Robust test 2

Zhong

2025-06-05

#This is a data set consisting of measurements of water salinity (i.e., its salt concentration) and river discharge taken in North Carolina’s Pamlico Sound, recording some bi-weekly averages in March, April, and May from 1972 to 1977. This data is available in the robustbase package, data(salinity)

#(a):Model this relationship using:OLS and GMestimator KS2014setting ,Tabulate the estimated coefficients and their standard errors for both fits.

#1.load the data

## Load the robustbase package  
library(robustbase)

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

#input data  
data("salinity")  
#view structure of data   
head(salinity)

## X1 X2 X3 Y  
## 1 8.2 4 23.005 7.6  
## 2 7.6 5 23.873 7.7  
## 3 4.6 0 26.417 4.3  
## 4 4.3 1 24.868 5.9  
## 5 5.9 2 29.895 5.0  
## 6 5.0 3 24.200 6.5

str(salinity)

## 'data.frame': 28 obs. of 4 variables:  
## $ X1: num 8.2 7.6 4.6 4.3 5.9 5 6.5 8.3 10.1 13.2 ...  
## $ X2: int 4 5 0 1 2 3 4 5 0 1 ...  
## $ X3: num 23 23.9 26.4 24.9 29.9 ...  
## $ Y : num 7.6 7.7 4.3 5.9 5 6.5 8.3 8.2 13.2 12.6 ...

#step 2 using ols fit model  
#OLS fit  
ols\_fit <- lm(Y~X1+X2+X3,data = salinity)  
#summary OLS fit  
summary(ols\_fit)

##   
## Call:  
## lm(formula = Y ~ X1 + X2 + X3, data = salinity)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -2.6646 -0.7547 0.2267 0.6517 2.7202   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 9.59026 3.12509 3.069 0.00527 \*\*   
## X1 0.77711 0.08622 9.013 3.59e-09 \*\*\*  
## X2 -0.02551 0.16108 -0.158 0.87548   
## X3 -0.29504 0.10680 -2.762 0.01083 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 1.33 on 24 degrees of freedom  
## Multiple R-squared: 0.8264, Adjusted R-squared: 0.8047   
## F-statistic: 38.08 on 3 and 24 DF, p-value: 2.769e-09

#step 3 using GMestimator KS2014setting  
gm\_fit <- lmrob(Y~X1+X2+X3,data = salinity,setting="KS2014")  
#"KS2014"是一套预设的鲁棒控制参数，来自 Koller & Stahel (2014)，适用于很多实际问题，鲁棒性好，自动配置好一整套鲁棒回归参数  
summary(gm\_fit)

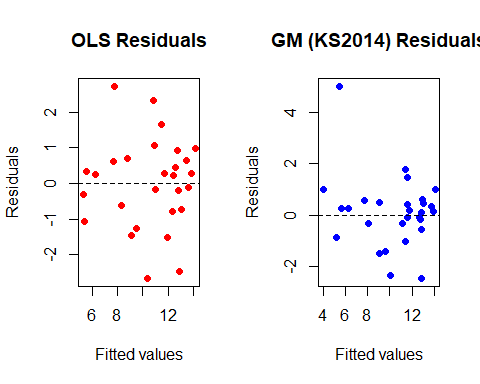
##   
## Call:  
## lmrob(formula = Y ~ X1 + X2 + X3, data = salinity, setting = "KS2014")  
## \--> method = "SMDM"  
## Residuals:  
## Min 1Q Median 3Q Max   
## -2.4487 -0.3815 0.1830 0.5066 5.0127   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 16.00586 3.38822 4.724 8.38e-05 \*\*\*  
## X1 0.72857 0.07478 9.743 8.15e-10 \*\*\*  
## X2 -0.13910 0.14068 -0.989 0.332617   
## X3 -0.53693 0.12066 -4.450 0.000168 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Robust residual standard error: 1.137   
## Multiple R-squared: 0.8764, Adjusted R-squared: 0.861   
## Convergence in 20 IRWLS iterations  
##   
## Robustness weights:   
## 21 weights are ~= 1. The remaining 7 ones are  
## 1 8 9 13 15 16 17   
## 0.9616 0.9711 0.8851 0.9602 0.6755 0.1248 0.7170   
## Algorithmic parameters:   
## tuning.chi1 tuning.chi2 tuning.chi3 tuning.chi4   
## -5.000e-01 1.500e+00 NA 5.000e-01   
## bb tuning.psi1 tuning.psi2 tuning.psi3   
## 5.000e-01 -5.000e-01 1.500e+00 9.500e-01   
## tuning.psi4 refine.tol rel.tol scale.tol   
## NA 1.000e-07 1.000e-07 1.000e-10   
## solve.tol zero.tol eps.outlier eps.x   
## 1.000e-07 1.000e-10 3.571e-03 6.083e-11   
## warn.limit.reject warn.limit.meanrw   
## 5.000e-01 5.000e-01   
## nResample max.it best.r.s k.fast.s k.max   
## 1000 500 20 2 2000   
## maxit.scale trace.lev mts compute.rd numpoints   
## 200 0 1000 0 10   
## fast.s.large.n   
## 2000   
## setting psi subsampling   
## "KS2014" "lqq" "nonsingular"   
## cov compute.outlier.stats   
## ".vcov.w" "SMDM"   
## seed : int(0)

#step 4 tabulate coefficients and standard errrors  
  
results <- data.frame(  
 ols\_coef = coef(ols\_fit),#data.frame 里面是=，不是<-，注意区别！！！  
 gm\_coef = coef(gm\_fit),  
 ols\_se = summary(ols\_fit)$coefficients[,2],  
 gm\_se = summary(gm\_fit)$coefficients[,2]#这里不要加逗号！！！  
)  
print(results)

## ols\_coef gm\_coef ols\_se gm\_se  
## (Intercept) 9.59026290 16.0058553 3.12508625 3.38822130  
## X1 0.77710540 0.7285679 0.08622195 0.07477515  
## X2 -0.02551234 -0.1391039 0.16107880 0.14067680  
## X3 -0.29503582 -0.5369297 0.10680378 0.12065720

#residual polts of two models

# 1. 设置一个图形窗口，画两张图并排  
par(mfrow = c(1, 2))  
  
# 2. OLS 残差图  
plot(ols\_fit$fitted.values, resid(ols\_fit),#plot散点图，横轴拟合值，纵轴残差  
 main = "OLS Residuals",#图表名称  
 xlab = "Fitted values",#X轴名称  
 ylab = "Residuals",#Y轴名称  
 col = "red", pch = 19)#点的形状是实心圆（代码 19 是一个点样式）  
abline(h = 0, lty = 2)#abline在图上画一条直线，h=0表示y = 0 的水平线，ity=2表示这条线是虚线（线型类型 2）  
  
# 3. GM 残差图  
plot(gm\_fit$fitted.values, resid(gm\_fit),  
 main = "GM (KS2014) Residuals",  
 xlab = "Fitted values",  
 ylab = "Residuals",  
 col = "blue", pch = 19)  
abline(h = 0, lty = 2)



#（b）Identify outliers and classify observations into vertical, good and bad leverage points.

#step 1 计算残差和杠杆值，cook distance  
library(car)

## Warning: package 'car' was built under R version 4.4.2

## Loading required package: carData

## Warning: package 'carData' was built under R version 4.4.2

df <- salinity  
df$restudent <- rstudent(ols\_fit)#残差（即实际值与拟合值的差）非常大（标准化残差 > 2 或 < -2）  
  
df$hat <- hatvalues(ols\_fit)# 杠杆值 hᵢ  
df$cook <- cooks.distance(ols\_fit)# Cook's D，用于衡量单个点对拟合结果的影响力  
  
#step 2 设定判断标准  
n <- nrow(salinity) # 总观测数  
p <- length(coef(ols\_fit)) # 参数数（包括截距）  
lev.thresh <- 2 \* p / n # 杠杆值阈值  
inf.thresh <- 4 / n # Cook's D 阈值  
  
#step 3 找出异常值  
outliers <- subset(df,abs(restudent)>2)#找出外部学生化残差的绝对值 > 2 的点，这些是可能的离群点。  
  
leverage\_1 <-subset(df,hat>lev.thresh)#找出杠杆值大于阈值的点，即高杠杆点  
influential\_1 <- subset(df,cook>inf.thresh)#找出Cook's 距离大于阈值的点，这些是可能会显著改变拟合结果的有影响力观测值。  
  
#step 4 可视化表格化这些极端点  
  
print(outliers)

## X1 X2 X3 Y restudent hat cook  
## 9 10.1 0 22.274 13.2 2.058125 0.17632629 0.19976314  
## 15 13.3 0 23.927 10.4 -2.155903 0.14803311 0.17526026  
## 16 10.4 1 33.443 10.5 3.788854 0.54665604 2.78035172  
## 17 10.5 2 24.859 7.7 -2.206240 0.04298953 0.04707657

print(leverage\_1)

## X1 X2 X3 Y restudent hat cook  
## 16 10.4 1 33.443 10.5 3.788854 0.546656 2.780352

print(influential\_1)

## X1 X2 X3 Y restudent hat cook  
## 9 10.1 0 22.274 13.2 2.058125 0.1763263 0.1997631  
## 15 13.3 0 23.927 10.4 -2.155903 0.1480331 0.1752603  
## 16 10.4 1 33.443 10.5 3.788854 0.5466560 2.7803517

#(c)(i) Compare parameter estimates, standard errors, and the set of flagged outliers. #(ii) Explain why the GM weight function reduces the influence of the bad leverage observations you identified in part (b) and hence shifts the fitted surface.