Delivery 3: AMA

Marc Falcón Barau, Julian Fransen,Victor Garcia Pizarro

2024-10-03

Table of Contents

# Import Data

We are using the Aircraft data from the R package sm. This dataset records six characteristics of aircraft designs that emerged during the twentieth century.

library(sm)

## Warning: package 'sm' was built under R version 4.2.3

## Package 'sm', version 2.2-6.0: type help(sm) for summary information

library(KernSmooth)

## KernSmooth 2.23 loaded  
## Copyright M. P. Wand 1997-2009

data(aircraft)  
# help(aircraft)  
attach(aircraft)  
  
lgPower <- log(Power)  
lgSpan <- log(Span)  
lgLength <- log(Length)  
lgWeight <- log(Weight)  
lgSpeed <- log(Speed)  
lgRange <- log(Range)

Now, we are interested in creating a heteroscedastic regression model with and . To do this, we need to calculate the error term , which must be approximated from the variance function , as it is not constant in this case. To find this function, we use the following approach:

1. Calculate the expected mean function .
2. Transform the estimated residuals ().
3. Estimate the function .
4. Calculate the variance function by computing .

First, we will use the locpolreg() function to create nonparametric models and apply cross-validation (CV) to select the optimal bandwidth. Then, we will use sm.regression() and dpill() to estimate the bandwidth.

# Approach 1: locpolreg() and CV approach

## 1. Fit a nonparametric regression to data (xi, yi) and save the estimated values m(xi).

source("locpolreg.R")  
  
optimal\_h1a <- h.select(x=Yr, y=lgWeight, method="cv")  
model1a <- locpolreg(x = Yr, y = lgWeight, h = optimal\_h1a, doing.plot=FALSE)  
  
# Get the estimated regression values  
estimated\_m <- model1a$mtgr

## 2. Transform the estimated residuals ϵi = yi − m(xi):

residuals <- lgWeight - estimated\_m  
stimated\_residuals <- log(residuals^2)

## 3. Fit a nonparametric regression to data (xi, zi) and call the estimated functionˆq(x). Observe that ˆq(x) is an estimate of log σ2(x).

optimal\_h2a <- h.select(x = Yr, y = stimated\_residuals, method="cv")  
model2a <- locpolreg(x = Yr, y = stimated\_residuals, h = optimal\_h2a, doing.plot=FALSE)

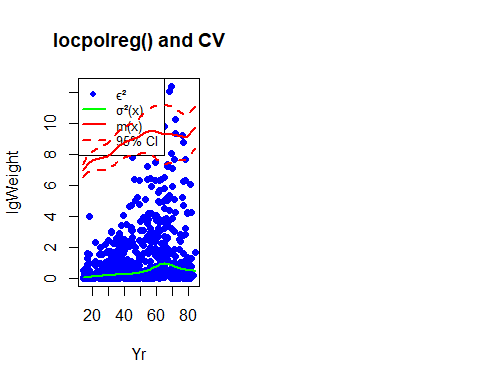
## 4. Estimate σ2(x)

estimated\_variance <- exp(model2a$mtgr)

## 5. Final plot

Finally, we combine all the elements that we have estimated: residuals, mean, and variance.

par(mfrow = c(1, 2))  
  
# Calculate and plot ϵ²ᵢ against xᵢ  
plot(Yr, residuals ^ 2,   
 xlab = "Yr",   
 ylab = "lgWeight",   
 main = "locpolreg() and CV",  
 col = "blue",   
 pch = 16) # Set the same y-axis range for both  
  
# Add a legend with all elements   
legend("topleft",   
 legend = c("ϵ²", "σ²(x)",   
 "m(x)", "95% CI"),  
 col = c("blue", "green", "red", "red"),  
 pch = c(16, NA, NA, NA), # points are residuals  
 lty = c(NA, 1, 1, 2), # Line type for other elements  
 lwd = c(NA, 2, 2, 2), # Line width   
 cex = 0.8,  
 xpd = TRUE  
)   
  
# Superpose estimated σ²(x)  
lines(Yr, estimated\_variance, col = "green", lwd = 2)   
  
# Superpose estimated mean with 95% CI  
lines(Yr, estimated\_m, col = "red", lwd = 2)   
estimated\_sd = sqrt(estimated\_variance)  
  
lower\_band <- estimated\_m - 1.96 \* estimated\_sd  
upper\_band <- estimated\_m + 1.96 \* estimated\_sd  
  
lines(Yr, lower\_band, col = "red", lty = 2, lwd = 2)   
lines(Yr, upper\_band, col = "red", lty = 2, lwd = 2)



# Approach 2: sm.regresion and dpill() approach

## 1. Fit a nonparametric regression to data (xi, yi) and save the estimated values m(xi).

This approach creates a different number of estimated points than , so it is necessary to perform linear interpolation to maintain the same number of points and ensure they can be plotted later.

optimal\_h1b <- dpill(x = Yr, y = lgWeight)  
model1b <- sm.regression(x = Yr, y = lgWeight, h = optimal\_h1b, display="none")  
  
# Get the estimated regression values  
estimated\_m <- model1b$estimate  
estimated\_m\_interpolated <- approx(model1b$eval.points, model1b$estimate, xout = Yr)$y

## 2. Transform the estimated residuals ϵi = yi − m(xi):

residuals <- lgWeight - estimated\_m\_interpolated  
stimated\_residuals <- log(residuals^2)

## 3. Fit a nonparametric regression to data (xi, zi) and call the estimated functionˆq(x). Observe that ˆq(x) is an estimate of log σ2(x).

optimal\_h2b <- dpill(x = Yr, y = stimated\_residuals)  
model2b <- sm.regression(x = Yr, y = stimated\_residuals, h = optimal\_h2b, display="none")

## missing data are removed

estimated\_q\_interpolated <- approx(model2b$eval.points, model2b$estimate, xout = Yr)$y

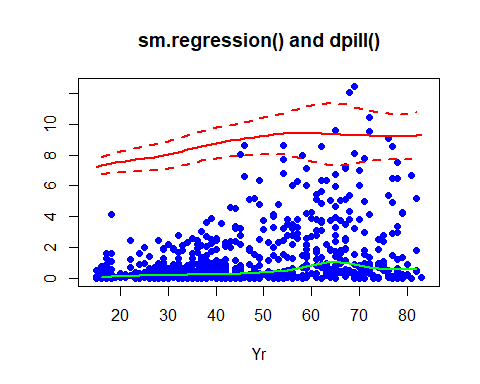
## 4. Estimate σ2(x)

estimated\_variance <- exp(estimated\_q\_interpolated)

## 5. Final plot

Again, we plot all elements we have calculated

# Calculate and plot ϵ²ᵢ against xᵢ  
plot(Yr, residuals ^ 2,   
 xlab = "Yr",   
 ylab = "",   
 main = "sm.regression() and dpill()",  
 col = "blue",   
 pch = 16) # Set the same y-axis range for both  
  
# Superpose estimated σ²(x)  
lines(Yr, estimated\_variance, col = "green", lwd = 2)   
  
# Superpose estimated mean with 95% CI  
lines(Yr, estimated\_m\_interpolated, col = "red", lwd = 2)   
estimated\_sd = sqrt(estimated\_variance)  
  
lower\_band <- estimated\_m\_interpolated - 1.96 \* estimated\_sd  
upper\_band <- estimated\_m\_interpolated + 1.96 \* estimated\_sd  
  
lines(Yr, lower\_band, col = "red", lty = 2, lwd = 2)   
lines(Yr, upper\_band, col = "red", lty = 2, lwd = 2)



# Conclusions

Both approaches are very similar, exhibiting low variance close to zero, as expected. The residuals are also well-distributed around zero, particularly in the earlier years. Overall, we observe that variance increases over time, resulting in wider confidence intervals (CIs) and a bigger residuals.