Introduction to linear regression

Adapted from chapter 3 of An Introduction to Statistical Learning

Hicham Zmarrou

2017-09-28

### Motivation

#### Why are we learning linear regression?

* widely used
* runs fast
* easy to use (not a lot of tuning required)
* highly interpretable
* basis for many other methods

### Example: Advertising Data

Let's take a look at some data, ask some questions about that data, and then use linear regression to answer those questions!

# read data into a DataFrame  
suppressWarnings(suppressPackageStartupMessages(library(tidyverse)))  
suppressWarnings(suppressPackageStartupMessages(library(ISLR)))  
suppressWarnings(suppressPackageStartupMessages(library(MASS)))  
Advertising<- read\_csv('http://www-bcf.usc.edu/~gareth/ISL/Advertising.csv')

## Warning: Missing column names filled in: 'X1' [1]

## Parsed with column specification:  
## cols(  
## X1 = col\_integer(),  
## TV = col\_double(),  
## radio = col\_double(),  
## newspaper = col\_double(),  
## sales = col\_double()  
## )

Advertising[,1] <- NULL   
Advertising

## # A tibble: 200 x 4  
## TV radio newspaper sales  
## <dbl> <dbl> <dbl> <dbl>  
## 1 230.1 37.8 69.2 22.1  
## 2 44.5 39.3 45.1 10.4  
## 3 17.2 45.9 69.3 9.3  
## 4 151.5 41.3 58.5 18.5  
## 5 180.8 10.8 58.4 12.9  
## 6 8.7 48.9 75.0 7.2  
## 7 57.5 32.8 23.5 11.8  
## 8 120.2 19.6 11.6 13.2  
## 9 8.6 2.1 1.0 4.8  
## 10 199.8 2.6 21.2 10.6  
## # ... with 190 more rows

Advertisingg <- gather(Advertising, medium, spending,TV:newspaper)

#### What are the features?

TV: advertising dollars spent on TV for a single product in a given market (in thousands of dollars) Radio: advertising dollars spent on Radio Newspaper: advertising dollars spent on Newspaper

#### What is the response?

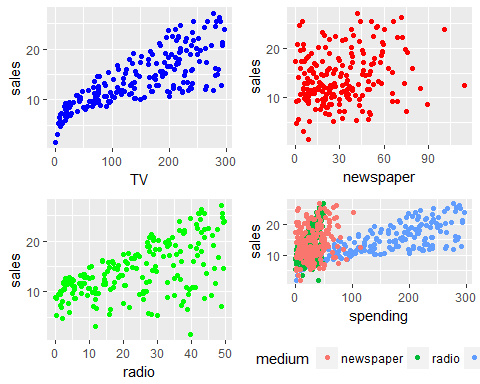
Sales: sales of a single product in a given market (in thousands of widgets)

# print the shape of the DataFrame  
sizeAdvertising= c(nrow(Advertising),ncol(Advertising))   
sizeAdvertising

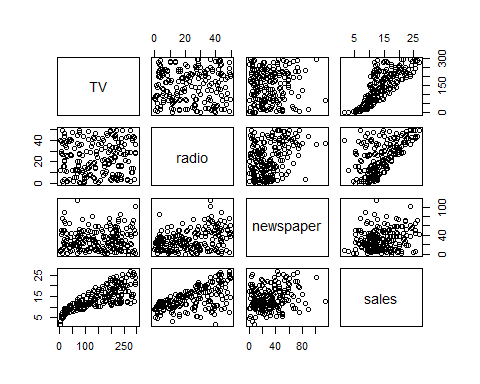
## [1] 200 4

There are 200 observations, and thus 200 markets in the dataset.

# visualize the relationship between the features and the response using scatterplots  
source("multiplot.R")  
p1<- ggplot(data = Advertising) + geom\_point(mapping = aes(x = TV, y = sales),color = "blue")  
p2<- ggplot(data = Advertising) + geom\_point(mapping = aes(x = radio, y = sales),color = "green")  
p3<- ggplot(data = Advertising) + geom\_point(mapping = aes(x = newspaper, y = sales),color = "red")  
p4<- ggplot(data = Advertisingg )+ geom\_point(mapping = aes(x = spending, y = sales, color = medium))+theme(legend.position = "bottom")  
  
multiplot(p1, p2, p3, p4, cols=2)



pairs(Advertising)



#### Questions About the Advertising Data

Let's pretend you work for the company that manufactures and markets this widget. The company might ask you the following: On the basis of this data, how should we spend our advertising money in the future? This general question might lead you to more specific questions: Is there a relationship between ads and sales? How strong is that relationship? Which ad types contribute to sales? What is the effect of each ad type of sales? Given ad spending in a particular market, can sales be predicted? We will explore these questions below!

#### Simple Linear Regression

Simple linear regression is an approach for predicting a quantitative response using a single feature (or "predictor" or "input variable"). It takes the following form:

With oth words it says what is the expected valye of given a realisation of mathematically

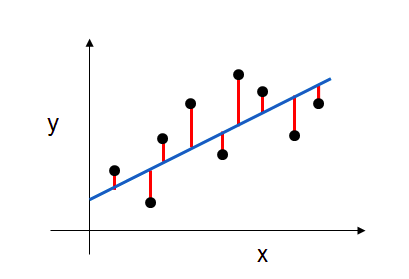
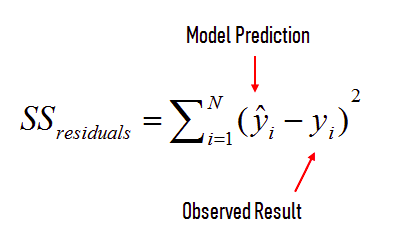
What does each term represent?

* is the response
* is the feature/predictor
* is the intercept
* is the coefficient for x

Together, and are called the model coefficients. To create your model, you must "learn" the values of these coefficients. And once we've learned these coefficients, we can use the model to predict Sales!

### Estimating ("Learning") Model Coefficients

Generally speaking, coefficients are estimated using the least squares criterion, which means we are find the line (mathematically) which minimizes the sum of squared residuals (or "sum of squared errors"):

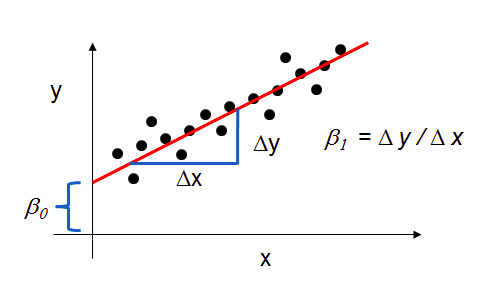
#### What elements are present in the diagram?

* The black dots are the observed values of x and y.
* The blue line is our least squares line.
* The red lines are the residuals, which are the distances between the observed values and the least squares line.

#### How do the model coefficients relate to the least squares line?

* is the intercept (the value of when =0)
* is the slope (the change in divided by change in )

Here is a graphical depiction of those calculations:



Let's use lm function ind the base R to estimate the model coefficients for the advertising data:

# create a fitted model in one line  
library(knitr)  
lrmod <- lm(sales ~ TV, data= Advertising)  
summary(lrmod)

Call: lm(formula = sales ~ TV, data = Advertising)

Residuals: Min 1Q Median 3Q Max -8.3860 -1.9545 -0.1913 2.0671 7.2124

Coefficients: Estimate Std. Error t value Pr(>|t|)  
(Intercept) 7.032594 0.457843 15.36 <2e-16  ***TV 0.047537 0.002691 17.67 <2e-16***  --- Signif. codes: 0 '***' 0.001 '****' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.259 on 198 degrees of freedom Multiple R-squared: 0.6119, Adjusted R-squared: 0.6099 F-statistic: 312.1 on 1 and 198 DF, p-value: < 2.2e-16

we can also save our model in a text file

sink("lrmod.txt")  
lm(sales ~ TV, data= Advertising)

##   
## Call:  
## lm(formula = sales ~ TV, data = Advertising)  
##   
## Coefficients:  
## (Intercept) TV   
## 7.03259 0.04754

sink()

#### Interpreting Model Coefficients

How do we interpret the TV coefficient ()?

A "unit" increase in TV ad spending is associated with a 0.047537 "unit" increase in Sales. Or more clearly: An additional $1,000 spent on TV ads is associated with an increase in sales of 47.537 widgets.

Note that if an increase in TV ad spending was associated with a decrease in sales, would be negative.

#### Using the Model for Prediction

Let's say that there was a new market where the TV advertising spend was $50,000. What would we predict for the Sales in that market?

# manually calculate the prediction  
7.032594 + 0.047537\*50

## [1] 9.409444

Thus, we would predict Sales of 9,409 widgets in that market. Of course, we can also the lm function to make the prediction:

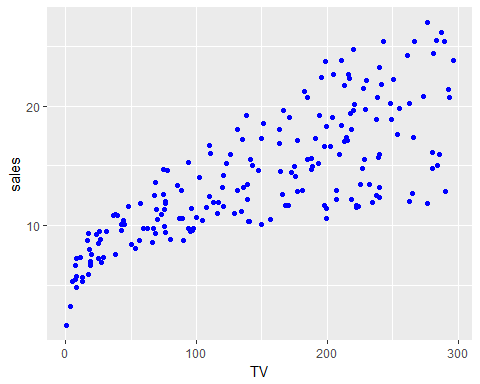
predict(lrmod, data.frame(TV = c(50,75,125)))

## 1 2 3   
## 9.409426 10.597842 12.974674

#### Plotting the Least Squares Line

Let's make predictions for the smallest and largest observed values of x, and then use the predicted values to plot the least squares line

ggplot() + geom\_point(data = Advertising,aes(TV,sales), color = "blue")



#### Plotting the Least Squares Line¶

Let's make predictions for the smallest and largest observed values of x, and then use the predicted values to plot the least squares line

#### Confidence in our Model

Question: Is linear regression a high bias/low variance model, or a low bias/high variance model?

Answer: High bias/low variance. Under repeated sampling, the line will stay roughly in the same place (low variance), but the average of those models won't do a great job capturing the true relationship (high bias). Note that low variance is a useful characteristic when you don't have a lot of training data!

A closely related concept is confidence intervals. lm calculates 95% confidence intervals for our model coefficients, which are interpreted as follows: If the population from which this sample was drawn was sampled 100 times, approximately 95 of those confidence intervals would contain the "true" coefficient.

confint(lrmod)

## 2.5 % 97.5 %  
## (Intercept) 6.12971927 7.93546783  
## TV 0.04223072 0.05284256

Keep in mind that we only have a single sample of data, and not the entire population of data. The "true" coefficient is either within this interval or it isn't, but there's no way to actually know. We estimate the coefficient with the data we do have, and we show uncertainty about that estimate by giving a range that the coefficient is probably within. Note that using 95% confidence intervals is just a convention. You can create 90% confidence intervals (which will be more narrow), 99% confidence intervals (which will be wider), or whatever intervals you like.

#### Hypothesis Testing and p-values

Closely related to confidence intervals is hypothesis testing. Generally speaking, you start with a null hypothesis and an alternative hypothesis (that is opposite the null). Then, you check whether the data supports rejecting the null hypothesis or failing to reject the null hypothesis.

(Note that "failing to reject" the null is not the same as "accepting" the null hypothesis. The alternative hypothesis may indeed be true, except that you just don't have enough data to show that.) As it relates to model coefficients, here is the conventional hypothesis test: null hypothesis: There is no relationship between TV ads and Sales (and thus equals zero) alternative hypothesis: There is a relationship between TV ads and Sales (and thus is not equal to zero)

Null hypothesis : There is no relationship between and .

Alternative hypothesis : There is some relationship between and .

This is mathematically equivalent to

How do we test this hypothesis? Intuitively, we reject the null (and thus believe the alternative) if the 95% confidence interval does not include zero. Conversely, the p-value represents the probability that the coefficient is actually zero:

We use the broom package to convert the messy output of built-in functions in R, such as lm, nls, or t.test, and turns them into tidy data frames.

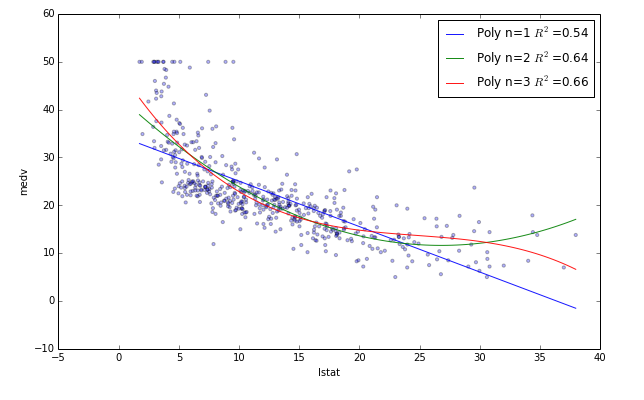
library(broom)  
# tidyres <-tidy(lrmod)  
tidy(lrmod) %>% .$p.value

## [1] 1.40630e-35 1.46739e-42

If the 95% confidence interval includes zero, the p-value for that coefficient will be greater than 0.05. If the 95% confidence interval does not include zero, the p-value will be less than 0.05. Thus, a p-value less than 0.05 is one way to decide whether there is likely a relationship between the feature and the response. (Again, using 0.05 as the cutoff is just a convention.) In this case, the p-value for TV is far less than 0.05, and so we believe that there is a relationship between TV ads and Sales. Note that we generally ignore the p-value for the intercept.

#### How Well Does the Model Fit the data?

The most common way to evaluate the overall fit of a linear model is by the R-squared value. R-squared is the proportion of variance explained, meaning the proportion of variance in the observed data that is explained by the model, or the reduction in error over the null model. (The null model just predicts the mean of the observed response, and thus it has an intercept and no slope.)



R-squared is between 0 and 1, and higher is better because it means that more variance is explained by the model. Here's an example of what R-squared "looks like" You can see that the **blue line** explains some of the variance in the data (R-squared=0.54), the **green line** explains more of the variance (R-squared=0.64), and the **red line** fits the training data even further (R-squared=0.66). (Does the red line look like it's overfitting?) Let's calculate the R-squared value for our simple linear model:

Let's calculate the R-squared value for our simple linear model:

rSquared <- glance(lrmod)[1,1]  
rSquared

## [1] 0.6118751

Is that a "good" R-squared value? It's hard to say. The threshold for a good R-squared value depends widely on the domain. Therefore, it's most useful as a tool for comparing different models.

#### Multiple Linear Regression

Simple linear regression can easily be extended to include multiple features. This is called multiple linear regression: Each represents a different feature, and each feature has its own coefficient. In this case: Let's use Statsmodels to estimate these coefficients:

mlrmod <- lm(sales ~ TV + radio + newspaper, data= Advertising)  
summary(mlrmod)

##   
## Call:  
## lm(formula = sales ~ TV + radio + newspaper, data = Advertising)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -8.8277 -0.8908 0.2418 1.1893 2.8292   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2.938889 0.311908 9.422 <2e-16 \*\*\*  
## TV 0.045765 0.001395 32.809 <2e-16 \*\*\*  
## radio 0.188530 0.008611 21.893 <2e-16 \*\*\*  
## newspaper -0.001037 0.005871 -0.177 0.86   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 1.686 on 196 degrees of freedom  
## Multiple R-squared: 0.8972, Adjusted R-squared: 0.8956   
## F-statistic: 570.3 on 3 and 196 DF, p-value: < 2.2e-16

tidy(mlrmod)

## term estimate std.error statistic p.value  
## 1 (Intercept) 2.938889369 0.311908236 9.4222884 1.267295e-17  
## 2 TV 0.045764645 0.001394897 32.8086244 1.509960e-81  
## 3 radio 0.188530017 0.008611234 21.8934961 1.505339e-54  
## 4 newspaper -0.001037493 0.005871010 -0.1767146 8.599151e-01

## What are a few key things we learn from this output?

* TV and Radio have significant p-values, whereas Newspaper does not. Thus we reject the null hypothesis for TV and Radio (that there is no association between those features and Sales), and fail to reject the null hypothesis for Newspaper.
* TV and Radio ad spending are both positively associated with Sales, whereas Newspaper ad spending is slightly negatively associated with Sales. (However, this is irrelevant since we have failed to reject the null hypothesis for Newspaper.)
* This model has a higher R-squared (0.897) than the previous model, which means that this model provides a better fit to the data than a model that only includes TV.

### Feature Selection

How do I decide **which features to include** in a linear model? Here's one idea:

* Try different models, and only keep predictors in the model if they have small p-values.
* Check whether the R-squared value goes up when you add new predictors.

#### What are the drawbacks to this approach?

* Linear models rely upon a lot of assumptions (such as the features being independent), and if those assumptions are violated (which they usually are), R-squared and p-values are less reliable.
* Using a p-value cutoff of 0.05 means that if you add 100 predictors to a model that are pure noise, 5 of them (on average) will still be counted as significant.
* R-squared is susceptible to overfitting, and thus there is no guarantee that a model with a high R-squared value will generalize. Below is an example:

# only include TV and Radio in the model  
options(digits=10)  
mlrmod <- lm(sales ~ TV + radio, data = Advertising)  
rSquared <- glance(lrmod)[1,1]  
rSquared

## [1] 0.6118750509

#summary(mlrmod)

# add Newspaper to the model (which we believe has no association with Sales)  
options(digits=10)  
mlrmod <- lm(sales ~ TV + radio + newspaper, data = Advertising)  
# mlrmod <- lm(Sales ~ ., data = Advertising)  
rSquared <- glance(lrmod)[1,1]  
rSquared

## [1] 0.6118750509

#summary(mlrmod)

R-squared will always increase as you add more features to the model, even if they are unrelated to the response. Thus, selecting the model with the highest R-squared is not a reliable approach for choosing the best linear model.

There is alternative to R-squared called adjusted R-squared that penalizes model complexity (to control for overfitting), but it generally under-penalizes complexity.

So is there a better approach to feature selection? Cross-validation. It provides a more reliable estimate of out-of-sample error, and thus is a better way to choose which of your models will best generalize to out-of-sample data. There is extensive functionality for cross-validation in scikit-learn, including automated methods for searching different sets of parameters and different models. Importantly, cross-validation can be applied to any model, whereas the methods described above only apply to linear models.

### Handling categorical predictors with two categories

Up to now, all of our predictors have been numeric. What if one of our predictors was categorical? Let's create a new feature called Size, and randomly assign observations to be small or large:

# set a seed for reproducibility  
set.seed(12345)  
# create a Series of booleans in which roughly half are True  
nums = runif(nrow(Advertising), 0, 1)   
mask\_large = nums > 0.5   
# initially set Size to small, then change roughly half to be large  
Advertising[['Size']] = rep('small',nrow(Advertising))  
Advertising$Size[mask\_large] = 'large'  
Advertising$Size = as.factor(Advertising$Size)  
#Advertising = Advertising %>% mutate(isLarge = factor(if\_else(Size == "large",1,0))) %>% dplyr::select(-Size)   
Advertising

## # A tibble: 200 x 5  
## TV radio newspaper sales Size  
## <dbl> <dbl> <dbl> <dbl> <fctr>  
## 1 230.1 37.8 69.2 22.1 large  
## 2 44.5 39.3 45.1 10.4 large  
## 3 17.2 45.9 69.3 9.3 large  
## 4 151.5 41.3 58.5 18.5 large  
## 5 180.8 10.8 58.4 12.9 small  
## 6 8.7 48.9 75.0 7.2 small  
## 7 57.5 32.8 23.5 11.8 small  
## 8 120.2 19.6 11.6 13.2 large  
## 9 8.6 2.1 1.0 4.8 large  
## 10 199.8 2.6 21.2 10.6 large  
## # ... with 190 more rows

#levels(Advertising$Size)  
mlrmodc2 <- lm(sales ~ ., data = Advertising)  
summary(mlrmod)

##   
## Call:  
## lm(formula = sales ~ TV + radio + newspaper, data = Advertising)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -8.8276871 -0.8908135 0.2418018 1.1893186 2.8292226   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2.938889369 0.311908236 9.42229 < 2e-16 \*\*\*  
## TV 0.045764645 0.001394897 32.80862 < 2e-16 \*\*\*  
## radio 0.188530017 0.008611234 21.89350 < 2e-16 \*\*\*  
## newspaper -0.001037493 0.005871010 -0.17671 0.85992   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 1.68551 on 196 degrees of freedom  
## Multiple R-squared: 0.8972106, Adjusted R-squared: 0.8956373   
## F-statistic: 570.2707 on 3 and 196 DF, p-value: < 2.2204e-16

How do we interpret the IsLarge coefficient? For a given amount of TV/Radio/Newspaper ad spending, being a small market is associated with an average increase in Sales of 256.56 widgets (as compared to a large market, which is called the baseline level).

### Handling categorical predictors with more than two categories

Let's create a new feature called Area, and randomly assign observations to be rural, suburban, or urban:

# Advertising<- read\_csv('http://www-bcf.usc.edu/~gareth/ISL/Advertising.csv')  
# Advertising[,1] <- NULL   
# set a seed for reproducibility  
nums = runif(nrow(Advertising), 0, 1)   
# initially set Size to small, then change  
mask\_suburban = (nums > 0.33) & (nums < 0.66)  
mask\_urban = nums > 0.66  
Advertising[['Area']]= rep('rural',nrow(Advertising))  
  
Advertising$Area[mask\_suburban] = 'suburban'  
Advertising$Area[mask\_urban] = 'urban'  
Advertising

## # A tibble: 200 x 6  
## TV radio newspaper sales Size Area  
## <dbl> <dbl> <dbl> <dbl> <fctr> <chr>  
## 1 230.1 37.8 69.2 22.1 large suburban  
## 2 44.5 39.3 45.1 10.4 large urban  
## 3 17.2 45.9 69.3 9.3 large rural  
## 4 151.5 41.3 58.5 18.5 large suburban  
## 5 180.8 10.8 58.4 12.9 small urban  
## 6 8.7 48.9 75.0 7.2 small urban  
## 7 57.5 32.8 23.5 11.8 small rural  
## 8 120.2 19.6 11.6 13.2 large rural  
## 9 8.6 2.1 1.0 4.8 large suburban  
## 10 199.8 2.6 21.2 10.6 large suburban  
## # ... with 190 more rows

mlrmodc <- lm(sales ~ ., data = Advertising)  
summary(mlrmodc)

##   
## Call:  
## lm(formula = sales ~ ., data = Advertising)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -9.0260769 -0.7723998 0.3012214 1.1607341 2.9484246   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2.887134714 0.367696960 7.85194 2.7676e-13 \*\*\*  
## TV 0.045616164 0.001444274 31.58415 < 2.22e-16 \*\*\*  
## radio 0.188979764 0.008703531 21.71300 < 2.22e-16 \*\*\*  
## newspaper -0.001455054 0.005922492 -0.24568 0.80619   
## Sizesmall 0.245495212 0.247210898 0.99306 0.32192   
## Areasuburban -0.009424008 0.310997497 -0.03030 0.97586   
## Areaurban -0.075634462 0.301130171 -0.25117 0.80195   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 1.69337 on 193 degrees of freedom  
## Multiple R-squared: 0.8978378, Adjusted R-squared: 0.8946618   
## F-statistic: 282.6922 on 6 and 193 DF, p-value: < 2.2204e-16

#### How do we interpret the coefficients?

Holding all other variables fixed, being a suburban area is associated with an average decrease in Sales of 131.589 widgets (as compared to the baseline level, which is rural). Being an urban area is associated with an average increase in Sales of 157.600 widgets (as compared to rural).

A final note about dummy encoding: If you have categories that can be ranked (i.e., strongly disagree, disagree, neutral, agree, strongly agree), you can potentially use a single dummy variable and represent the categories numerically (such as 1, 2, 3, 4, 5).

### Interactions

Interactions can be introduced to the lm() procedure in a number of ways.

We can use the : operator to introduce a single interaction of interest.

Advertising<- read\_csv('http://www-bcf.usc.edu/~gareth/ISL/Advertising.csv')

## Warning: Missing column names filled in: 'X1' [1]

## Parsed with column specification:  
## cols(  
## X1 = col\_integer(),  
## TV = col\_double(),  
## radio = col\_double(),  
## newspaper = col\_double(),  
## sales = col\_double()  
## )

Advertising[,1] <- NULL   
lrimod = lm(sales ~ . + TV:newspaper, data = Advertising)  
coef(lrimod)

## (Intercept) TV radio newspaper   
## 3.8730824491123 0.0392939602048 0.1901312252450 -0.0320449674871   
## TV:newspaper   
## 0.0002016961778

The response ~ . ^ k syntax can be used to model all k-way interactions. (As well as the appropriate lower order terms.) Here we fit a model with all two-way interactions, and the lower order main effects.

#Advertising<- read\_csv('http://www-bcf.usc.edu/~gareth/ISL/Advertising.csv')  
#Advertising[,1] <- NULL  
lrimod2 = lm(sales ~ . ^ 2, data = Advertising)  
coef(lrimod2)

## (Intercept) TV radio newspaper   
## 6.460158490e+00 2.032709624e-02 2.292918941e-02 1.703393701e-02   
## TV:radio TV:newspaper radio:newspaper   
## 1.139280010e-03 -7.971435409e-05 -1.095976065e-04

The \* operator can be used to specify all interactions of a certain order, as well as all lower order terms according to the usual hierarchy. Here we see a three-way interaction and all lower order terms.

mod\_4 = lm(sales ~ TV \* radio \* newspaper, data = Advertising)  
coef(mod\_4)

## (Intercept) TV radio   
## 6.555887275e+00 1.971029575e-02 1.962160266e-02   
## newspaper TV:radio TV:newspaper   
## 1.310565497e-02 1.161522514e-03 -5.545501237e-05   
## radio:newspaper TV:radio:newspaper   
## 9.062944481e-06 -7.609955489e-07

### Interpretation

#### Interaction between two continuous variables

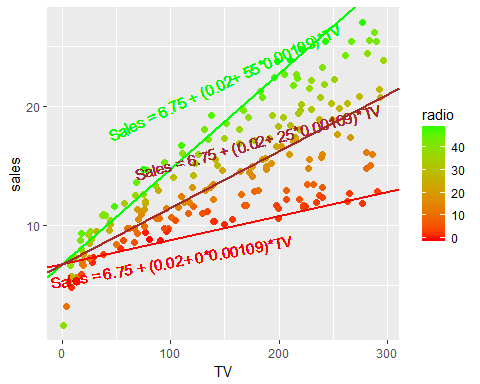
summary(lm(sales ~ TV \* radio, data = Advertising))

##   
## Call:  
## lm(formula = sales ~ TV \* radio, data = Advertising)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -6.3365780 -0.4027581 0.1830750 0.5947664 1.5246204   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 6.7502202031 0.2478713699 27.23275 < 2.22e-16 \*\*\*  
## TV 0.0191010738 0.0015041455 12.69895 < 2.22e-16 \*\*\*  
## radio 0.0288603399 0.0089052729 3.24081 0.0014005 \*\*   
## TV:radio 0.0010864947 0.0000524204 20.72656 < 2.22e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.9435154 on 196 degrees of freedom  
## Multiple R-squared: 0.9677905, Adjusted R-squared: 0.9672975   
## F-statistic: 1963.057 on 3 and 196 DF, p-value: < 2.2204e-16

### What does the coefficients means:

* (Intercept): With 0 TV ads and with 0 Radio ads the sales will be 6.750
* TV: increasing TV ads by 1000 at zero Radio ads increase sales by 19.1 widgets
* Radio: increasing Radio ads by 1000 at zero TV ads increase sales by widgets
* <TV:Radio>: the effect of TV ads on sales increases by 1.09 for every unit increase in Radio ads. For example at Radio ads of 0 the slope between TV ads and sales is almost equal , at Radio ads of 10 the slope between TV ads and sales will be almost equal

# plot  
ggplot(Advertising, aes(x = TV, y = sales, color = radio))+ geom\_point(size = 2) + scale\_color\_gradient(low = "red",   
 high = "green") + geom\_abline(intercept = 6.75, slope = 0.02, color = "red", size = 1) +  
 geom\_abline(intercept = 6.75, slope = (55 \* 0.00109)+0.02, color = "green",   
 size = 1) + geom\_abline(intercept = 6.75, slope = (25 \* 0.00109) +0.02, color = "brown",   
 size = 1) +  
 geom\_text(label = "Sales = 6.75 + (0.02+ 0\*0.00109)\*TV", x = 100, y = 7, color = "red", size = 4, angle=10) +   
 geom\_text(label = "Sales = 6.75 + (0.02+ 25\*0.00109)\*TV", x = 180, y = 17, color = "brown", size = 4, angle=15) +   
 geom\_text(label = "Sales = 6.75 + (0.02+ 55\*0.00109)\*TV", x = 150, y = 22, color = "green", size = 4, angle=25)



### Interaction between one continuous and one categorical variables

Now let's turn to another case, there we are weighting standardize soil samples, we added a temperature treatment with two levels (Low, High) and we measured the soil nitrogen concentration, we would like to see the effects of the nitrogen concentration and its interaction with temperature on soil weight.

set.seed(12345)  
# create a Series of booleans in which roughly half are True  
nums = runif(nrow(Advertising), 0, 1)   
mask\_large = nums > 0.5   
# initially set Size to small, then change roughly half to be large  
Advertising[['Size']] = rep('small',nrow(Advertising))  
Advertising$Size[mask\_large] = 'large'  
Advertising$Size = as.factor(Advertising$Size)  
#Advertising = Advertising %>% mutate(isLarge = factor(if\_else(Size == "large",1,0))) %>% dplyr::select(-Size)   
Advertising

## # A tibble: 200 x 5  
## TV radio newspaper sales Size  
## <dbl> <dbl> <dbl> <dbl> <fctr>  
## 1 230.1 37.8 69.2 22.1 large  
## 2 44.5 39.3 45.1 10.4 large  
## 3 17.2 45.9 69.3 9.3 large  
## 4 151.5 41.3 58.5 18.5 large  
## 5 180.8 10.8 58.4 12.9 small  
## 6 8.7 48.9 75.0 7.2 small  
## 7 57.5 32.8 23.5 11.8 small  
## 8 120.2 19.6 11.6 13.2 large  
## 9 8.6 2.1 1.0 4.8 large  
## 10 199.8 2.6 21.2 10.6 large  
## # ... with 190 more rows

levels(Advertising$Size)

## [1] "large" "small"

mlrmodcc <- lm(sales ~ TV\*Size, data = Advertising)  
summary(mlrmodcc)

##   
## Call:  
## lm(formula = sales ~ TV \* Size, data = Advertising)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -8.5850756 -1.7503829 -0.1749077 2.0141088 7.2101746   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 7.434414183 0.575741759 12.91276 < 2e-16 \*\*\*  
## TV 0.044620481 0.003555404 12.55004 < 2e-16 \*\*\*  
## Sizesmall -1.064360979 0.952970654 -1.11689 0.26541   
## TV:Sizesmall 0.007030783 0.005515518 1.27473 0.20391   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 3.261745 on 196 degrees of freedom  
## Multiple R-squared: 0.6150667, Adjusted R-squared: 0.6091749   
## F-statistic: 104.3931 on 3 and 196 DF, p-value: < 2.2204e-16

Again let's go through the coefficients:

* (Intercept): Under large market size conditions and with a TV ads of , sales are equla to 7.434414.
* TV: increase of $1000 in TV ad spending is associated with an increase in Sales of 44.620 widgets.
* SizeSmall: Under small market conditions and TV ad spending equals Sales are 1.064 lower compared to large market size condition.
* <TV:Sizesmall> : the effect of TV ads on sales is higher by 0.007 under small market conditions compared to large market conditions. In other words the slope between TV ads and sales is 0.044+0.007 = 0.051 in small size markets.

### What Didn't We Cover?

* Detecting collinearity
* Diagnosing model fit
* Transforming predictors to fit non-linear relationships
* Assumptions of linear regression
* Automatic model selection
* And so much more!

You could certainly go very deep into linear regression, and learn how to apply it really, really well. It's an excellent way to start your modeling process when working a regression problem. However, it is limited by the fact that it can only make good predictions if there is a linear relationship between the features and the response, which is why more complex methods (with higher variance and lower bias) will often outperform linear regression. Therefore, we want you to understand linear regression conceptually, understand its strengths and weaknesses, be familiar with the terminology, and know how to apply it. However, we also want to spend time on many other machine learning models, which is why we aren't going deeper here.

### Resources

To go much more in-depth on linear regression, read Chapter 3 of An Introduction to Statistical Learning (resources), from which this lesson was adapted. Alternatively, watch the related videos.