Part-1.R

ae7le

2024-12-12

data = read.csv("C:/Users/ae7le/OneDrive/Documents/Sara Schenirer/Intro to Data Science/Final\_Project/auto-mpg.csv", header=TRUE)  
  
# Check if numeric fields are all numeric  
is.numeric(data$mpg)

## [1] TRUE

is.numeric(data$cylinder)

## [1] TRUE

is.numeric(data$displacement)

## [1] TRUE

is.numeric(data$horsepower)

## [1] FALSE

is.numeric(data$weight)

## [1] TRUE

is.numeric(data$model.year)

## [1] TRUE

is.numeric(data$origin)

## [1] TRUE

# Delete rows with non-numeric data in horsepower and convert data from string to int  
data <- data[!is.na(as.numeric(data$horsepower)), ]

## Warning in `[.data.frame`(data, !is.na(as.numeric(data$horsepower)), ): NAs  
## introduced by coercion

data$horsepower <- as.numeric(data$horsepower)  
is.numeric(data$horsepower)

## [1] TRUE

# Split data into train and test  
nrow(data)

## [1] 392

data\_train <- head(data, 300)  
data\_test <- tail(data, 92)  
  
# Create a full, forwards and backwards models  
library(MASS)  
full\_model <- lm(mpg ~ . - car.name, data=data\_train)  
summary(full\_model)

##   
## Call:  
## lm(formula = mpg ~ . - car.name, data = data\_train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -9.2458 -1.6491 0.0753 1.6162 13.4896   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 4.870896 4.966009 0.981 0.32748   
## cylinder -0.419028 0.298327 -1.405 0.16121   
## displacement 0.009598 0.006767 1.418 0.15717   
## horsepower -0.017772 0.012131 -1.465 0.14399   
## weight -0.005373 0.000573 -9.378 < 2e-16 \*\*\*  
## acceleration -0.035080 0.093845 -0.374 0.70882   
## model.year 0.461178 0.060713 7.596 4.16e-13 \*\*\*  
## origin 0.926860 0.300230 3.087 0.00221 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 2.706 on 292 degrees of freedom  
## Multiple R-squared: 0.8216, Adjusted R-squared: 0.8173   
## F-statistic: 192.1 on 7 and 292 DF, p-value: < 2.2e-16

forward\_model <- stepAIC(full\_model, direction = "forward", scope = formula(~ .))

## Start: AIC=605.16  
## mpg ~ (cylinder + displacement + horsepower + weight + acceleration +   
## model.year + origin + car.name) - car.name

summary(forward\_model)

##   
## Call:  
## lm(formula = mpg ~ (cylinder + displacement + horsepower + weight +   
## acceleration + model.year + origin + car.name) - car.name,   
## data = data\_train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -9.2458 -1.6491 0.0753 1.6162 13.4896   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 4.870896 4.966009 0.981 0.32748   
## cylinder -0.419028 0.298327 -1.405 0.16121   
## displacement 0.009598 0.006767 1.418 0.15717   
## horsepower -0.017772 0.012131 -1.465 0.14399   
## weight -0.005373 0.000573 -9.378 < 2e-16 \*\*\*  
## acceleration -0.035080 0.093845 -0.374 0.70882   
## model.year 0.461178 0.060713 7.596 4.16e-13 \*\*\*  
## origin 0.926860 0.300230 3.087 0.00221 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 2.706 on 292 degrees of freedom  
## Multiple R-squared: 0.8216, Adjusted R-squared: 0.8173   
## F-statistic: 192.1 on 7 and 292 DF, p-value: < 2.2e-16

backwards\_model <- stepAIC(full\_model, direction = "backward")

## Start: AIC=605.16  
## mpg ~ (cylinder + displacement + horsepower + weight + acceleration +   
## model.year + origin + car.name) - car.name  
##   
## Df Sum of Sq RSS AIC  
## - acceleration 1 1.02 2139.0 603.30  
## <none> 2138.0 605.16  
## - cylinder 1 14.45 2152.5 605.18  
## - displacement 1 14.73 2152.8 605.22  
## - horsepower 1 15.72 2153.7 605.35  
## - origin 1 69.78 2207.8 612.79  
## - model.year 1 422.47 2560.5 657.25  
## - weight 1 643.96 2782.0 682.14  
##   
## Step: AIC=603.3  
## mpg ~ cylinder + displacement + horsepower + weight + model.year +   
## origin  
##   
## Df Sum of Sq RSS AIC  
## - cylinder 1 14.03 2153.1 603.26  
## <none> 2139.0 603.30  
## - displacement 1 15.87 2154.9 603.52  
## - horsepower 1 17.30 2156.3 603.72  
## - origin 1 70.21 2209.2 610.99  
## - model.year 1 424.13 2563.2 655.57  
## - weight 1 847.83 2986.9 701.46  
##   
## Step: AIC=603.26  
## mpg ~ displacement + horsepower + weight + model.year + origin  
##   
## Df Sum of Sq RSS AIC  
## - displacement 1 4.02 2157.1 601.82  
## - horsepower 1 14.28 2167.3 603.24  
## <none> 2153.1 603.26  
## - origin 1 68.98 2222.1 610.72  
## - model.year 1 412.90 2566.0 653.89  
## - weight 1 893.44 3046.5 705.39  
##   
## Step: AIC=601.82  
## mpg ~ horsepower + weight + model.year + origin  
##   
## Df Sum of Sq RSS AIC  
## - horsepower 1 10.27 2167.4 601.24  
## <none> 2157.1 601.82  
## - origin 1 67.88 2225.0 609.11  
## - model.year 1 409.28 2566.4 651.94  
## - weight 1 1286.27 3443.4 740.13  
##   
## Step: AIC=601.24  
## mpg ~ weight + model.year + origin  
##   
## Df Sum of Sq RSS AIC  
## <none> 2167.4 601.24  
## - origin 1 61.7 2229.0 607.66  
## - model.year 1 522.1 2689.4 663.99  
## - weight 1 4761.9 6929.3 947.92

summary(backwards\_model)

##   
## Call:  
## lm(formula = mpg ~ weight + model.year + origin, data = data\_train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -9.1293 -1.6271 0.0525 1.6617 13.5735   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2.260259 4.396926 0.514 0.60760   
## weight -0.005739 0.000225 -25.502 < 2e-16 \*\*\*  
## model.year 0.475641 0.056329 8.444 1.39e-15 \*\*\*  
## origin 0.778639 0.268320 2.902 0.00399 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 2.706 on 296 degrees of freedom  
## Multiple R-squared: 0.8191, Adjusted R-squared: 0.8173   
## F-statistic: 446.9 on 3 and 296 DF, p-value: < 2.2e-16

# Other models based on what was significant in the previous ones  
model1 <- lm(mpg ~ weight + model.year, data=data\_train)  
summary(model1)

##   
## Call:  
## lm(formula = mpg ~ weight + model.year, data = data\_train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -8.3528 -1.6733 0.0694 1.5108 13.5434   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 5.0707696 4.3421861 1.168 0.244   
## weight -0.0061381 0.0001802 -34.065 < 2e-16 \*\*\*  
## model.year 0.4701286 0.0569958 8.248 5.26e-15 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 2.74 on 297 degrees of freedom  
## Multiple R-squared: 0.814, Adjusted R-squared: 0.8127   
## F-statistic: 649.8 on 2 and 297 DF, p-value: < 2.2e-16

model2 <- lm(mpg ~ weight + model.year + origin, data=data\_train)  
summary(model2)

##   
## Call:  
## lm(formula = mpg ~ weight + model.year + origin, data = data\_train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -9.1293 -1.6271 0.0525 1.6617 13.5735   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2.260259 4.396926 0.514 0.60760   
## weight -0.005739 0.000225 -25.502 < 2e-16 \*\*\*  
## model.year 0.475641 0.056329 8.444 1.39e-15 \*\*\*  
## origin 0.778639 0.268320 2.902 0.00399 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 2.706 on 296 degrees of freedom  
## Multiple R-squared: 0.8191, Adjusted R-squared: 0.8173   
## F-statistic: 446.9 on 3 and 296 DF, p-value: < 2.2e-16

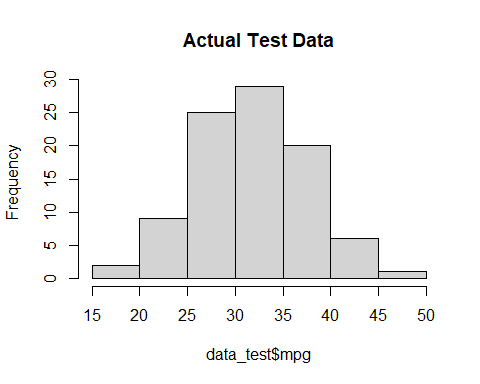
model3 <- lm(mpg ~ weight, data=data\_train)  
summary(model3)

##   
## Call:  
## lm(formula = mpg ~ weight, data = data\_train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -9.2011 -1.9157 -0.0812 1.7341 15.0246   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 40.5619792 0.6461532 62.77 <2e-16 \*\*\*  
## weight -0.0062905 0.0001984 -31.71 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 3.032 on 298 degrees of freedom  
## Multiple R-squared: 0.7714, Adjusted R-squared: 0.7706   
## F-statistic: 1005 on 1 and 298 DF, p-value: < 2.2e-16

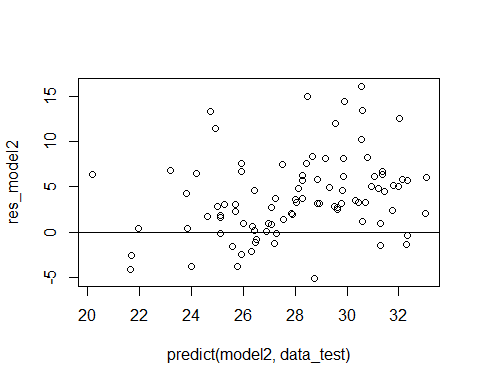
model4 <- lm(mpg ~ model.year, data=data\_train)  
summary(model4)

##   
## Call:  
## lm(formula = mpg ~ model.year, data = data\_train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -8.859 -5.097 -1.517 4.193 19.903   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -28.9942 9.3445 -3.103 0.0021 \*\*   
## model.year 0.6691 0.1254 5.337 1.88e-07 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 6.058 on 298 degrees of freedom  
## Multiple R-squared: 0.08723, Adjusted R-squared: 0.08417   
## F-statistic: 28.48 on 1 and 298 DF, p-value: 1.878e-07

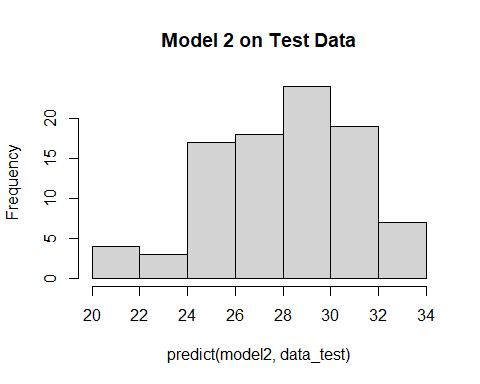
# Make histogram of actual test data  
hist(data\_test$mpg, main = 'Actual Test Data')



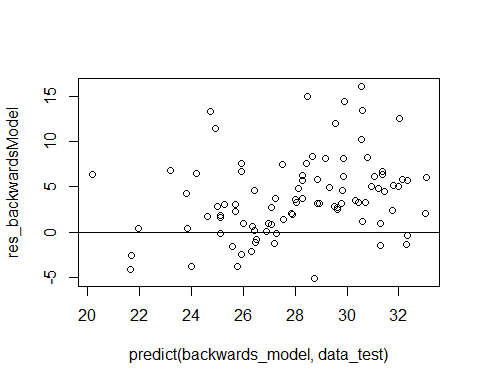
# Test model2 and backwards model on test data by making a histogram of predicted data and a residual plot  
res\_model2 <- data\_test$mpg - predict(model2, data\_test)  
plot(predict(model2, data\_test), res\_model2)  
abline(0,0)



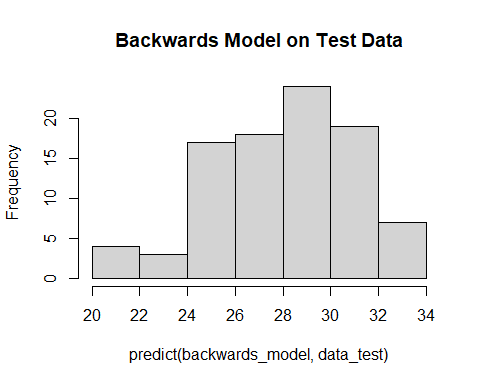
hist(predict(model2, data\_test), main = "Model 2 on Test Data")



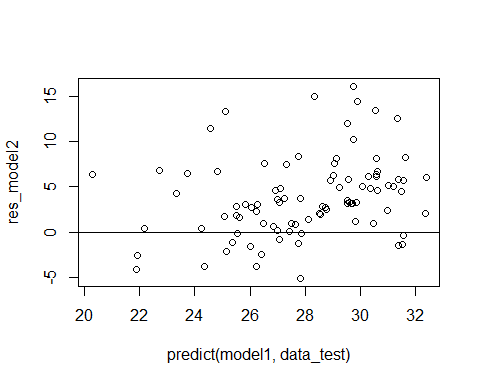
res\_backwardsModel <- data\_test$mpg - predict(backwards\_model, data\_test)  
plot(predict(backwards\_model, data\_test), res\_backwardsModel)  
abline(0,0)



hist(predict(backwards\_model, data\_test), main = "Backwards Model on Test Data")



# Test model1 also because other two tested models were very similar  
res\_model1 <- data\_test$mpg - predict(model1, data\_test)  
plot(predict(model1, data\_test), res\_model2)  
abline(0,0)



hist(predict(model1, data\_test), main = "Model 1 on Test Data")

