ST 503 Hw 5

Robin Baldeo

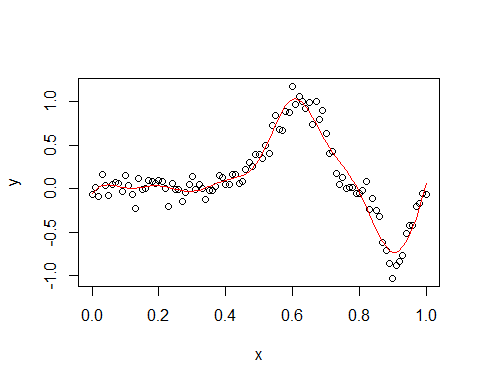
November 12, 2018

### Question 1(Exercise 10.7 in LMR)

set.seed(1)  
funky<- function(x) sin(2\*pi\*x^3)^3  
x<- seq(0,1, by=.01)  
y<- funky(x) + .1\*rnorm(101)

#### (A)

plot(x, y)  
mod12<- lm(y~bs(x,12))  
pr<- predict(mod12,newdata = as.data.frame(x ))  
lines(x, pr, col = "red")



#### (B)

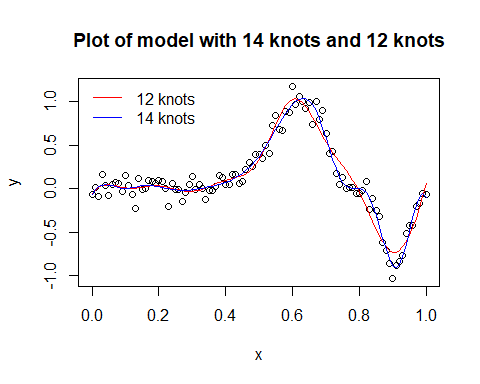
#using r built in AIC  
AIC(mod12)

## [1] -118.4576

The AIC for the model with 12 knots is -118.4575744.

#### (C)

plot(x, y, main= "Plot of model with 14 knots and 12 knots")  
lines(x, pr, col = "red")  
  
f<- function(p){  
 mod<- lm(y~bs(x,p))  
 return(AIC(mod))  
}  
  
  
aic\_<- matrix(, nrow= length(3:20), ncol = 2)  
best <- matrix(, nrow= 1, ncol = 2)  
  
#extract model with the lowest AIC  
for(i in seq(3:20)){  
 j = i + 2;  
 aic\_[i, 1]= f(j); aic\_[i, 2]= j  
 if(i == length(3:20)){  
 for(v in 1:length(3:20)){  
 if(aic\_[v, 1] == min(aic\_[, 1])){  
 #sotring the lowest aic to use as best plot  
 best[1,1] = aic\_[v,1]; best[1,2] = aic\_[v,2]  
 aic\_[v,1]  
 break  
 }  
 }  
 }  
}  
  
#best model 14 knots  
  
legend("topleft", legend = c("12 knots", "14 knots"),col=c("red", "blue"), lty=c(1,1, 1, 5), bty = "n")  
mod14<- lm(y~bs(x,best[1,2]))  
pr2<- predict(mod14,newdata = as.data.frame(x ))  
lines(x, pr2, col = "blue")



best

## [,1] [,2]  
## [1,] -178.3831 14

The lowest aic is -178.3830933, this is from the model with 14 knots. Therefore, the model with 14 knots is the best model.

### Question 2(Exercise 10.8 in LMR)

#### (A)

mod<- lm(odor~temp + gas + pack + I(temp^2)+ I(gas^2) + I(pack^2) + I(temp\*gas) + I(temp\* pack)+ I(pack\* gas), data= odor)  
summary(mod)

##   
## Call:  
## lm(formula = odor ~ temp + gas + pack + I(temp^2) + I(gas^2) +   
## I(pack^2) + I(temp \* gas) + I(temp \* pack) + I(pack \* gas),   
## data = odor)  
##   
## Residuals:  
## 1 2 3 4 5 6 7 8   
## -20.6250 -6.8750 6.8750 20.6250 15.5000 1.7500 -1.7500 -15.5000   
## 9 10 11 12 13 14 15   
## 5.1250 -22.3750 22.3750 -5.1250 -0.3333 -4.3333 4.6667   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -30.667 12.978 -2.363 0.06451 .   
## temp -12.125 7.947 -1.526 0.18761   
## gas -17.000 7.947 -2.139 0.08542 .   
## pack -21.375 7.947 -2.690 0.04332 \*   
## I(temp^2) 32.083 11.698 2.743 0.04067 \*   
## I(gas^2) 47.833 11.698 4.089 0.00946 \*\*  
## I(pack^2) 6.083 11.698 0.520 0.62524   
## I(temp \* gas) 8.250 11.239 0.734 0.49588   
## I(temp \* pack) 1.500 11.239 0.133 0.89903   
## I(pack \* gas) -1.750 11.239 -0.156 0.88236   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 22.48 on 5 degrees of freedom  
## Multiple R-squared: 0.882, Adjusted R-squared: 0.6696   
## F-statistic: 4.152 on 9 and 5 DF, p-value: 0.06569

#### (B)

o1<- update(mod,.~.-I(temp\*pack));summary(o1)

##   
## Call:  
## lm(formula = odor ~ temp + gas + pack + I(temp^2) + I(gas^2) +   
## I(pack^2) + I(temp \* gas) + I(pack \* gas), data = odor)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -22.3750 -6.0000 -0.3333 6.0000 22.3750   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -30.667 11.868 -2.584 0.04155 \*   
## temp -12.125 7.268 -1.668 0.14630   
## gas -17.000 7.268 -2.339 0.05792 .   
## pack -21.375 7.268 -2.941 0.02591 \*   
## I(temp^2) 32.083 10.698 2.999 0.02404 \*   
## I(gas^2) 47.833 10.698 4.471 0.00423 \*\*  
## I(pack^2) 6.083 10.698 0.569 0.59023   
## I(temp \* gas) 8.250 10.278 0.803 0.45278   
## I(pack \* gas) -1.750 10.278 -0.170 0.87040   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 20.56 on 6 degrees of freedom  
## Multiple R-squared: 0.8816, Adjusted R-squared: 0.7237   
## F-statistic: 5.583 on 8 and 6 DF, p-value: 0.02518

o2<- update(o1,.~.-I(pack\*gas));summary(o2)

##   
## Call:  
## lm(formula = odor ~ temp + gas + pack + I(temp^2) + I(gas^2) +   
## I(pack^2) + I(temp \* gas), data = odor)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -20.6250 -6.8750 -0.3333 5.7708 24.1250   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -30.667 11.014 -2.784 0.02713 \*   
## temp -12.125 6.745 -1.798 0.11528   
## gas -17.000 6.745 -2.520 0.03979 \*   
## pack -21.375 6.745 -3.169 0.01573 \*   
## I(temp^2) 32.083 9.928 3.232 0.01442 \*   
## I(gas^2) 47.833 9.928 4.818 0.00193 \*\*  
## I(pack^2) 6.083 9.928 0.613 0.55943   
## I(temp \* gas) 8.250 9.539 0.865 0.41575   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 19.08 on 7 degrees of freedom  
## Multiple R-squared: 0.881, Adjusted R-squared: 0.762   
## F-statistic: 7.403 on 7 and 7 DF, p-value: 0.008503

o3<- update(o2,.~.-I(temp\*gas));summary(o3)

##   
## Call:  
## lm(formula = odor ~ temp + gas + pack + I(temp^2) + I(gas^2) +   
## I(pack^2), data = odor)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -20.625 -9.625 -1.375 4.021 28.875   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -30.667 10.840 -2.829 0.0222 \*   
## temp -12.125 6.638 -1.827 0.1052   
## gas -17.000 6.638 -2.561 0.0336 \*   
## pack -21.375 6.638 -3.220 0.0122 \*   
## I(temp^2) 32.083 9.771 3.284 0.0111 \*   
## I(gas^2) 47.833 9.771 4.896 0.0012 \*\*  
## I(pack^2) 6.083 9.771 0.623 0.5509   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 18.77 on 8 degrees of freedom  
## Multiple R-squared: 0.8683, Adjusted R-squared: 0.7695   
## F-statistic: 8.789 on 6 and 8 DF, p-value: 0.003616

o4<- update(o3,.~.-I(pack^2));summary(o4)

##   
## Call:  
## lm(formula = odor ~ temp + gas + pack + I(temp^2) + I(gas^2),   
## data = odor)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -17.933 -9.635 -4.067 4.620 26.933   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -26.923 8.707 -3.092 0.012884 \*   
## temp -12.125 6.408 -1.892 0.091024 .   
## gas -17.000 6.408 -2.653 0.026350 \*   
## pack -21.375 6.408 -3.336 0.008720 \*\*   
## I(temp^2) 31.615 9.404 3.362 0.008366 \*\*   
## I(gas^2) 47.365 9.404 5.036 0.000703 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 18.12 on 9 degrees of freedom  
## Multiple R-squared: 0.8619, Adjusted R-squared: 0.7852   
## F-statistic: 11.23 on 5 and 9 DF, p-value: 0.001169

o5<- update(o4,.~.-temp) ;summary(o5)

##   
## Call:  
## lm(formula = odor ~ gas + pack + I(temp^2) + I(gas^2), data = odor)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -30.058 -8.572 -3.058 9.812 31.933   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -26.923 9.766 -2.757 0.02024 \*   
## gas -17.000 7.187 -2.365 0.03960 \*   
## pack -21.375 7.187 -2.974 0.01395 \*   
## I(temp^2) 31.615 10.548 2.997 0.01341 \*   
## I(gas^2) 47.365 10.548 4.490 0.00116 \*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 20.33 on 10 degrees of freedom  
## Multiple R-squared: 0.807, Adjusted R-squared: 0.7297   
## F-statistic: 10.45 on 4 and 10 DF, p-value: 0.00135

#best model based on backward elimination   
be<- paste((summary(o5))$call, "")

The best model based using the backwards elimination is lm , odor ~ gas + pack + I(temp^2) + I(gas^2) , odor . Where odor is the response and gas, pack, temp^2 and gas^2 are predictors.

#### (C)

aic<- step(mod, direction = "backward")

## Start: AIC=96.9  
## odor ~ temp + gas + pack + I(temp^2) + I(gas^2) + I(pack^2) +   
## I(temp \* gas) + I(temp \* pack) + I(pack \* gas)  
##   
## Df Sum of Sq RSS AIC  
## - I(temp \* pack) 1 9.0 2535.4 94.951  
## - I(pack \* gas) 1 12.2 2538.7 94.970  
## - I(pack^2) 1 136.6 2663.1 95.688  
## - I(temp \* gas) 1 272.3 2798.7 96.433  
## <none> 2526.4 96.898  
## - temp 1 1176.1 3702.5 100.631  
## - gas 1 2312.0 4838.4 104.644  
## - pack 1 3655.1 6181.5 108.319  
## - I(temp^2) 1 3800.6 6327.1 108.668  
## - I(gas^2) 1 8448.1 10974.5 116.929  
##   
## Step: AIC=94.95  
## odor ~ temp + gas + pack + I(temp^2) + I(gas^2) + I(pack^2) +   
## I(temp \* gas) + I(pack \* gas)  
##   
## Df Sum of Sq RSS AIC  
## - I(pack \* gas) 1 12.2 2547.7 93.023  
## - I(pack^2) 1 136.6 2672.1 93.738  
## - I(temp \* gas) 1 272.3 2807.7 94.481  
## <none> 2535.4 94.951  
## - temp 1 1176.1 3711.5 98.667  
## - gas 1 2312.0 4847.4 102.672  
## - pack 1 3655.1 6190.5 106.341  
## - I(temp^2) 1 3800.6 6336.1 106.689  
## - I(gas^2) 1 8448.1 10983.5 114.942  
##   
## Step: AIC=93.02  
## odor ~ temp + gas + pack + I(temp^2) + I(gas^2) + I(pack^2) +   
## I(temp \* gas)  
##   
## Df Sum of Sq RSS AIC  
## - I(pack^2) 1 136.6 2684.3 91.807  
## - I(temp \* gas) 1 272.3 2819.9 92.546  
## <none> 2547.7 93.023  
## - temp 1 1176.1 3723.8 96.717  
## - gas 1 2312.0 4859.7 100.710  
## - pack 1 3655.1 6202.8 104.371  
## - I(temp^2) 1 3800.6 6348.3 104.718  
## - I(gas^2) 1 8448.1 10995.8 112.958  
##   
## Step: AIC=91.81  
## odor ~ temp + gas + pack + I(temp^2) + I(gas^2) + I(temp \* gas)  
##   
## Df Sum of Sq RSS AIC  
## - I(temp \* gas) 1 272.3 2956.6 91.256  
## <none> 2684.3 91.807  
## - temp 1 1176.1 3860.4 95.257  
## - gas 1 2312.0 4996.3 99.126  
## - pack 1 3655.1 6339.4 102.697  
## - I(temp^2) 1 3712.5 6396.9 102.833  
## - I(gas^2) 1 8332.9 11017.2 110.987  
##   
## Step: AIC=91.26  
## odor ~ temp + gas + pack + I(temp^2) + I(gas^2)  
##   
## Df Sum of Sq RSS AIC  
## <none> 2956.6 91.256  
## - temp 1 1176.1 4132.7 94.279  
## - gas 1 2312.0 5268.6 97.922  
## - pack 1 3655.1 6611.7 101.328  
## - I(temp^2) 1 3712.5 6669.1 101.458  
## - I(gas^2) 1 8332.9 11289.5 109.354

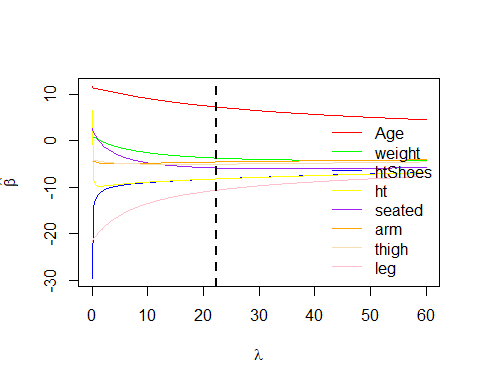
aic$coefficients

## (Intercept) temp gas pack I(temp^2) I(gas^2)   
## -26.92308 -12.12500 -17.00000 -21.37500 31.61538 47.36538

Using the backwards elimination in the Step function. The best model gives odor as the response and temp, gas, pack, temp^2 and gas^2 are predictors. Using calculus, the model is optimized when temp = -5.2149113 and gas = -5.5723976.

### Question 3(Exercise 11.3 in LMR)

lambda<- seq(0,60, length.out = 200)  
rMod<- lm.ridge(hipcenter~. , data = seatpos, lambda= lambda)  
#summary(rMod)  
  
plot(lambda, 0 \* lambda, type="n", xlab=expression(lambda), ylab=expression(hat(beta)), ylim=range(rMod$coef))  
  
#assigning colors to the variables in the plot.  
col<- c("red", "green", "blue", "yellow", "purple", "orange", "wheat", "pink")  
  
for(j in 1:8) {  
 lines(lambda, rMod$coef[j,], col=col[j])  
}  
legend("bottomright", legend = c("Age", "weight", "htShoes", "ht", "seated", "arm", "thigh", "leg"),col=col, lty=rep(1, by = 8), bty = "n")  
abline(v=lambda[which.min(rMod$GCV)], lwd=2, lty = 2, col = "black" )



pred<- coef(rMod)[which.min(rMod$GCV),-1]  
  
#values from question 1  
po = matrix ( c (64.800, 263.700, 181.080, 178.560, 91.440, 35.640, 40.950, 38.790 ) , nrow =1)  
  
#prediction using values from question 1  
pre<- 403.0148+ t(as.matrix(pred))%\*%t(po)

Prediction using the information from question 11.1, give a prediction values of -194.5742785.

### Question 4 (Simulation question )

#### (A)

x<- matrix(rnorm(10\* 100, 0, 1), ncol = 10)  
  
#subset matrix to first 3  
x\_<- x[,1:3]   
   
e<- matrix(rnorm(100, 0, 1), nrow = 100)  
  
y<-0 + 1\*x[,1] + 2\* x[,2] + 3\* x[,3] + e  
  
#fitting model to least square  
tr<- summary(lm(y~x\_))$sigma

The true is 1.0092516.

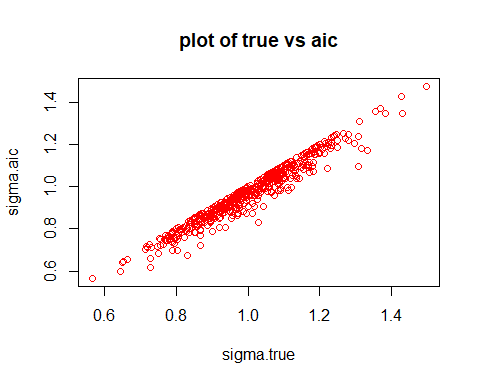
#### (B)

# This function takes the response y and the predictor matrix x and preform the subset and return the model sigma square with the lowest aic  
  
sim<- function(x,y){  
   
 #data frame to hold the response and predictors  
 df<- data.frame(y, x)  
 oo<- regsubsets(y~., data = df, nvmax = 10)  
 oo.s<- summary(oo)  
  
 #aic cacluation for the 11  
 aic <- nrow(df)\* log(oo.s$rss) + 2 \* (2:11)  
  
 #adding the the index to the coresssponding AIC  
 temp<- matrix(nrow=10, ncol = 2)  
 for( i in 1:length(aic)){  
 temp[i,1]= aic[i]  
 temp[i,2]= i  
   
 }  
   
 index<- temp[temp[,1]== min(temp[,1]), ]  
   
 #identify the model with the lowest AIC then store model in temp2  
 temp2<- (oo.s$which)[index[2],]  
   
 #construct string with best model  
 str = "lm(y~"  
   
 co<- 0  
 for(p in 1:length(temp2)){  
 if(p > 1){  
 k = p -1  
 if(temp2[p]== TRUE){  
 co = co + 1  
 if(co == 1){  
 str<- append(str, paste("X", k, sep =""))  
 }else{  
 str<- append(str, paste("+X", k, sep =""))  
 }  
 }  
 if(p == length(temp2)){  
 str<- append(str, paste(",data = df)", sep =""))  
 }  
 }  
 }  
   
 #concatination of model into one string   
 t<- paste(str, collapse = " ")  
   
 #return sigma from model with lowest AIC  
 return((summary(eval(parse(text=t)))$sigma)^2)  
}  
#  
  
aic\_sig<- sim(x=x, y = y)

The aic is 1.0185888 from the best model with the lowest AIC. While the true is 1.0092516.

#### (C)

sigma.true<- as.double(500)  
sigma.aic<- as.double(500)  
  
for(m in 1:500){  
 # simulation of x  
 x<- matrix(rnorm(10\* 100, 0, 1), ncol = 10)  
 #subset x to first 3 rows  
 x\_<- x[,1:3]   
 #simulation of error  
 e<- matrix(rnorm(100, 0, 1), nrow = 100)  
 #simulation of y  
 y<-0 + 1\*x[,1] + 2\* x[,2] + 3\* x[,3] + e  
   
 #storing true values  
 sigma.true[m]<- (summary(lm(y~x\_))$sigma)^2  
   
 #storing aic  
 sigma.aic[m]<- sim(x=x, y = y)  
   
   
}  
  
df<- data.frame(sigma.aic, sigma.true)  
plot(sigma.true,sigma.aic, col = "red", main = "plot of true vs aic")



Most of the points are cluster between .8 and 1.3, points to aic being biased.

#### (D)

df<- data.frame(sigma.true, sigma.aic, sigma.true- sigma.aic)  
  
#%percentage that contian the true sigma  
prop<- length(df$sigma.true...sigma.aic[(df$sigma.true...sigma.aic== 0)])/ length(df)

I think this scatter plot from part c confirm that the aic are biased. We see that 43.33% of the simulation aic is identical to the true . While the other points are not unbiased. Concluding that the simulation proves the statement of the problems is correct.