coe[,"Std. Error"],4)  
)  
View(results\_2)

#(b) Produce scatterplot of standardised residuals versus fitted values. Attach a horizontal reference at ∣r∣=±2.5.

#OLS散点图

par(mfrow = c(1, 3))  
#标准化残差  
ols\_residuals <- rstandard(ols\_model)  
#fitted value  
ols\_fitted <- fitted(ols\_model)  
#plot  
plot(ols\_fitted,ols\_residuals,  
 main="(OLS)",  
 xlab="fitted values",  
 ylab="standardised residuals",  
 pch=19,col="darkblue")  
#添加辅助线  
abline(h=abs(2.5),col="red",lty=2)#2是虚线  
abline(h = 0, col = "black", lty = 1)#1是实线  
  
#Huber 标准差VS拟合值散点图  
  
# Huber 标准化残差  
huber\_resid <- residuals(huber\_model)  
huber\_scale <- huber\_model$s  
huber\_rstd <- huber\_resid / huber\_scale  
  
huber\_fitted <- fitted(huber\_model)  
  
plot(huber\_fitted, huber\_rstd,  
 main = "(Huber)",  
 xlab = "Fitted Values",  
 ylab = "Standardized Residuals",  
 pch = 19, col = "darkgreen")  
  
abline(h = 2.5, col = "red", lty = 2)  
abline(h = -2.5, col = "red", lty = 2)  
abline(h = 0, col = "black", lty = 1)  
  
# 3 Tukey标准差VS拟合值散点图  
# Tukey 标准化残差  
tukey\_resid <- residuals(tukey\_model)  
tukey\_scale <- tukey\_model$s  
tukey\_rstd <- tukey\_resid / tukey\_scale  
  
tukey\_fitted <- fitted(tukey\_model)  
  
plot(tukey\_fitted, tukey\_rstd,  
 main = "(Tukey)",  
 xlab = "Fitted Values",  
 ylab = "Standardized Residuals",  
 pch = 19, col = "purple")  
  
abline(h = 2.5, col = "red", lty = 2)  
abline(h = -2.5, col = "red", lty = 2)  
abline(h = 0, col = "black", lty = 1)



#(c) Identify the observation(s) flagged as outliers under each of the estimators.

#outliers  
which(abs(ols\_residuals) > 2.5)

## 18   
## 18

outlier\_2 <- rstudent(ols\_model)  
#leverage  
leve\_2 <- hatvalues(ols\_model)  
n\_2 <- nrow(coleman)  
p\_2 <- length(coef(ols\_model))  
lev.thresh <- 2 \* p\_2 / n\_2 # 杠杆值阈值  
inf.thresh <- 4 / n\_2 # Cook's D 阈值  
  
#cook distance  
cook\_dis <- cooks.distance(ols\_model)  
outliers\_2 <-subset(coleman,abs(rstudent(ols\_model))>2.5)   
leverage\_2 <- subset(coleman,leve\_2>lev.thresh)  
cook\_2 <- subset(coleman,cook\_dis>inf.thresh)  
print(outliers\_2)

## X1 X2 X3 X4 X5 Y  
## 18 2.52 16.7 -10.99 24.8 6.01 31.7

print(leverage\_2)

## X1 X2 X3 X4 X5 Y  
## 10 2.44 9.99 -0.05 28.01 5.57 37.2

print(cook\_2)

## X1 X2 X3 X4 X5 Y  
## 18 2.52 16.7 -10.99 24.8 6.01 31.7

#d) Explain briefly why the Huber and Tukey procedures assign different weights to the same extreme observation(s). #Answer:Huber and Tukey procedures assign different weights to the same extreme observations because they use different weighting functions. #Huber’s method applies a soft threshold—it downweights large residuals but never assigns zero weight, so extreme observations still retain some influence. #Tukey’s method uses a hard threshold—observations with residuals beyond a certain cutoff receive zero weight, effectively excluding them from the fit. #As a result, Tukey is more aggressive in treating outliers, and may classify observations as outliers that Huber does not.