Notes 07 - Multiple Linear Regression (MLR)

STS 2300

Updated: 2025-03-22

Table of Contents

[Reading for Notes 07 1](#_Toc193571456)

[Learning Goals for Notes 07 2](#_Toc193571457)

[An example: Predicting Car Mileage 2](#_Toc193571458)

[The Multiple Linear Regression (MLR) Model 3](#_Toc193571459)

[Using R to get an estimated line 4](#_Toc193571460)

[Predictions with MLR Models 5](#_Toc193571461)

[Interpreting MLR Models 6](#_Toc193571462)

[The estimated y-intercept, 6](#_Toc193571463)

[The estimated slope(s) 7](#_Toc193571464)

[Using backwards selection to choose variables for an MLR model 8](#_Toc193571465)

[Considering data context 10](#_Toc193571466)

[P-value review 11](#_Toc193571467)

[Revisiting the Learning Goals for Notes 07 11](#_Toc193571468)

# Reading for Notes 07

Read [Chapter 6](https://moderndive.com/6-multiple-regression.html) of the Modern Dive textbook. You can skip Section 6.1 and 6.3.1 since we won’t cover categorical explanatory variables in this class. Note: You can take STS 2320 Statistical Modeling to learn more about many different forms of regression including ones using categorical explanatory variables. Section 6.1 is also similar to something you might see in STS 3250 Design and Analysis of Experiments.

# Learning Goals for Notes 07

* Be able to use R to generate multiple linear regression equations and to make predictions with them.
* Understand how to interpret the intercept and slope of a multiple linear regression equation in context.
* Be able to appropriately use backward selection to choose a multiple linear regression model.
* Be able to determine and interpret the coefficient of determination () and understand its use and limitations in multiple linear regression.

I’ll be using the following packages in this set of notes, so I’ll load them before I get started.

library(ggplot2)  
library(dplyr)  
library(patchwork)

## Warning: package 'patchwork' was built under R version 4.3.3

library(Lock5Data)

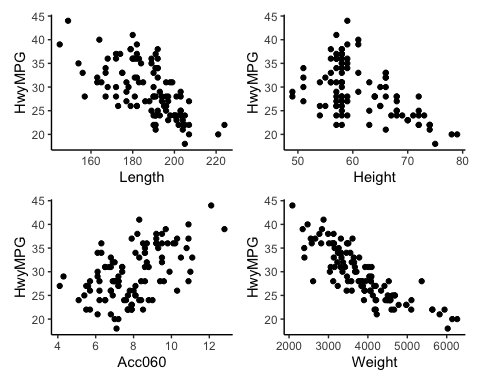
# An example: Predicting Car Mileage

The R package Lock5Data (from another statistics textbook) has a dataset called Cars2015. I’m going to read it into R and call it cars. The dataset consists of 24 variables recorded for 110 car models from 2015. For simplicity, I’m going to focus on how four of the variables might help me predict the highway mileage for a car from 2015.

cars <- Cars2015 |>   
 select(HwyMPG, Length, Height, Acc060, Weight)

Below I’ve used patchwork to see how each of the four possible explanatory variables are related to HwyMPG in the data.

g <- ggplot(cars, aes(y = HwyMPG))  
  
g1 <- g + geom\_point(aes(x = Length)) + theme\_classic()  
g2 <- g + geom\_point(aes(x = Height)) + theme\_classic()  
g3 <- g + geom\_point(aes(x = Acc060)) + theme\_classic()  
g4 <- g + geom\_point(aes(x = Weight)) + theme\_classic()  
  
(g1 + g2) / (g3 + g4)



**Practice**: Describe the relationships that each variable has with HwyMPG. Which have the strongest relationship?

**Answer:**

In simple linear regression (SLR), we used *one* quantitative explanatory variable to help us understand our quantitative response variable. With multiple linear regression (MLR) we can use we use *two or more* explanatory variables to help us understand our quantitative response variable.

# The Multiple Linear Regression (MLR) Model

Recall that in simple linear regression, we wrote the equation for the “true model” as:

and we wanted to estimate the “true y-intercept” () and the “true slope” ().

We can expand on this idea for multiple linear regression by essentially adding new “slopes” for *each* of our explanatory variables. This gets harder to visualize on a single graph, but the idea is the same. Since our example above had four explanatory variables, we could write the equation for the true model we want to estimate as:

Each of the values is an unknown parameter. However, we can make estimates for each of them and calculate an estimated model:

In this equation,

* would be our predicted highway miles per gallon,
* through would be the values for our four explanatory variables,
* and through would be the estimated slopes associated with each variable.

## Using R to get an estimated line

Suppose I have explanatory variables. To get my estimated line using R, my code would look like

lm(response ~ explanatory1 + explanatory2 + ... + explanatoryk, data = dataset)

Let’s try that for the car example.

lm(HwyMPG ~ Length + Height + Acc060 + Weight, data = cars)

##   
## Call:  
## lm(formula = HwyMPG ~ Length + Height + Acc060 + Weight, data = cars)  
##   
## Coefficients:  
## (Intercept) Length Height Acc060 Weight   
## 34.012518 0.074484 -0.172791 1.163281 -0.004477

Notice how in the output I have a number for the intercept and then another number associated with each explanatory variable in my model. Those are my estimated slopes. I could write my equation as:

**Question:** Do these coefficients match the direction we saw in the scatterplots above? If not, why do you think that might be?

**Answer:**

## Predictions with MLR Models

The process for making predictions in a multiple linear regression model is similar to a simple linear regression model, but I’ll need to plug something in for each of my explanatory variables.

The Nissan Pathfinder in 2015 had a length of 192 inches, a height of 72 inches, an acceleration from 0 to 60 mph of 7.7 seconds, and a weight of 4505 pounds. Suppose that we didn’t have data on the highway miles per gallon for this car. I could use my model to make a prediction:

To have R do this calculation for me, my code would be:

cars.lm <- lm(HwyMPG ~ Length + Height + Acc060 + Weight, data = cars)  
predict(cars.lm, newdata = data.frame(Length = 192,   
 Height = 72,   
 Acc060 = 7.7,  
 Weight = 4505))

## 1   
## 24.66199

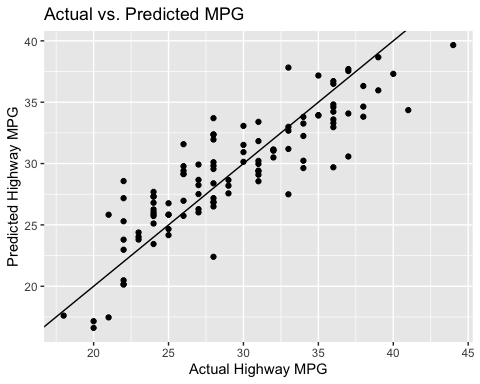
My prediction is that the car would have a highway mileage of 24.66 miles per gallon. The actual value in the dataset was 25 miles per gallon, so it looks like this would have been a good prediction!

Recall that the residual is the distance between the actual value and your prediction. My residual for the Nissan Pathfinder is:

This is also telling me that my observed mileage was just a little bit above my predicted mileage for this car.

One way to decide how well our model is working would be to compare the predicted mileages to the actual mileages.

cars$predictedHwyMPG <- predict(cars.lm)  
ggplot(cars) +  
 geom\_point(aes(x = HwyMPG, y = predictedHwyMPG)) +  
 geom\_abline(slope = 1) +  
 labs(title = "Actual vs. Predicted MPG", x = "Actual Highway MPG",   
 y = "Predicted Highway MPG")



**Question**: What do you notice? What might we wonder? (Hint: Consider how good our predictions are for different type of values for the response variable)

**Answer:**

# Interpreting MLR Models

## The estimated y-intercept,

Recall that in a simple linear regression model, we interpreted as the predicted value of the response variable when the explanatory variable was 0. For multiple linear regression, the only thing we need to change is that is the predicted value of the response variable when *all of the explanatory variables* are 0.

Thus, for the example above where , we could interpret it by saying:

The predicted highway mileage of a car model is 34.01 miles per gallon if that car has a length, height, weight, and acceleration speed (from 0 to 60 mph) of 0.

**Question:** Does this prediction seem like an extrapolation? Why or why not?

**Answer:**

## The estimated slope(s)

In a simple linear regression model we had one slope, , and we interpreted it as the predicted change in our response variable if our explanatory variable went up by one unit. In our multiple linear regression model, we have more than one explanatory variable, so we need to make an addition to our interpretation.

**General interpretation of MLR slope(s)**: The *predicted* response increases/decreases (based on + or - sign) by the absolute value of for every one unit increase in that explanatory variable *assuming all other explanatory variables are held constant*.

In other words, we want to make sure we’re talking about *only* changing that variable. Here is an example using the weight variable.

**Interpretation of weight slope**: The predicted highway mileage decreases by 0.004 miles per gallon for each additional pound a car weighs (assuming the length, height, and acceleration from 0 to 60 mph are held constant for the car).

This make sense that heavier cars would probably get worse gas mileage since they require more energy to move. However, sometimes these interpretations might seem a bit odd.

**Practice**: Try writing an interpretation for the slope for the height variable (-0.173).

**Interpretation:**

**Practice**: Try writing an interpretation for the slope for the length variable (0.074).

**Interpretation:**

**Question**: Do either of your interpretations seem a bit counter intuitive (or to not match our graphs from before)? Why do you think that is?

**Answer:**

# Using backwards selection to choose variables for an MLR model

Recall that the value tells us the proportion of the variability in our response variable that can be explained by our explanatory variables. It might seem like we should choose the model that has the highest value, but doing this would lead us to always add as many variables as possible to our model. That’s because adding a new variable can never cause us to explain *less* of the variability. However, a larger model is harder to interpret and may lead to us “over-fitting” our data, which would result in worse predictions.

An alternative approach is called **backward selection** (or backward elimination). With this approach, we put all of our explanatory variables in our model and then remove them one at a time until we have a model where all of the explanatory variables have estimated slopes with statistically significant p-values (according to a pre-chosen cutoff).

**Note #1:** Unfamiliar with p-values or want a refresher? See the section about p-values at the end of this set of notes.

**Note #2:** The **adjusted**  value partially addresses this concern by adjusting the value based on the number of variables in the model. Some people may also use this (or a wide variety of other methods) to choose when to stop removing variables from their model.

**Note #3:** You should not blindly apply selection methods such as backwards selection without thinking about your data first. It is better to have sound justification for whether a variable should be in the model than to simply let a selection method choose whether it should be included. Selection methods are intrinsically data-driven, meaning a different sample/dataset may produce different results.

Let’s add a few extra variables to our car example from above. (Note: We must use the dplyr pipe %>% because the data argument in lm() is not the first argument and the base pipe |> does not support this method.)

Cars2015 %>%  
 lm(HwyMPG ~ Length + Width + Height + Weight + Acc030 + Acc060, data = .) %>%  
 summary()

##   
## Call:  
## lm(formula = HwyMPG ~ Length + Width + Height + Weight + Acc030 +   
## Acc060, data = .)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -6.4063 -1.8025 0.2234 1.5775 6.5662   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 26.188849 11.671167 2.244 0.026979 \*   
## Length 0.055310 0.038116 1.451 0.149792   
## Width 0.171429 0.187356 0.915 0.362332   
## Height -0.203240 0.103559 -1.963 0.052395 .   
## Weight -0.004628 0.001162 -3.981 0.000128 \*\*\*  
## Acc030 1.146686 1.738659 0.660 0.511032 <- remove this one   
## Acc060 0.884064 0.612098 1.444 0.151685   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 2.803 on 103 degrees of freedom  
## Multiple R-squared: 0.7578, Adjusted R-squared: 0.7437   
## F-statistic: 53.71 on 6 and 103 DF, p-value: < 2.2e-16

We can see that several of our variables do **not** have statistically significant p-values. This includes some variables that were statistically significant in our smaller model earlier. The largest p-value (0.5110) belongs to the Acc030 variable.

**Backward selection approach**: Once we have a model with all explanatory variables, we will check for slopes that have p-values above 0.05 (or 0.1 or some other pre-chosen value). We will remove the variable associated with the largest of these p-values. Then we will re-check p-values for the slopes that remain. We will repeat this process until all of our slopes have p-values below our chosen cutoff.

Try removing the variable with the largest p-value and refitting the model.

Cars2015 %>%  
 lm(HwyMPG ~ Length + Width + Height + Weight + Acc060, data = .) %>%  
 summary()

##   
## Call:  
## lm(formula = HwyMPG ~ Length + Width + Height + Weight + Acc060,   
## data = .)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -6.5437 -1.7861 0.4014 1.5701 6.5510   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 24.404747 11.322459 2.155 0.033435 \*   
## Length 0.065023 0.035060 1.855 0.066485 .   
## Width 0.184425 0.185810 0.993 0.323233   
## Height -0.189566 0.101186 -1.873 0.063816 .   
## Weight -0.004902 0.001083 -4.527 1.59e-05 \*\*\*  
## Acc060 1.229946 0.314774 3.907 0.000166 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 2.796 on 104 degrees of freedom  
## Multiple R-squared: 0.7568, Adjusted R-squared: 0.7451   
## F-statistic: 64.71 on 5 and 104 DF, p-value: < 2.2e-16

Notice how now Acc060 has a **much smaller** p-value.

**Question:** Why did the p-value associated with Acc060 change so much?

**Answer:**

**Question:** Which variable should we try removing next? Why?

**Answer:**

**Practice:** Try removing that variable. Should we stop at this model or remove other variable(s)?

**Answer:**

# Considering data context

**Question**: This data was specifically from 2015 car models. Would it make sense to use it to make predictions for 2024 car models?

**Answer**:

Regardless, we should always be careful when we use data from one population to make predictions in another population. This is a common issue in a lot of artificial intelligence and machine learning (aka fancy black box statistics applications) problems these days. Some examples include [facial recognition software](https://www.ted.com/talks/joy_buolamwini_how_i_m_fighting_bias_in_algorithms) not recognizing people with dark skin because their data mostly included people with lighter skin, [Amazon downgrading female job applicants](https://www.aclu.org/blog/womens-rights/womens-rights-workplace/why-amazons-automated-hiring-tool-discriminated-against) because their past hires were mostly male, and much more.

In many of these cases, people using the models didn’t think they were being biased. They may have even avoided putting factors like race or sex into their model thinking that meant their decisions wouldn’t be impacted by those factors. However, our society is complex and all kinds of variables are related to other variables (like we saw just in a much less consequential example above). This means we really need to be careful about how we interpret and use the models that we create with data.

Other resources if this interests you:

* Book: Weapons of Math Destruction ([NPR Interview](https://www.npr.org/2016/09/12/493654950/weapons-of-math-destruction-outlines-dangers-of-relying-on-data-analytics))
* Book: Data Feminism ([free online version](https://data-feminism.mitpress.mit.edu/))
* Netflix Documentary: Coded Bias ([website](https://www.codedbias.com/))

# P-value review

A p-value is the probability of observing data at least as extreme as what you saw if the null hypothesis were true. This is a notoriously difficult definition to understand. We will spend more time learning about p-values later in the course.

For now, we can think of p-values as a measure of how likely would it be to see a slope value as large (positive or negative) as ours in a world where the explanatory variable has no linear relationship with the response variable (assuming any other explanatory variables remain in the model).

This means that a small p-value (closer to 0) indicates our slope value would be unusual if that explanatory variable weren’t important for our model. In that case, we want to keep that value in our model. Usually we will use 0.05 or 0.1 as our cutoff for what we consider “small.”

A large p-value indicates our slope value doesn’t look much different from what we would expect even if our explanatory variables weren’t really related to our response variable. In that case, we should feel ok removing the variable from our model since we haven’t convinced ourselves that it’s actually helping explain anything.

# Revisiting the Learning Goals for Notes 07

* Be able to use R to generate multiple linear regression equations and to make predictions with them.
  + Use the penguins data from the palmerpenguins package to create a multiple linear regression equation that uses bill length, bill depth, and flipper length to predict body mass.
  + Use your equation to predict the body mass for a penguin with a bill length of 39.1 mm, a bill depth of 17.4 mm, and a flipper length of 180 mm.
* Understand how to interpret the intercept and slope of a multiple linear regression equation in context.
  + Write an interpretation for the intercept of your equation from above
  + Write an interpretation for the flipper length slope for your equation above
* Be able to appropriately use backward selection to choose a multiple linear regression model.
  + Are there any variables that should be removed from the model you created? If so, use backward selection to arrive at a model where all variables are statistically significant at the 0.05 level.
* Be able to determine and interpret the coefficient of determination () and understand its use and limitations in multiple linear regression.
  + Find the value for your final model and interpret it in context.
  + (Bonus) How does the value for your final model compare to your original model with all three variables?