HW4 - CV

Tanuj Guha

2023-04-03

Load the data.

library (Stat2Data)  
data ("CountyHealth")  
names (CountyHealth)

## [1] "County" "MDs" "Hospitals" "Beds"

head (CountyHealth)

## County MDs Hospitals Beds  
## 1 Bay, FL 351 3 605  
## 2 Beaufort, NC 95 2 134  
## 3 Beaver, PA 260 2 567  
## 4 Bernalillo, NM 2797 11 1435  
## 5 Bibb, GA 769 5 976  
## 6 Clinton, PA 42 2 245

dim (CountyHealth)

## [1] 53 4

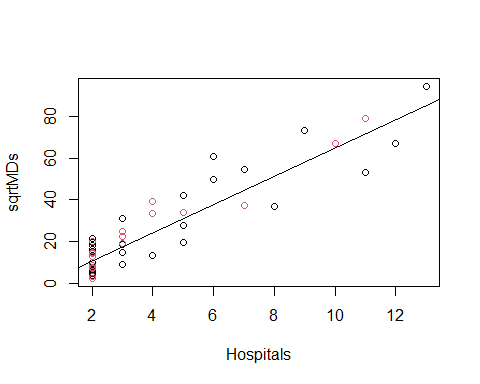
CountyHealth$sqrtMDs = sqrt (CountyHealth$MDs)

4.8.a

# Split the data, per the book exercise.  
  
train.CH = CountyHealth [1:35, ]  
test.CH = CountyHealth [36:53, ]  
  
# Set up a vector for coloring the training set black and the test set red.  
  
train.test = c (rep (1, 35), rep (2, 18))  
  
plot (sqrtMDs ~ Hospitals, data=CountyHealth, col=train.test)  
  
# Fit model on the training sample  
  
train.fit = lm (sqrtMDs ~ Hospitals, data=train.CH)  
summary (train.fit)

##   
## Call:  
## lm(formula = sqrtMDs ~ Hospitals, data = train.CH)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -18.582 -6.362 -2.918 8.277 23.170   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -3.1695 2.6915 -1.178 0.247   
## Hospitals 6.7853 0.5284 12.841 2.19e-14 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 9.627 on 33 degrees of freedom  
## Multiple R-squared: 0.8332, Adjusted R-squared: 0.8282   
## F-statistic: 164.9 on 1 and 33 DF, p-value: 2.194e-14

plot (sqrtMDs ~ Hospitals, data=CountyHealth, col=train.test)  
abline (train.fit)



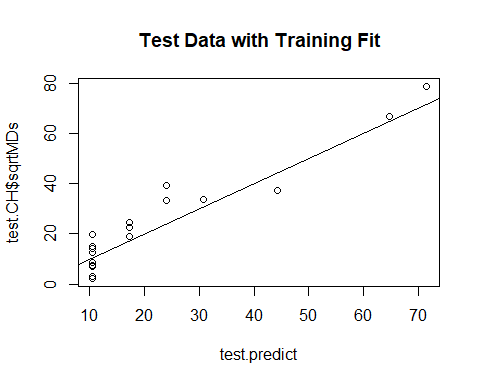
Writing the equation down:

model\_equation <- function(model, ...) {  
 format\_args <- list(...)  
   
 model\_coeff <- model$coefficients  
 format\_args$x <- abs(model$coefficients)  
 model\_coeff\_sign <- sign(model\_coeff)  
 model\_coeff\_prefix <- case\_when(model\_coeff\_sign == -1 ~ " - ",  
 model\_coeff\_sign == 1 ~ " + ",  
 model\_coeff\_sign == 0 ~ " + ")  
 model\_eqn <- paste(strsplit(as.character(model$call$formula), "~")[[2]], # 'y'  
 "=",  
 paste(if\_else(model\_coeff[1]<0, "- ", ""),  
 do.call(format, format\_args)[1],  
 paste(model\_coeff\_prefix[-1],  
 do.call(format, format\_args)[-1],  
 " \* ",  
 names(model\_coeff[-1]),  
 sep = "", collapse = ""),  
 sep = ""))  
 return(model\_eqn)  
}  
  
model\_equation(train.fit, digits = 4)

## [1] "sqrtMDs = - 3.170 + 6.785 \* Hospitals"

4.8.b

# Predictions on the testing sample  
  
test.predict = predict (train.fit, list (Hospitals = test.CH$Hospitals))  
plot (test.CH$sqrtMDs ~ test.predict, main="Test Data with Training Fit")  
abline (0, 1)



# Cross-validation correlation and R-squared  
  
(crval.cor = cor (test.CH$sqrtMDs, test.predict))

## [1] 0.9531439

(test.Rsq = crval.cor^2)

## [1] 0.9084832

The cor() function, in this case test the correlation between the predicted data, and the actual sqrtMDs from the corresponding values in the independent variable. The R^2 value is pretty high, which indicates that the model was fit well using the training set, as seen when the model was run on the test set.

4.8.c

# Shrinkage  
  
summary (train.fit)$r.squared - test.Rsq

## [1] -0.0752451

In this case, the test R^2 is greater than the training R^2. Somehow, the model predicts the data better than the model was fit to the training set. However, given the magnitude of difference, I would not personally suspect any cherry picking of the test set.