Module 1: Regression Algorithm for Testing and Predicting Business Data

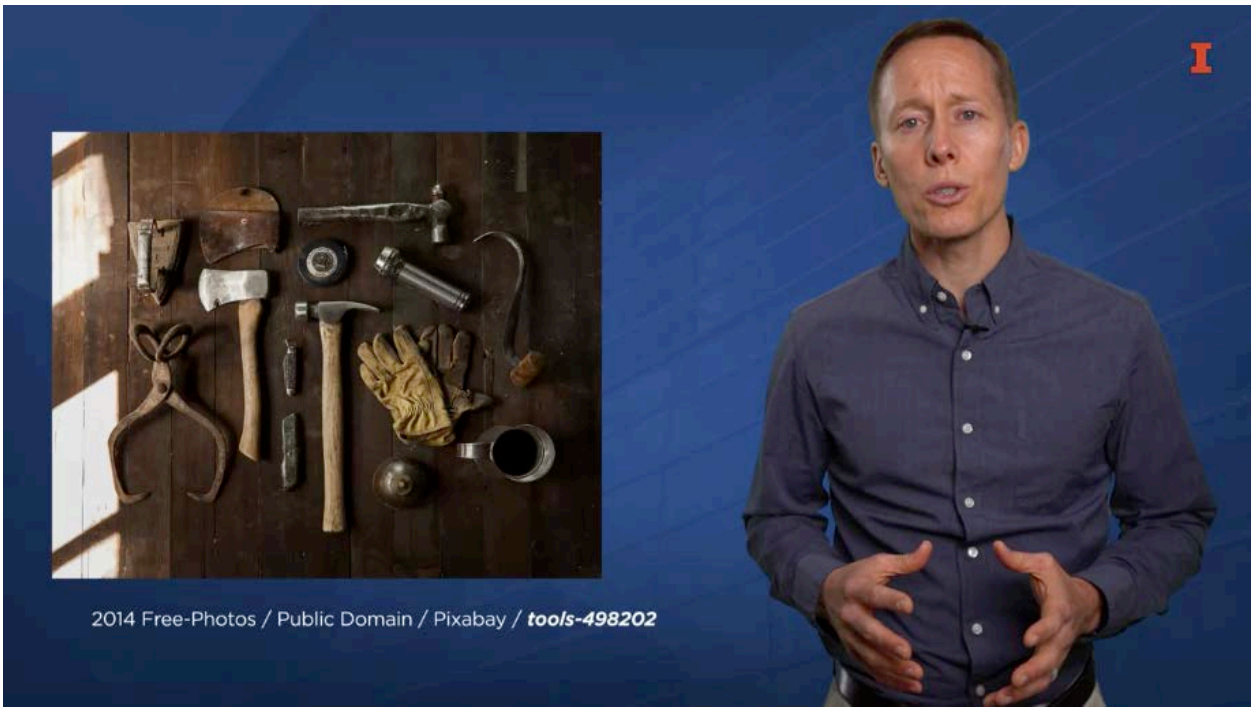
Table of Contents

Lesson 1-0: Course Introduction

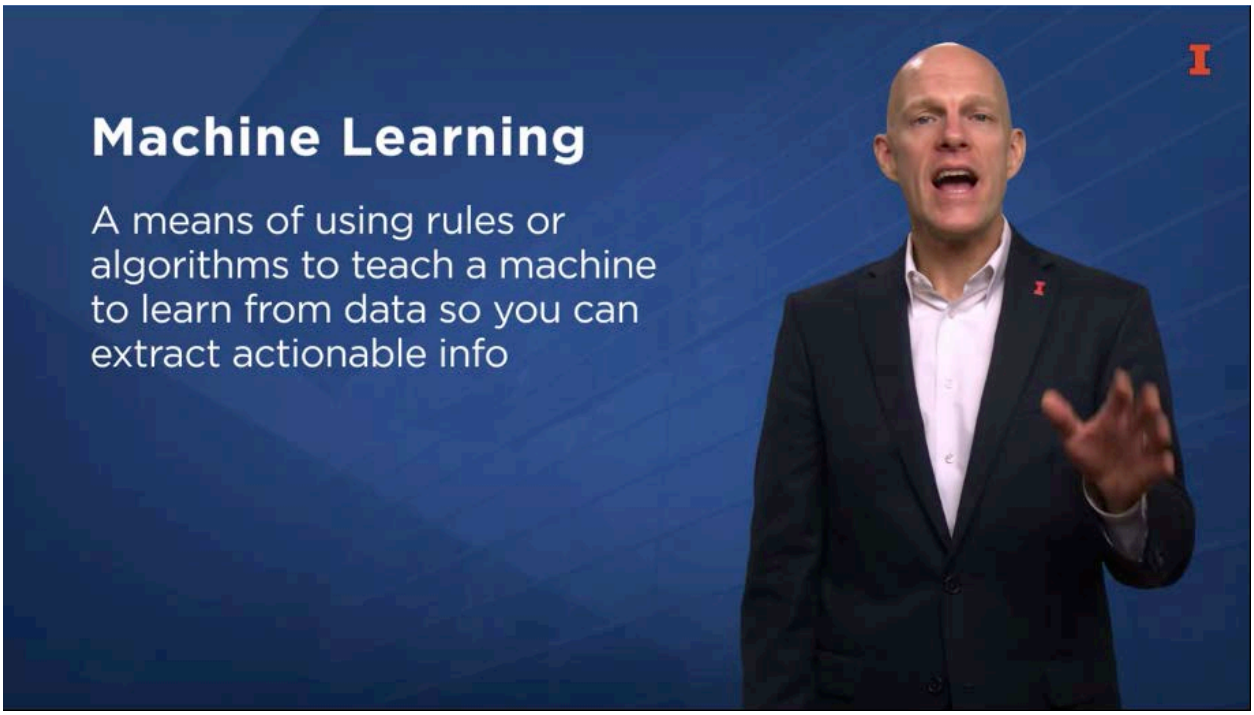
Lesson 1-0.1 Course Introduction

This course is about giving you the tools you need to use data to gain actionable business insight. Nearly everywhere we go and nearly everything we do, from shopping online to typing a text is infused and enhanced with big data, machine learning, and data analytics.

To win in business, and to even be a successful participant, you and I need to learn how to master tools that help us take data and turn it into usable business insights.



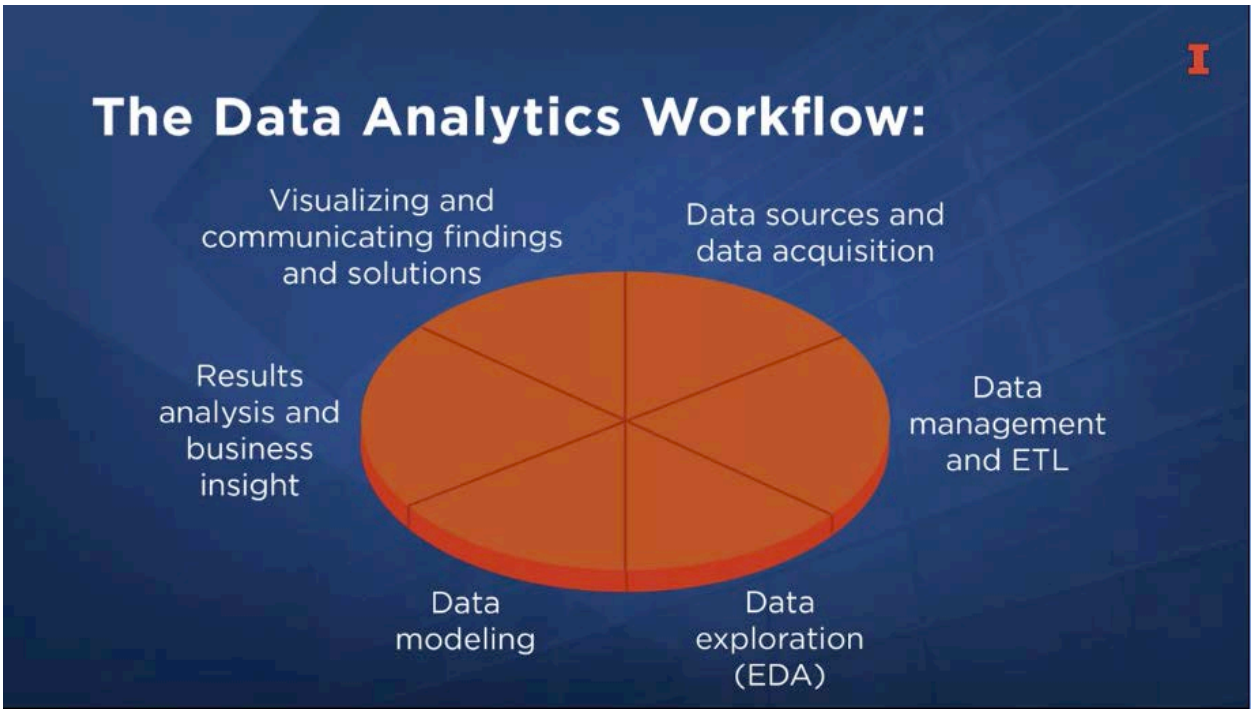
This course will give you necessary and cutting-edge tools to put you in the game. These tools will allow you to leverage statistical techniques and machine learning to understand the relationships and interrelations of the features of your data and to create models to use those features to predict future outcomes.



These tools are basic, classic algorithms in each of the major areas of machine learning. Machine learning is a means of using rules or algorithms to teach a machine to learn your data, allowing you to extract actionable information from the data.

In this course, we cover the two largest and most used methods of machine learning: supervised learning and unsupervised learning. Supervised learning examines data that has labels or outcomes, such as the amount of sales for a quarter, fraud or no fraud,

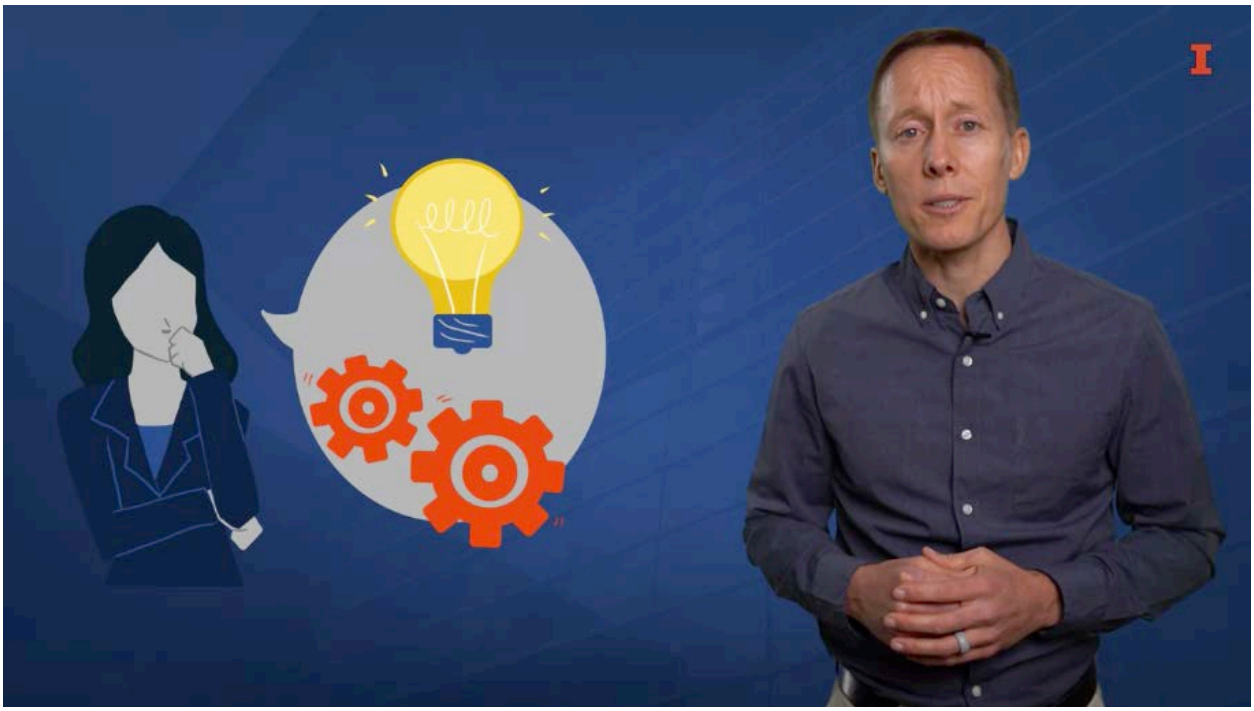
and loyalty customer or non-loyalty customer. The algorithms we will cover allow you to use your data to examine these outcomes and predict them in the future. We cover the following four supervised learning algorithms: Regression, logistic regression, K-nearest neighbors, and decision trees. Regression allows you to investigate and predict future numerical outcomes such as sales, costs, and gross margin. Logistic regression, K-nearest neighbor, and decision trees are classification algorithms and they allow you to investigate and predict discrete classification outcomes such as hire or fire, success or failure, fraud or no fraud. Unsupervised learning works differently in that it has no labeled outcomes, thus, these algorithms seek to generate labels by learning from the data. We covered two key unsupervised learning algorithms, k-means clustering, and DBSCAN clustering.



In each module, we'll use each of these tools to focus on the second half of the analytics workflow. The data analytics workflow consists of the following parts: First, acquiring and maintaining data, second, getting data ready for analysis, we often call this ETL for extracting, transforming, and loading data, third, data exploration, including the steps we take to understand and view our data, we often call this exploratory data analysis or EDA, fourth, data modeling, including predicting future outcomes and inferring relationships from our data to other data and situations, fifth, creating analysis results and business insight, and sixth, visualizing and communicating findings and solutions.

In particular, in this course, we'll focus on steps 4 and 5, and we'll use the algorithms to learn and master those steps. For example, in steps 4 and 5, you've acquired data,

cleaned it and prepare it for analysis, and visualized and explored it, now, you want to extract information and business insight from the data and predict future outcomes to help you answer business problems.



In each module, we will use realistic business data to practice using these tools. Thus, in each module, we will focus on solving business problems. Our goal in this course is to provide you with a solid framework and foundation for how to understand and practice business analytics. Thus, in the future, when you encounter a data analysis task that you have not encountered before, you will be able to slot it into the framework, understand it quickly, and move more rapidly through the process of assimilating the new information that you need to learn. More importantly, you will be able to put your new knowledge to work for you in solving the business problem.

We're excited for you to expand your toolkit by learning more about these tools, data prediction and modeling are the foundation for all data analysis.

Opening a new data set and understanding it, it's like unlocking a treasure chest or solving a puzzle, it's like peeling an onion and learning layer by layer, it's like opening a matryoshka doll or a nested doll and finding the truth at the center. As you go through these modules, dig in and play with the data. The more you practice, the more you expand your data toolkit and your ability to solve business problems.

References:

Free-Photos. (2014). **tools-498202** [Online Image]. CC0 1.0. Pixabay.

Free-Photos. (2015). **tie-690084** [Online Image]. CC0 1.0. Pixabay.

Piacquadio, A. (2018). **Wearing Black Zip Jacket Holding Smartphone Surrounded by Grey Concrete Buildings** [Online Image]. CC0 1.0. Pexels.

Lesson 1-0.2 About Professor Jessen Hobson

Hi, my name is Jessen Hobson. I'm excited for you to take this class. I'd like to give you just a short biography, a little bit about my life, where I've come from, and what I've done. I grew up in Boise, Idaho, which is out in the West, now the Intermountain West in the Mountains. Went from there, and got my undergraduate education at Brigham Young University. I started there, and then took two-year hiatus to serve a mission for my church in Antofagasta, Chile. I came back to BYU and graduated with a bachelor's and a master's in accounting. Also, met my wonderful wife along the way. I went from there to Washington, D.C., and worked as an auditor for PricewaterhouseCoopers. But even then I knew I wanted to go and get a Ph.D., so I could teach. Shortly thereafter, I went to the University of Texas at Austin and got a Ph.D. in accounting. After graduating, I went and worked at Florida State University, and then came here to the University of Illinois. I've taught accounting audit and most recently, business analytics and data analytics. I'm really involved in data analytics and really enjoy it. Most recently, and really throughout my whole career, I've focused on data, and how I can teach and use data to solve business problems. I'm excited to be here, I'm excited that you're going to take this class. Be great.

Lesson 1-0.3 About Professor Ronald Guymon

Hi, my name is Ron Guymon and I'm on the faculty here in the School of Accountancy at the Gies College of Business. I have accounting degrees from Brigham Young University and the University of Iowa. My professional experience has been a mixture of academics and practice. In academics, my teaching and research has focused on management accounting, and data analytics. In practice, I've worked as a data scientist. There's a little bit about my professional experience. Let me tell you a little bit now about my personal life.

All right. Well, here I am. I'm at the top of a goblin here in Goblin Valley. I love it up here. The hike is really fun. It's also a majestic view. You see panoramic landscapes. I love being able to be out here. As a young man, I was able to come and hike and camp in this area. Now, at this stage in life, I spend a lot of time doing the job I love but not being outside as much. So I get to bring my children down here sometimes and watch them run and hike around. I love it. My wife and I a few years ago had a chance to run a race down here. We ran clear out and around some of the area here. My favorite part of the race was ending here in what's called the Valley of the Hutus. At the very end, we ran up some stairs and finished where we could see all the goblins. It's one of my favorite races I've ever run.

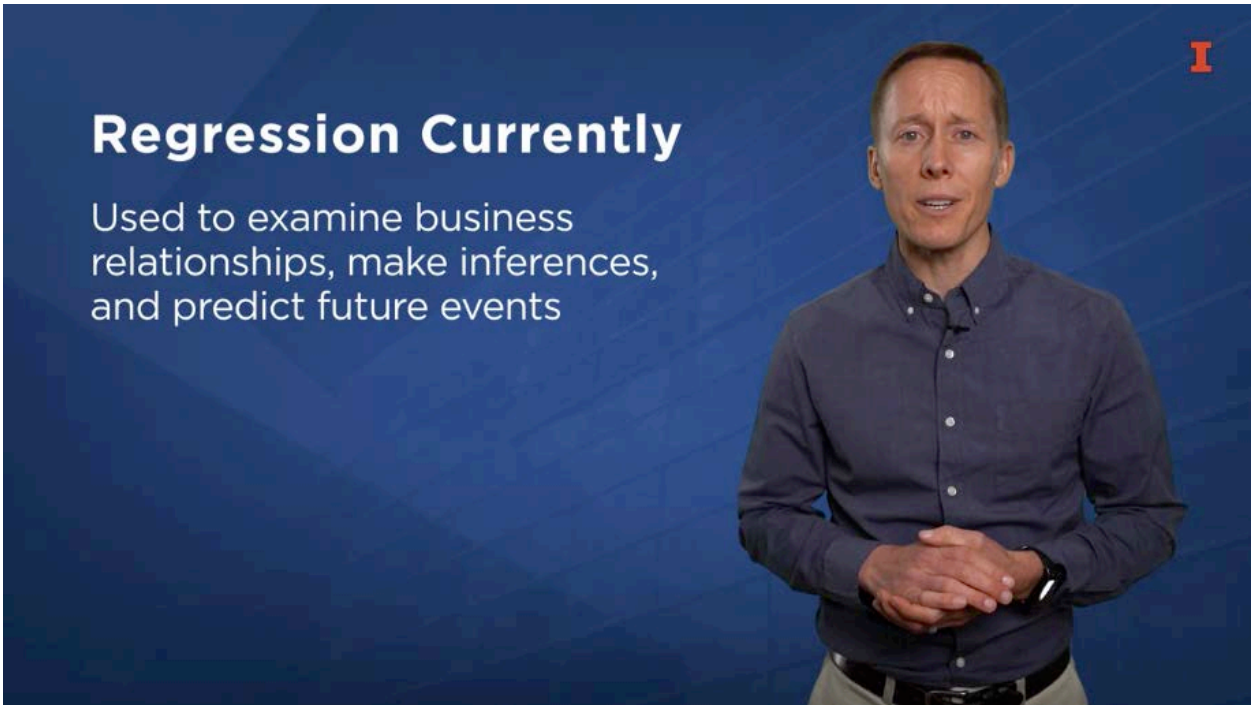
I'm here at Arches National Park, admiring these tremendously large arches. Behind me, we have a couple of arches that look like a bridge. When I think of bridges, I think of teachers. Why do I think of teachers when I think of bridges? Well, one of the greatest teachers I've ever known of, Thomas Monson, used to talk about the importance of building bridges for others to cross. He once shared a poem called The Bridge Builder, in which an old man is traveling and he gets to a giant chasm. At the bottom of this chasm is a river. Now this old man has a lot of experience and was able to find his way across the river and to the other side of the chasm. Once he gets across, he stops to build a bridge. Another traveler passing by asks him, "Why are you building a bridge if you've already crossed?" Field man replies that he's not building the bridge for himself, he's building the bridge for others so that when they have to cross, they'll have an easier go at it. I'm grateful for the opportunity to be a teacher and I aspire to be a good bridge builder. I hope you too as a result of your education, take time to build bridges for others. So I hope that gives you a better idea of who I am. I'm grateful to be working at the Gies College of Business and I hope you enjoy the course.

Lesson 1-1: Overview

Lesson 1-1.1 Introduction

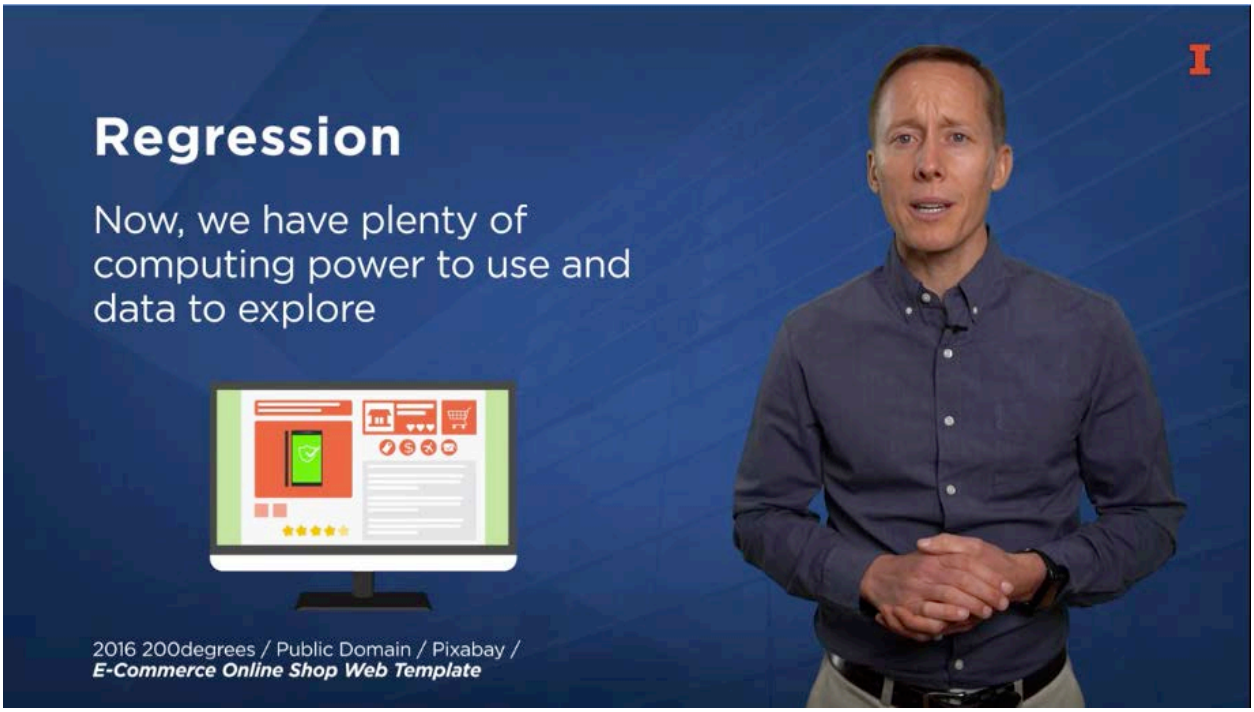
In this set of lessons, we want to introduce you to the regression algorithm. Regression has been used for over 200 years, so it's not a new algorithm. But just because it's an old algorithm doesn't mean that it's irrelevant. Neural networks have been around for about 80 years, so they're not particularly new either.

Regression was used in the 19th century to predict rough patterns of planetary movement. It has been used in academic research for many years to examine complex relationships including in the social sciences.



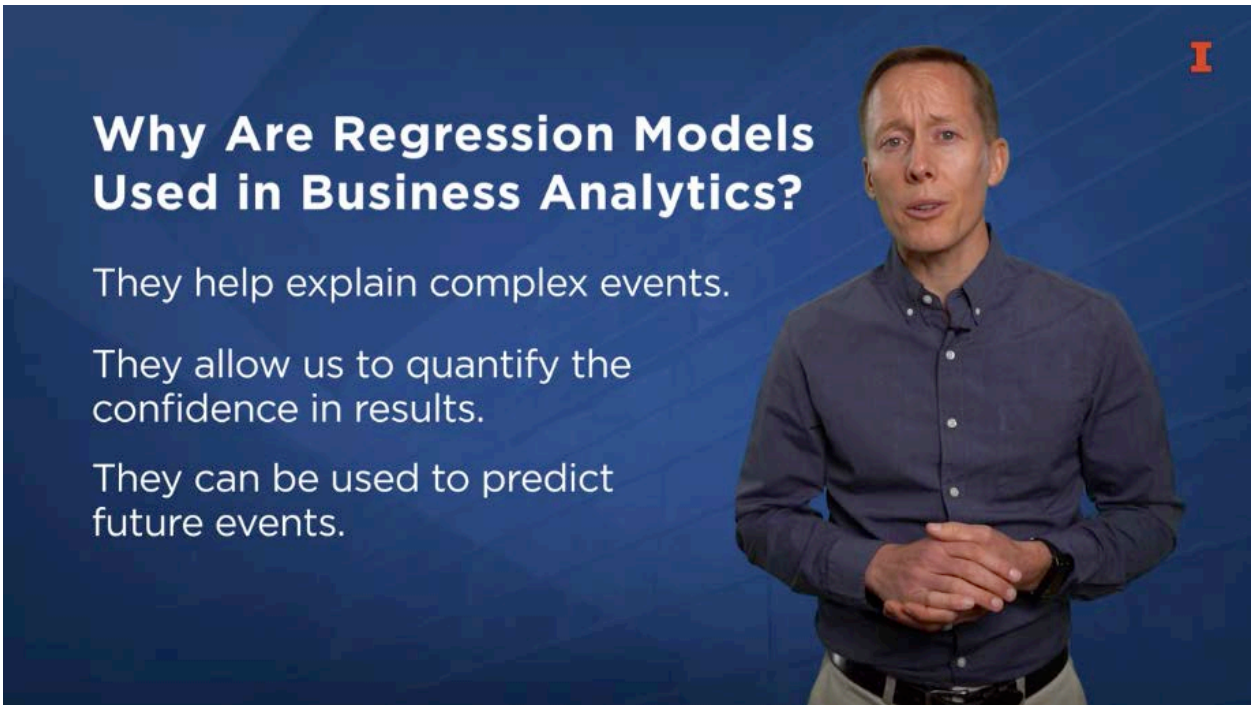
Regression is currently used to estimate the fundamental value of businesses, and to make inferences about various business practices on stock price. In my experience,

regression is used now more than ever as many businesses tried to examine relationships, make inferences, and predict future events.

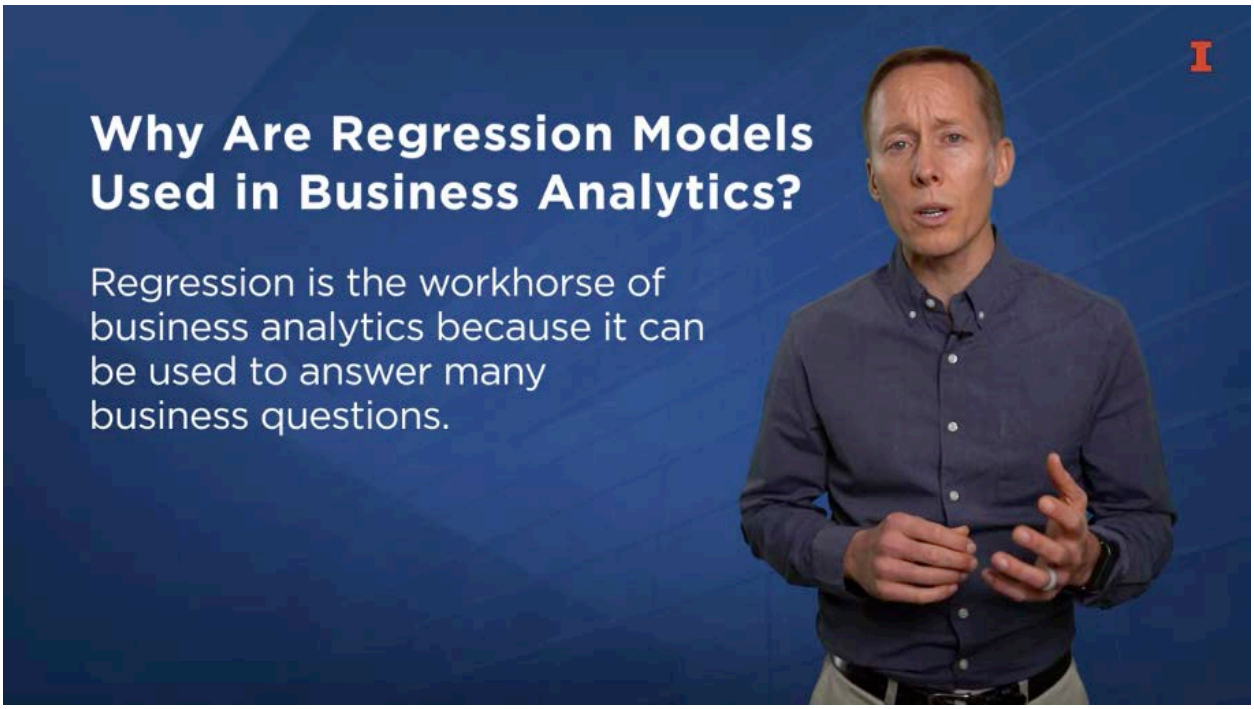


So why has it taken so long for business to implement regression? One major reason is because in our day, we have plenty of computing power and lots of data that needs to be investigated.

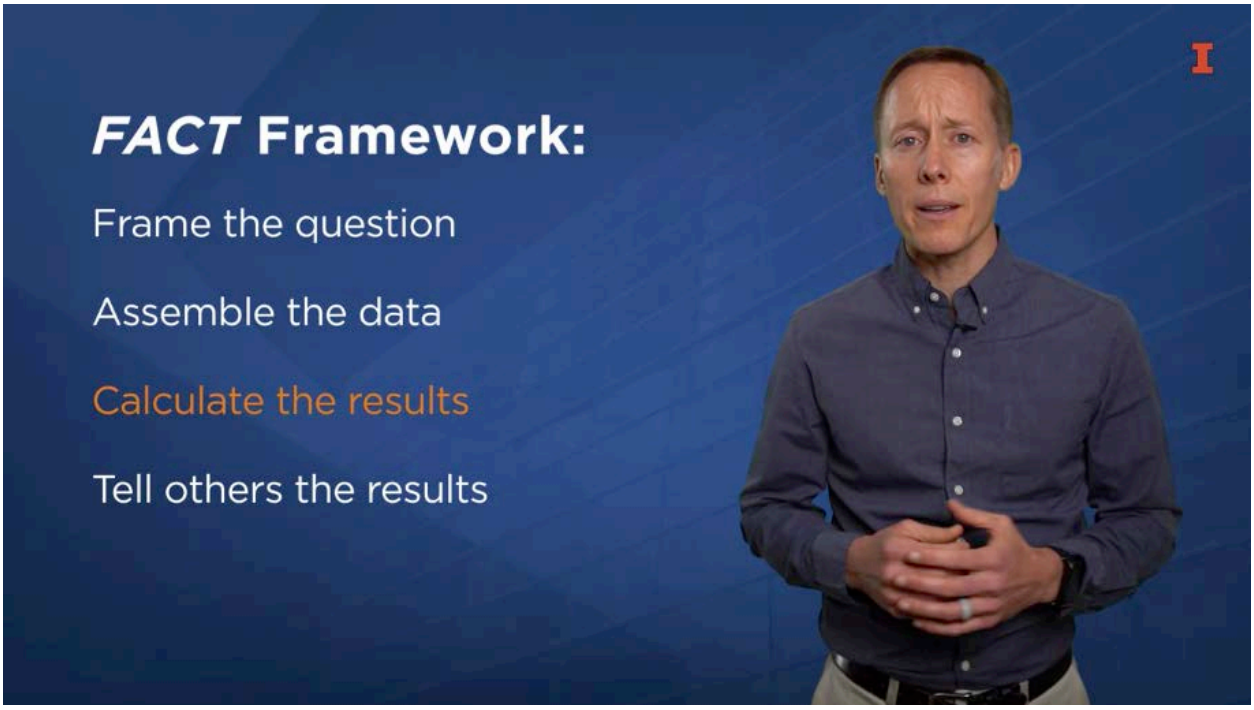
In the olden days, regression was painstakingly performed by making calculations by hand or using punch cards that were fed into computational machines. Gathering data about environmental events and human behavior was not easy either. You can imagine that such a labor-intensive process would make it infeasible for many organizations to use regression.



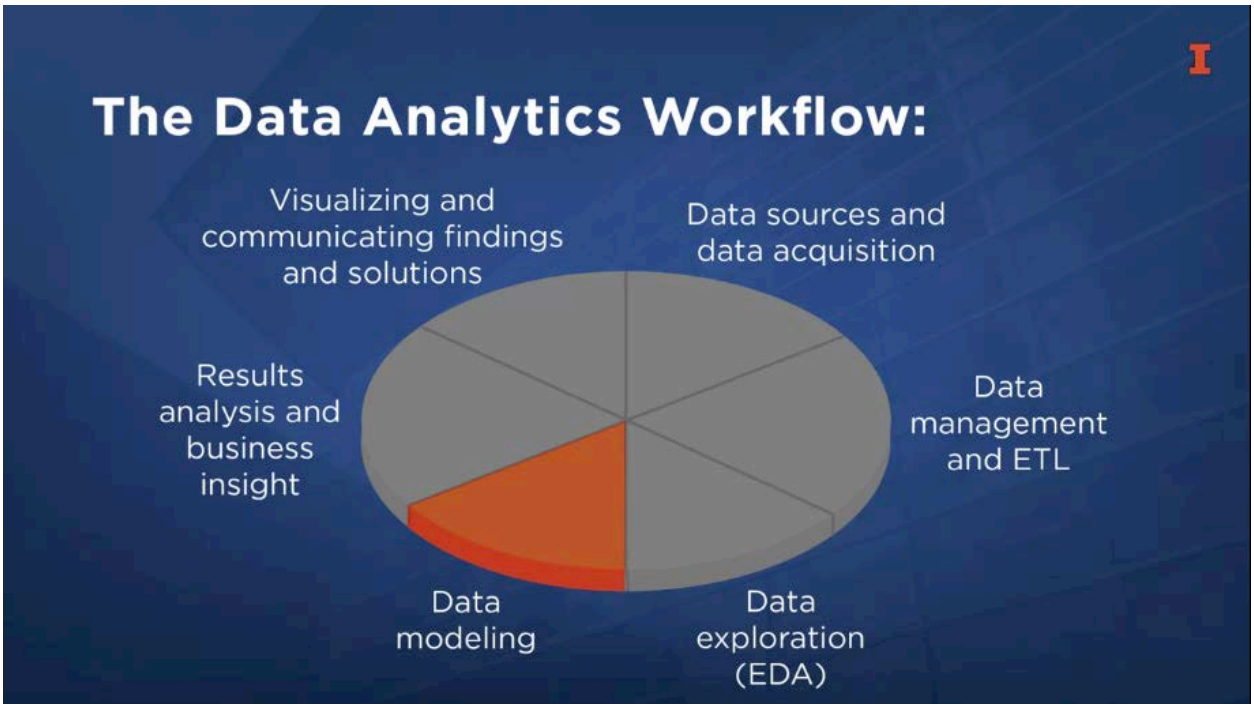
You might also wonder why regression continues to be used in business analytics. One reason is that regression models make it possible to explain complex events in terms of individual effects. Another reason is because regression results allow us to quantify the confidence that we place in the results. Finally, regression models can be used to predict future events.



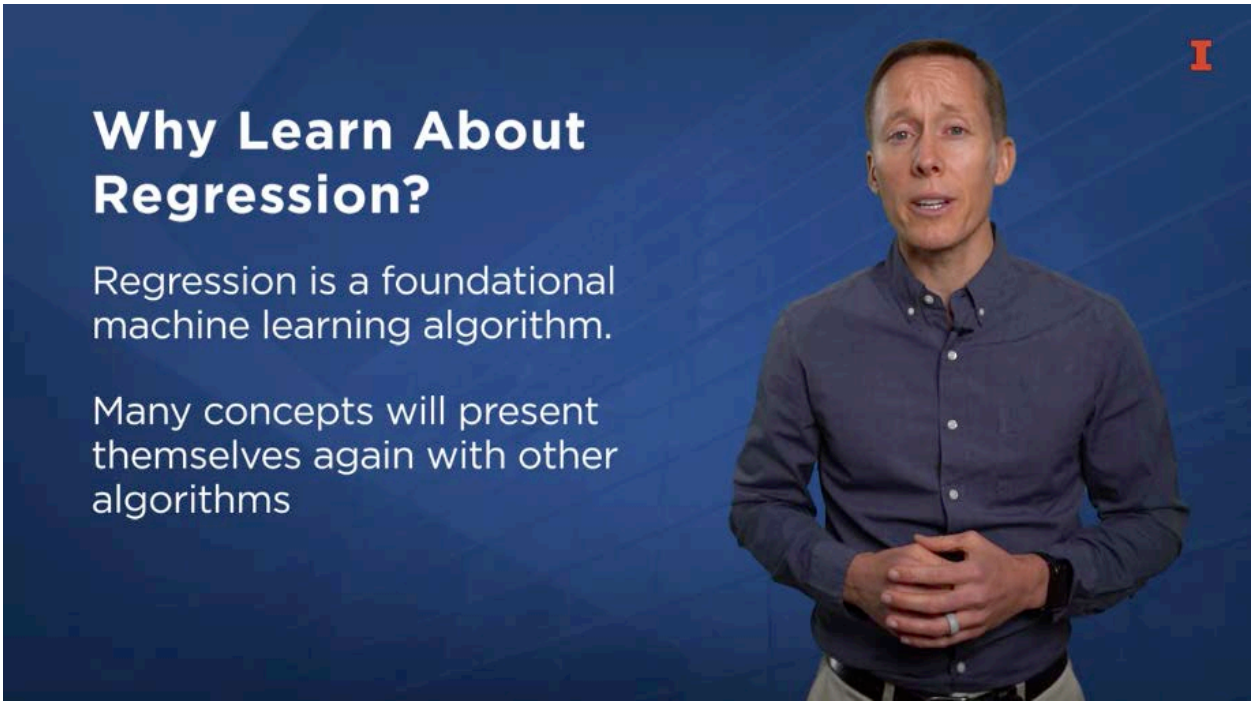
Because regression can be used to help answer so many business questions, I would say that regression is the workhorse of business analytics.



Where does regression fit into the data analytic pipeline? Regression is an important part of making calculations with data. Once the data has been assembled and exploratory data analysis had been conducted, regression can be used to create models for evaluating complex relationships.

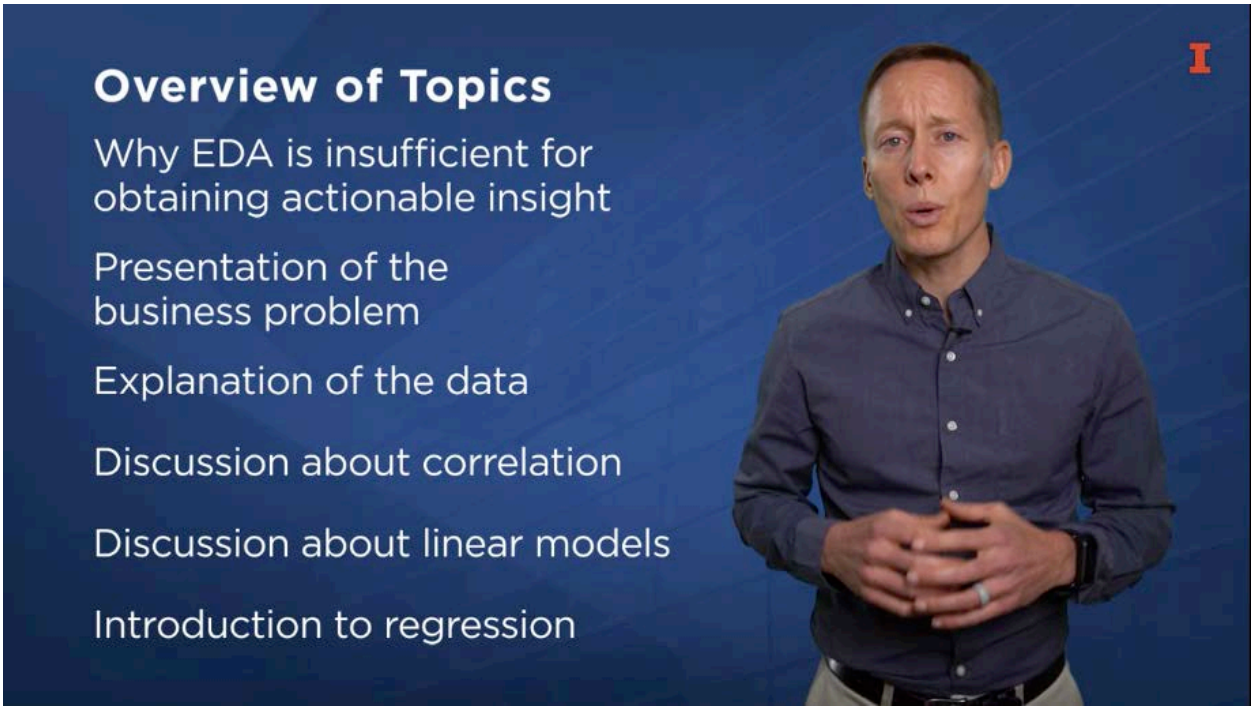


In these lessons, we will focus on how regression can be applied to one broad business question, but we will touch on a variety of other business questions. Hopefully, as you go through these questions, you will get ideas for business questions of your own that you would like to answer.

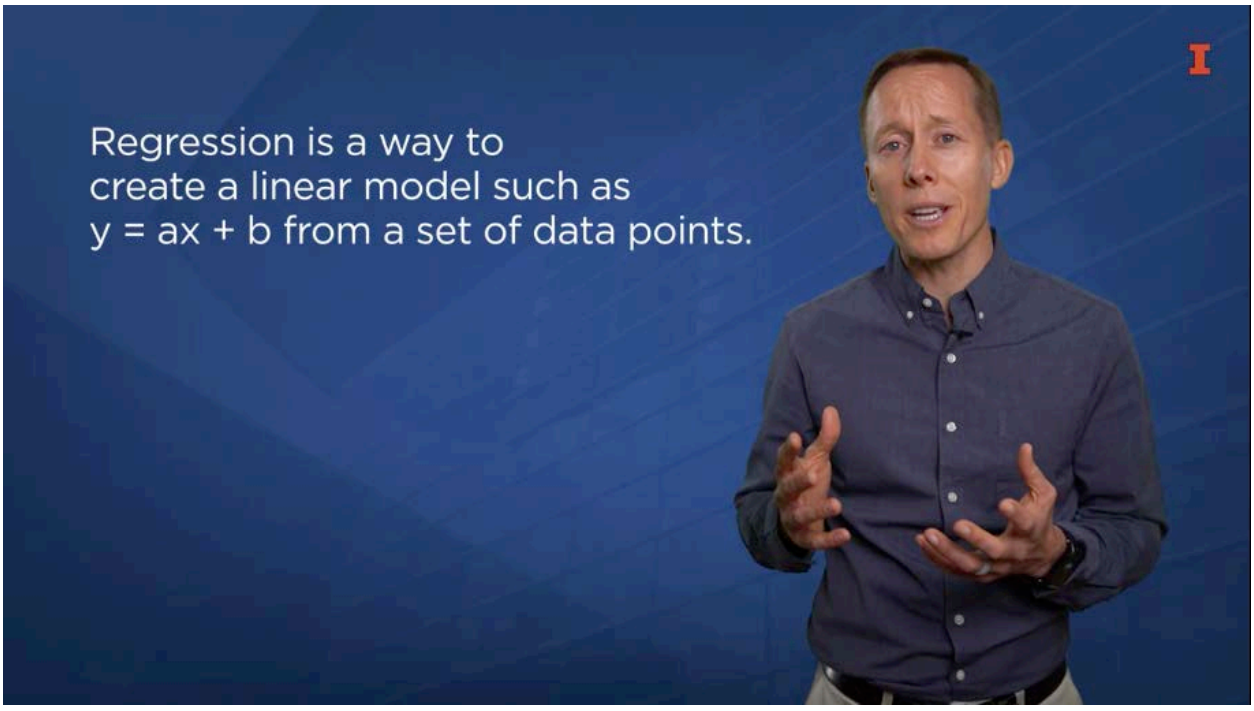


Another reason why it's worthwhile to learn about regression is because it is a foundational machine learning algorithm. As you learn about regression, you will

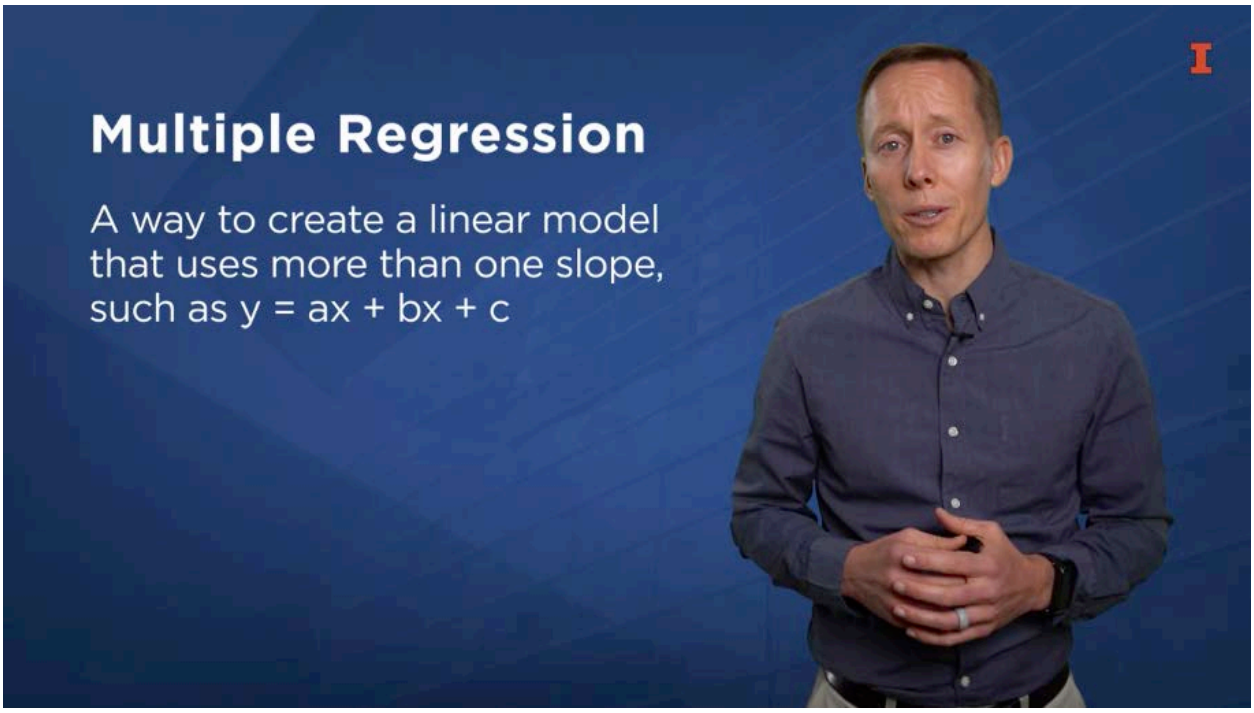
encounter some technical terms like R-squared, coefficient estimates and residuals. Don't get intimidated though. These terms are used to describe simple concepts. Also, keep in mind that these concepts will either come up as you learn about other algorithms or similar concepts will be used with the other algorithms.



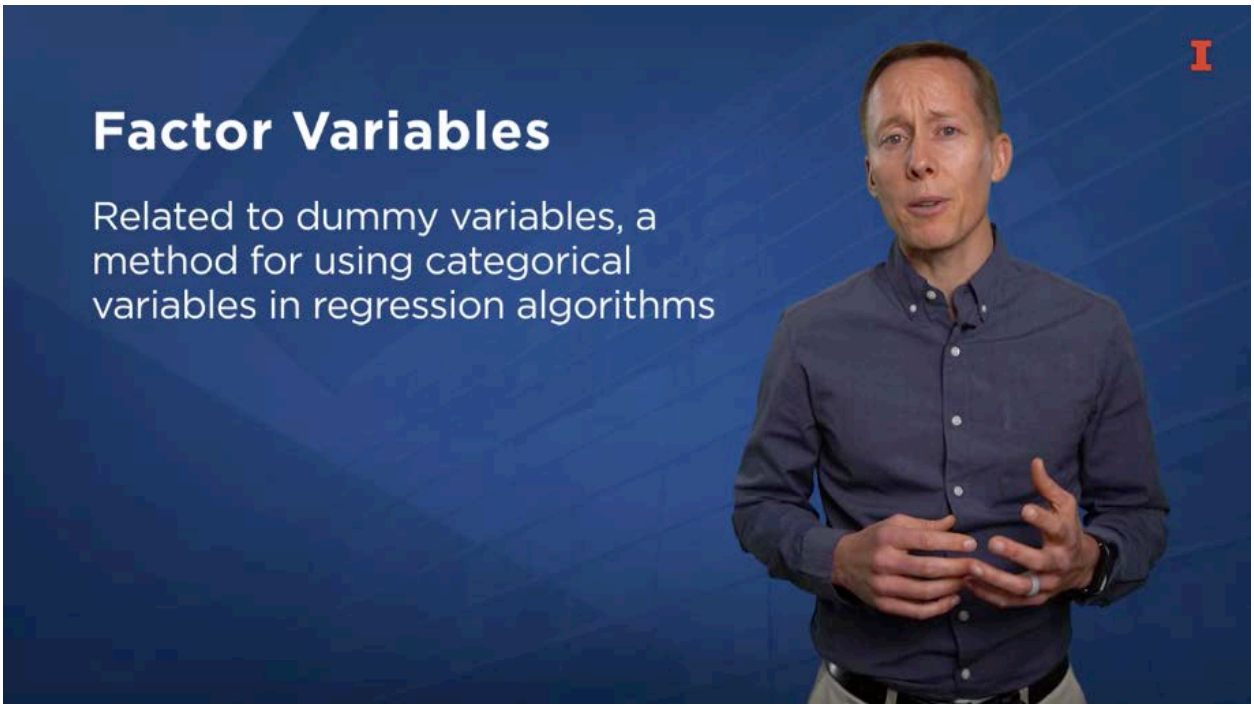
Before getting into regression, we will examine why exploratory data analysis is insufficient to obtain and implement actionable insight. We will then present the business problem and explain the data that we'll be using to find answers with regression. We will also review concepts with which you may already be familiar, such as correlation and linear models.



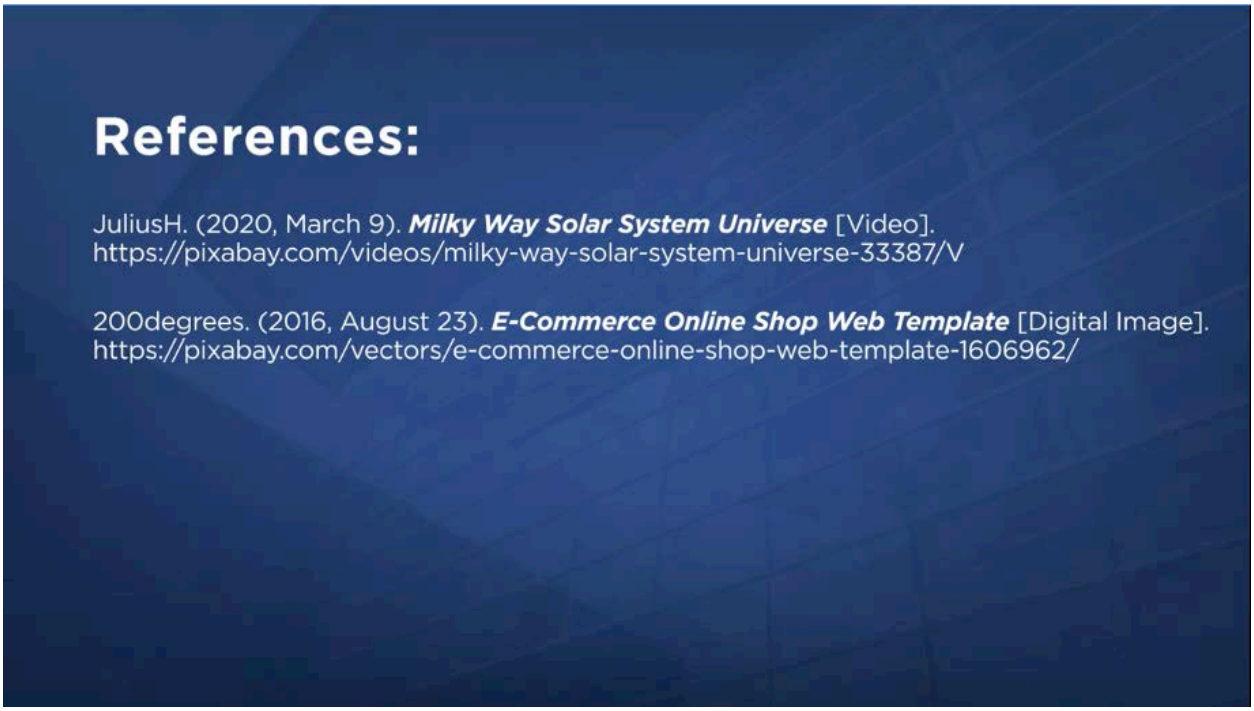
Regression is really just an extension of these concepts. Specifically, regression is a way to create a linear model such as y equals ax plus b from a set of data points. After introducing regression, we'll discuss how to run a simple regression algorithm in R, and then learn how to interpret the results and evaluate the extent to which the resulting linear model provides actionable insight.



We will also introduce you to multiple regression, which is a way to create a linear model that uses more than one slope, such as y equals ax plus bx plus c. Multiple regression is powerful because it allows you to consider the simultaneous effect of many influences at once. Running it in R is a straightforward extension of running a simple regression.



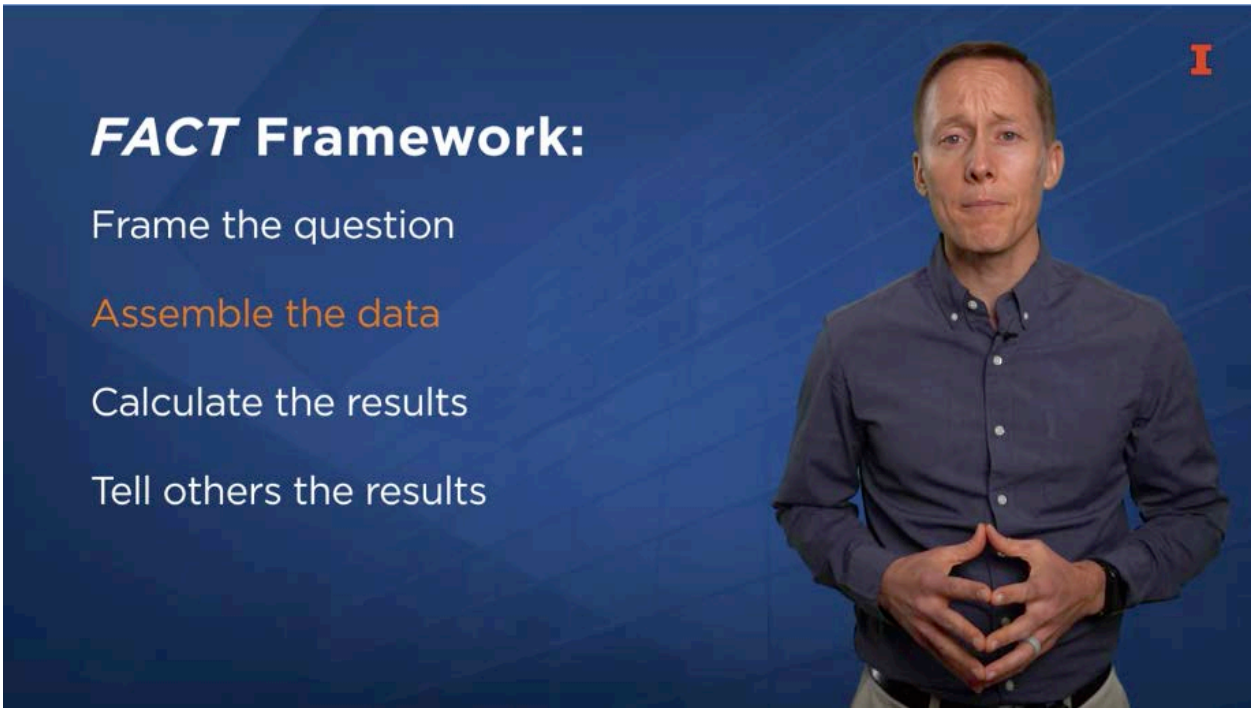
Finally, we will introduce you to why factor variables are so important in R, they're related to the concept of dummy variables, which is a method for using categorical variables in regression algorithms. If I sound like an infomercial in which the host keeps saying, "But wait, there's more," I hope you'll forgive me. However, regression really is a bargain, because you get so much insight from this algorithm.



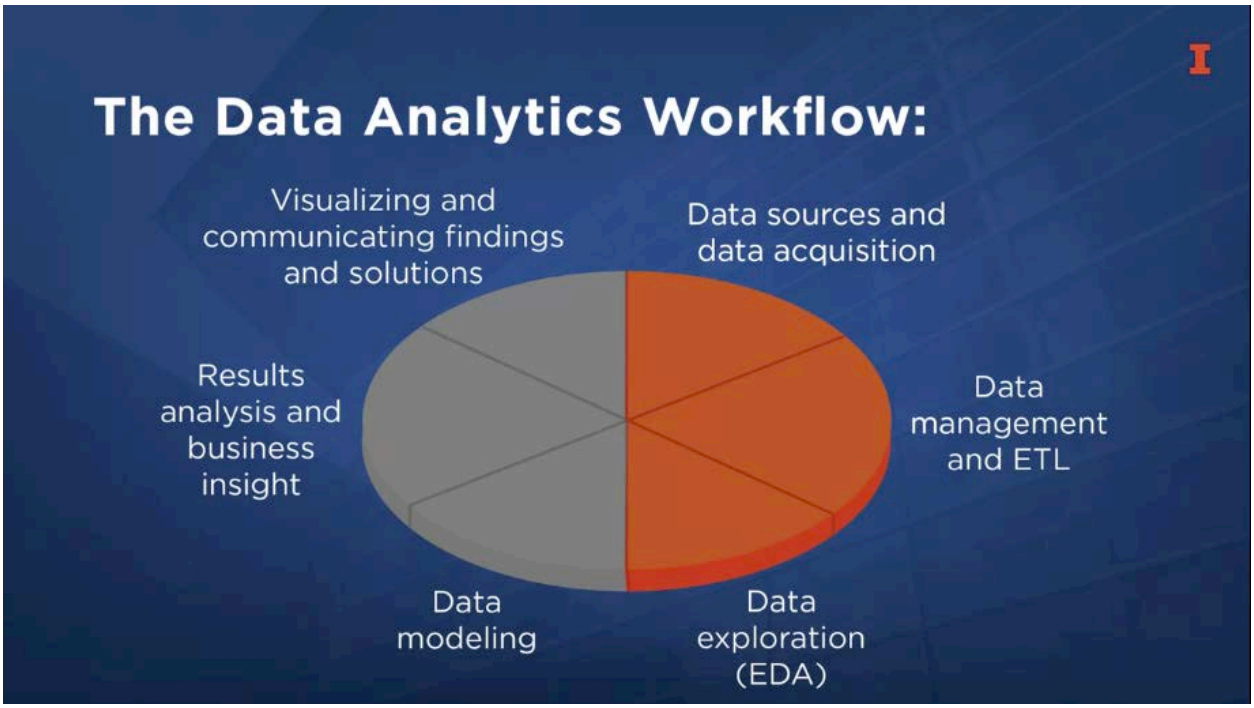
Lesson 1-2: Why Isn't EDA Enough?

Lesson 1-2.1 Why Isn't EDA Enough?

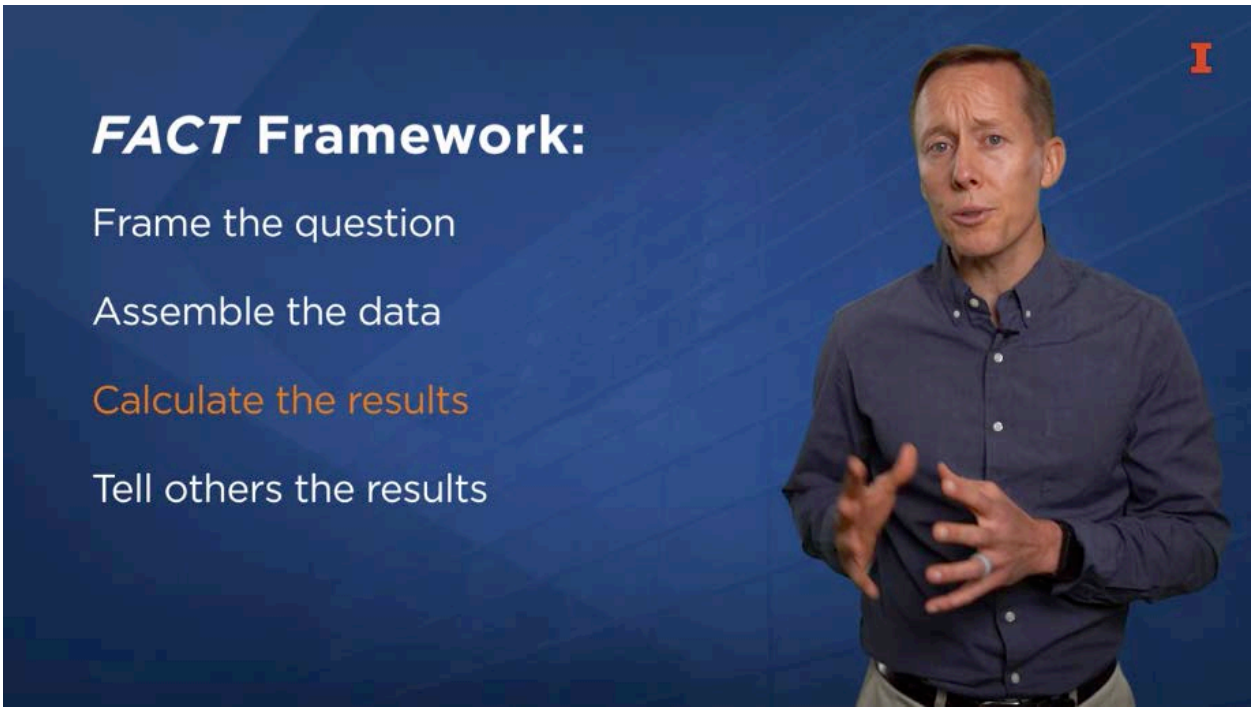
In this lesson, we want to explain the need to use sophisticated data analytic algorithms by exploring why exploratory data analysis or EDA, is often not sufficient to obtain and implement actionable insights.



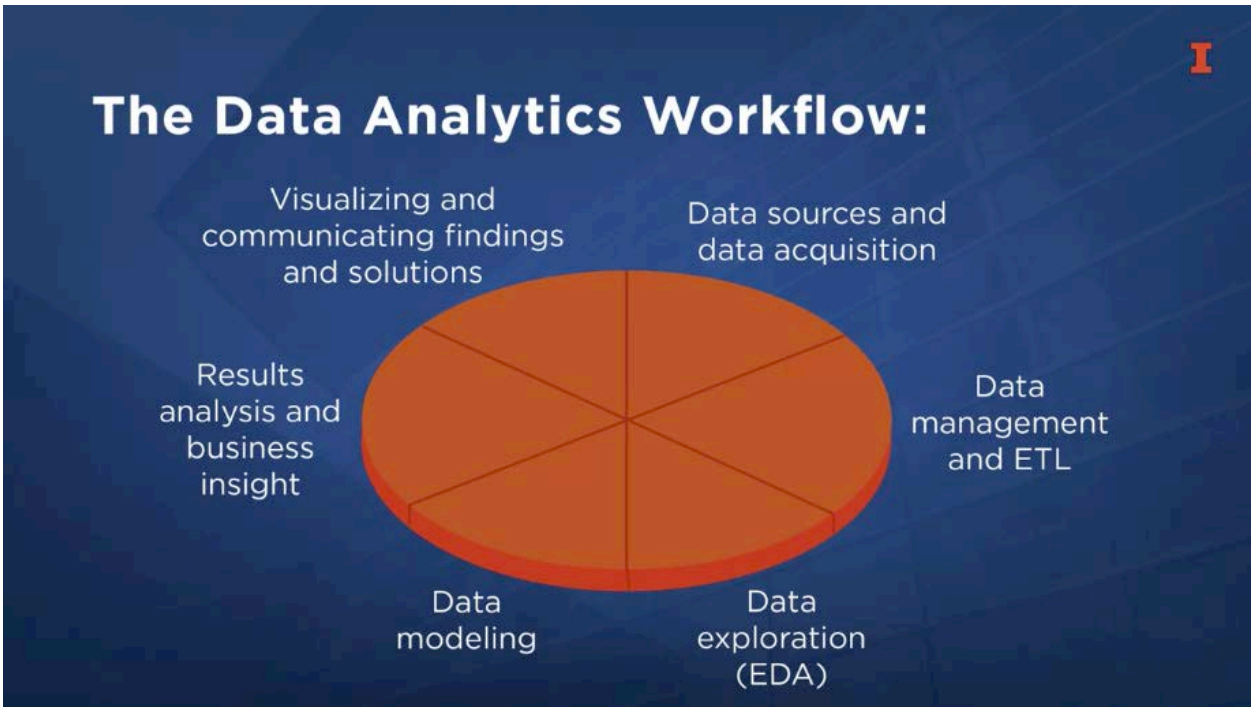
The data analytic pipeline consists of several steps that we have grouped within the fact framework.



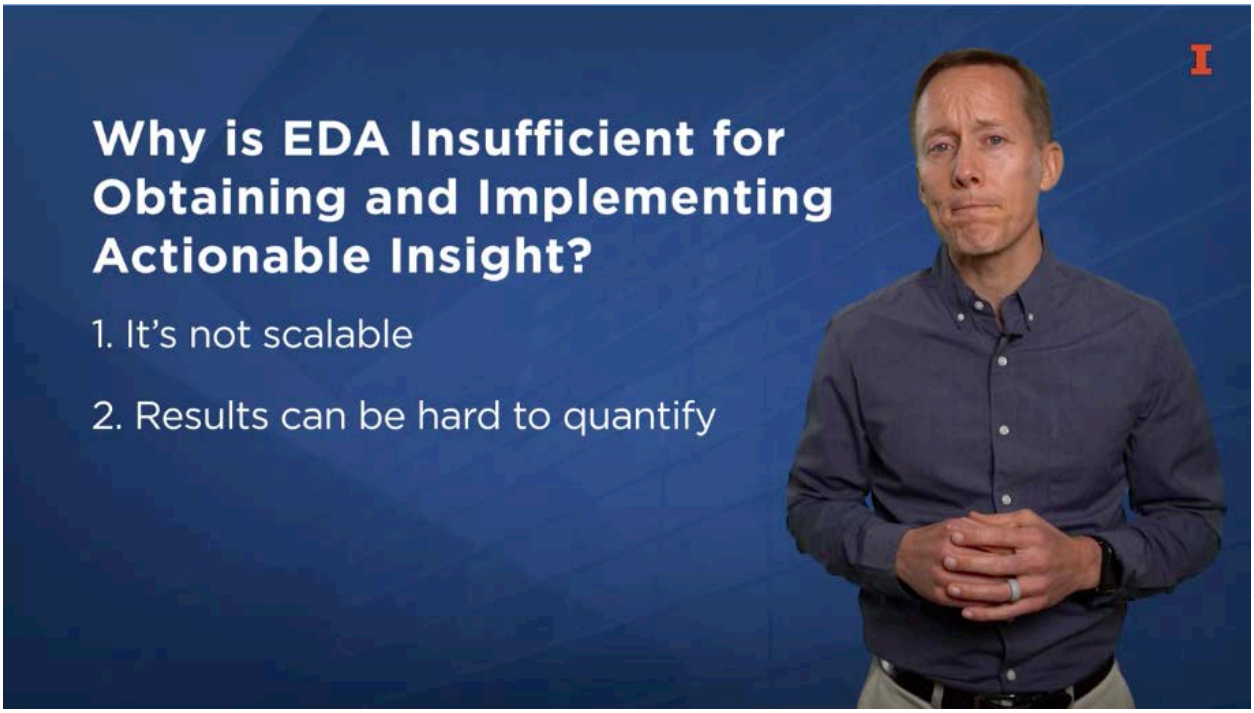
If we expand the a of the fact framework assembling data, the steps are identifying data sources and acquiring the data, extracting, transforming, and loading the data, also known as ETL, data wrangling, data preprocessing, and then exploratory data analysis. Exploratory data analysis can lead to many insights in and of itself.



In fact, EDA is often a transition point into the calculate the results portion of the fact framework.

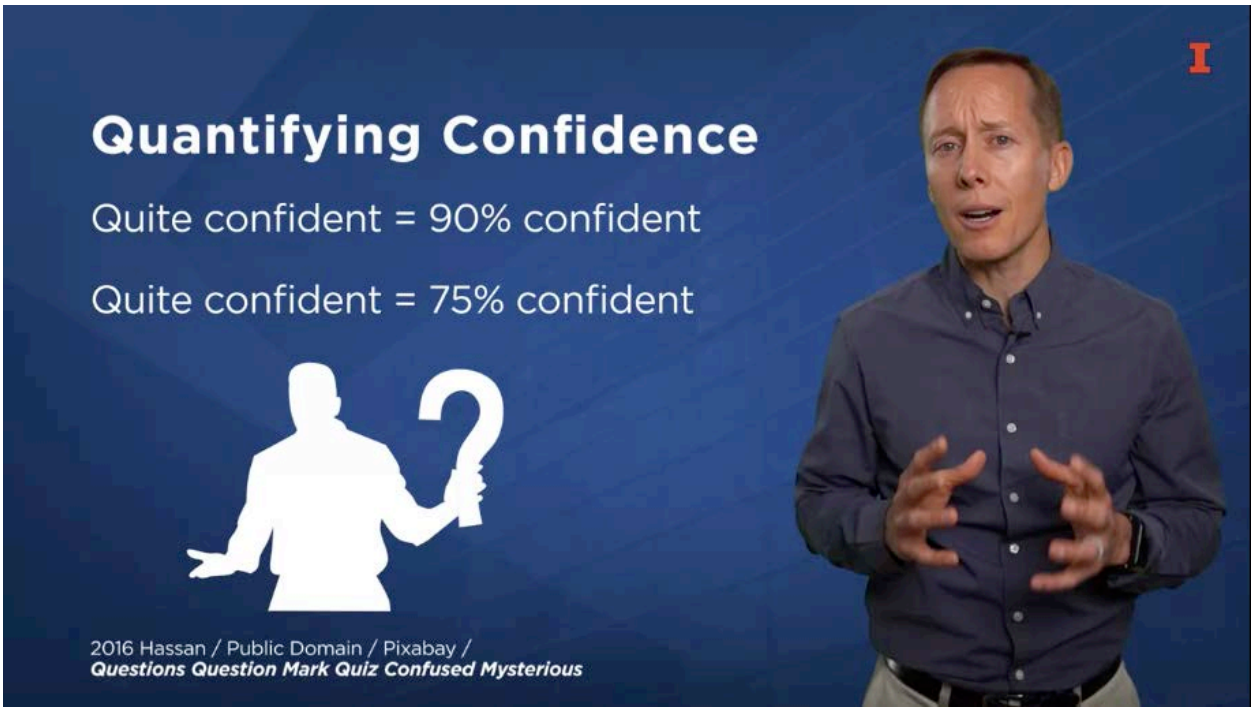


While EDA could potentially lead to actionable insights without using sophisticated algorithms, there are several reasons why it falls short.



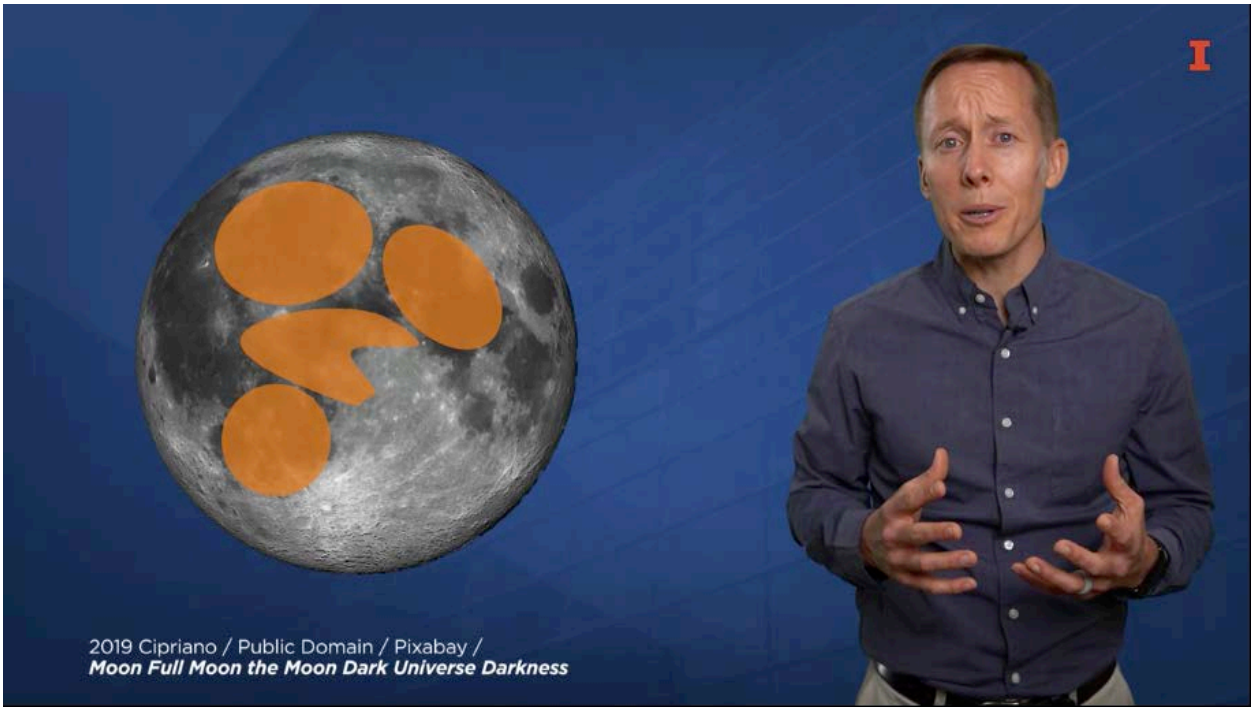
First, EDA is not scalable because obtaining actionable insight from EDA often requires a rewarding but labor-intensive process of filtering data with a variety of filters and visually explain the results on a variety of charts. While this may be feasible for a single small data set with a relatively small number of features or columns of data. It is not

feasible for a wide data set with many features, or for a large quantity of data sets. If you have a wide data set with many features, then identifying relationships, even simple ones, may require exploring distributions for many different columns or levels of data. At some point it's too much to keep track of without summarizing the results in a succinct way that allows you to systematically compare them. Of course, this process is even more difficult if you have a very complex relationship. EDA is also not scalable if you have a simple relationship that you want to explore with many different datasets. Can you imagine someone trying to compare even descriptive statistics for monthly sales data across all branches of a worldwide business that has hundreds or thousands of locations? The second reason why EDA is insufficient for obtaining and implementing actionable insight is because the insights are hard to quantify. Much exploratory data analysis is done with visualizations because data visualizations can quickly communicate trends and patterns. However, visualizations are usually not the best way to communicate specific numerical amounts. For example, a line chart makes it easy to see when something rises and falls, or when changes happen together but they do not quantify those with specific numbers.



Moreover, even if we could quantify patterns on charts with specific numbers, it's hard to quantify the confidence we have in those numbers. Using phrases like, I'm quite confident, for example, it could mean 90 percent confident to some people and only 75 percent confident to other people. So quantifying confidence is especially important in situations where we are motivated to find certain patterns. In those situations, we as humans can give selective attention to elements of visualizations or data analytic results. Quantifying confidence is especially important in situations where we're

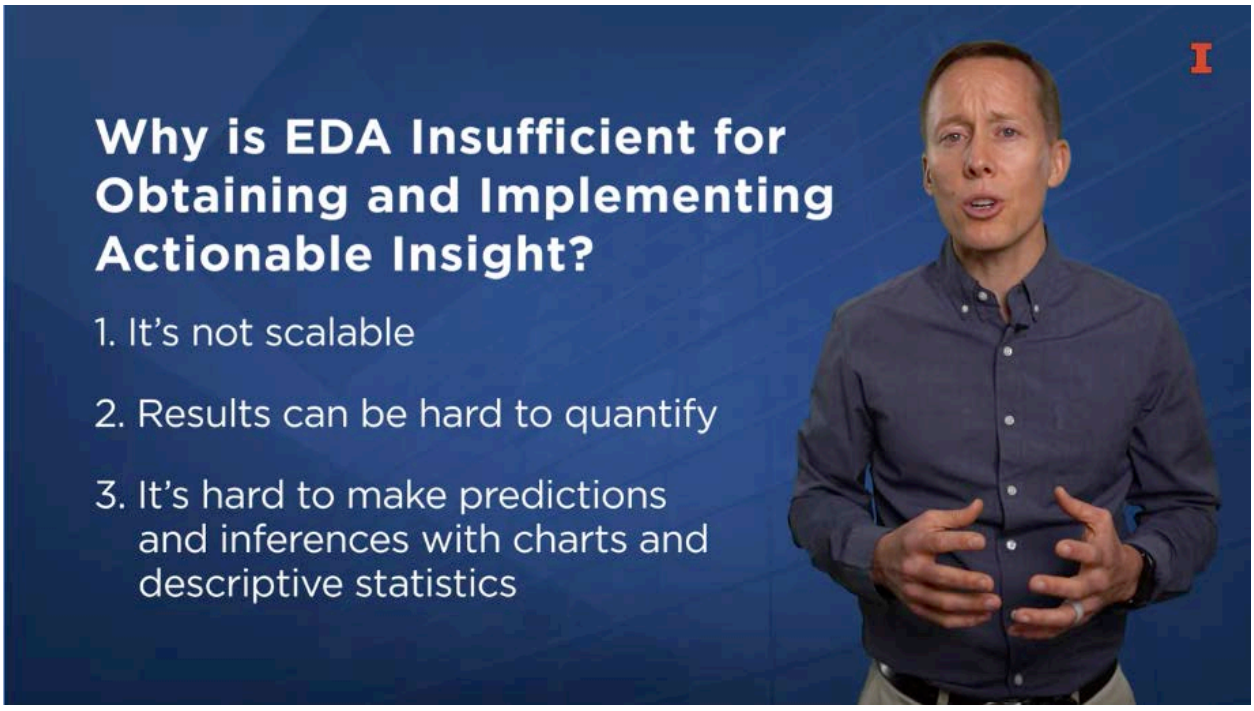
motivated to find certain patterns. In all situations then we as humans can give selective attention to elements of visualizations or data analytic results.



The man in the moon and the rabbit in the moon may be the most well-known and common examples of people giving selective attention to patterns. The craters and planets in a full moon can be interpreted to be a face or a rabbit.

But you have to decide to see them in that way by tilting your head or ignoring some other craters and planets.

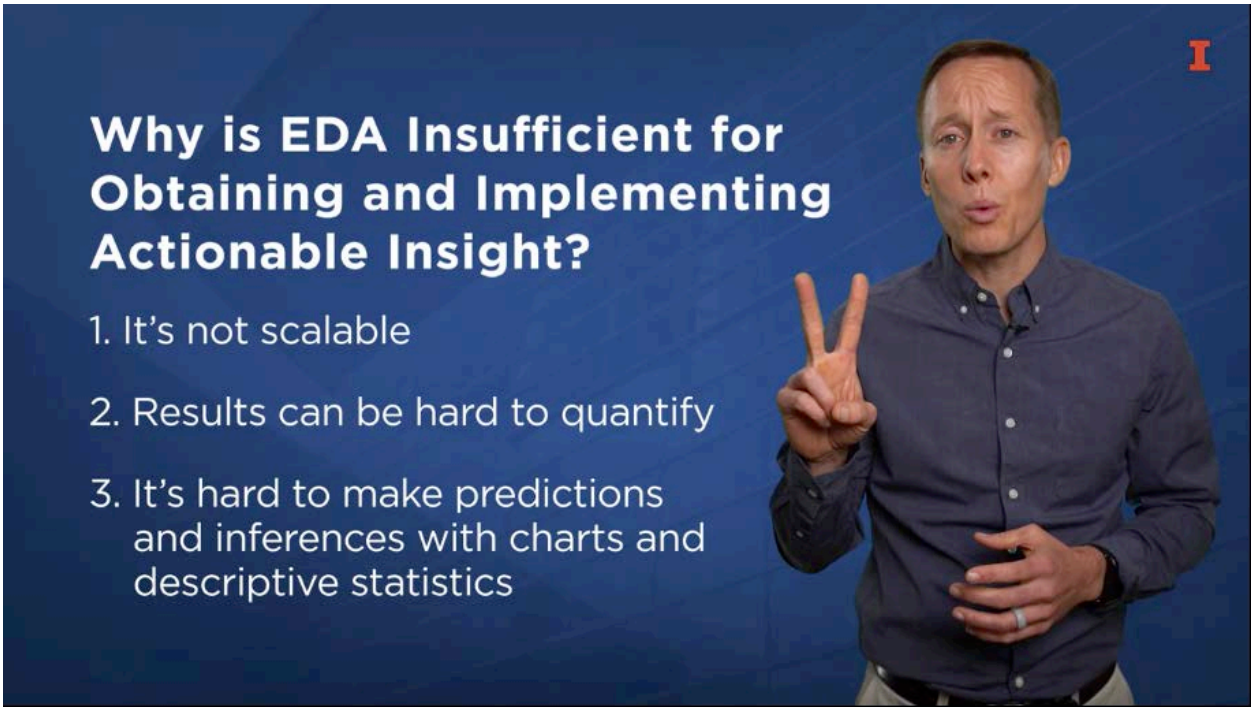
Similarly, if we are viewing data analytic results and we have an incentive to give more attention to some observations than others, then we may have more confidence in the existence of a certain pattern by ignoring information that runs counter to our desired results. Statistical techniques help us determine the extent to which we can place confidence in patterns because they are emotionless and we'll consider all of the observations.



The third reason why EDA is insufficient for obtaining and implementing actionable insight, is because the charts and descriptive statistics often associated with EDA are difficult to use for making predictions and inferences.

Some relationships are so complex that there are difficult to communicate with visualizations. Being able to take inputs and then process them into a point estimate is much easier to accomplish using a model, which is a mathematical representation of

relationships rather than a visualization. Models are especially helpful if you are using a computer to implement business decisions, which is often the case in business analytic environments. You can simply take inputs and with a few clicks of a keyboard, create a prediction that is based on a complex model.

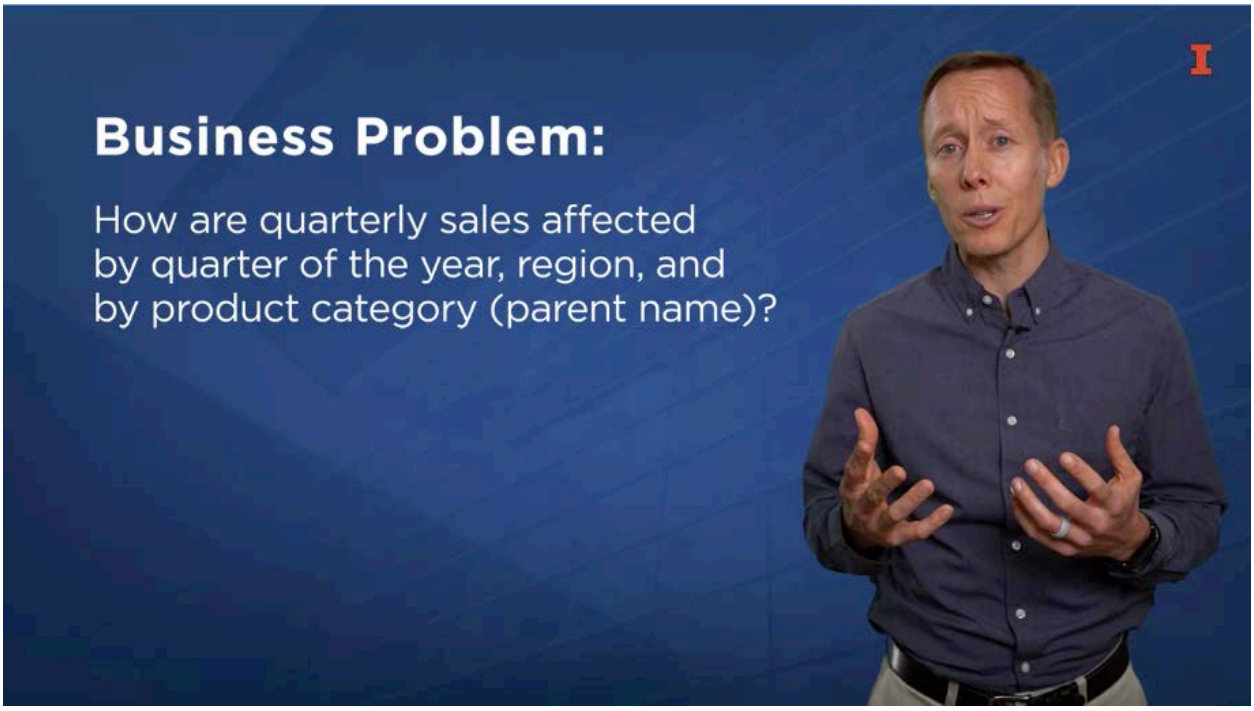


In some, while exploratory data analysis is an important part of the data analytic pipeline, it is often insufficient for obtaining and implementing actionable insight because, one, it is not scalable, two, quantifying point estimates and our confidence in them is difficult, and three, it is difficult to make predictions and inferences using visualizations and descriptive statistics.

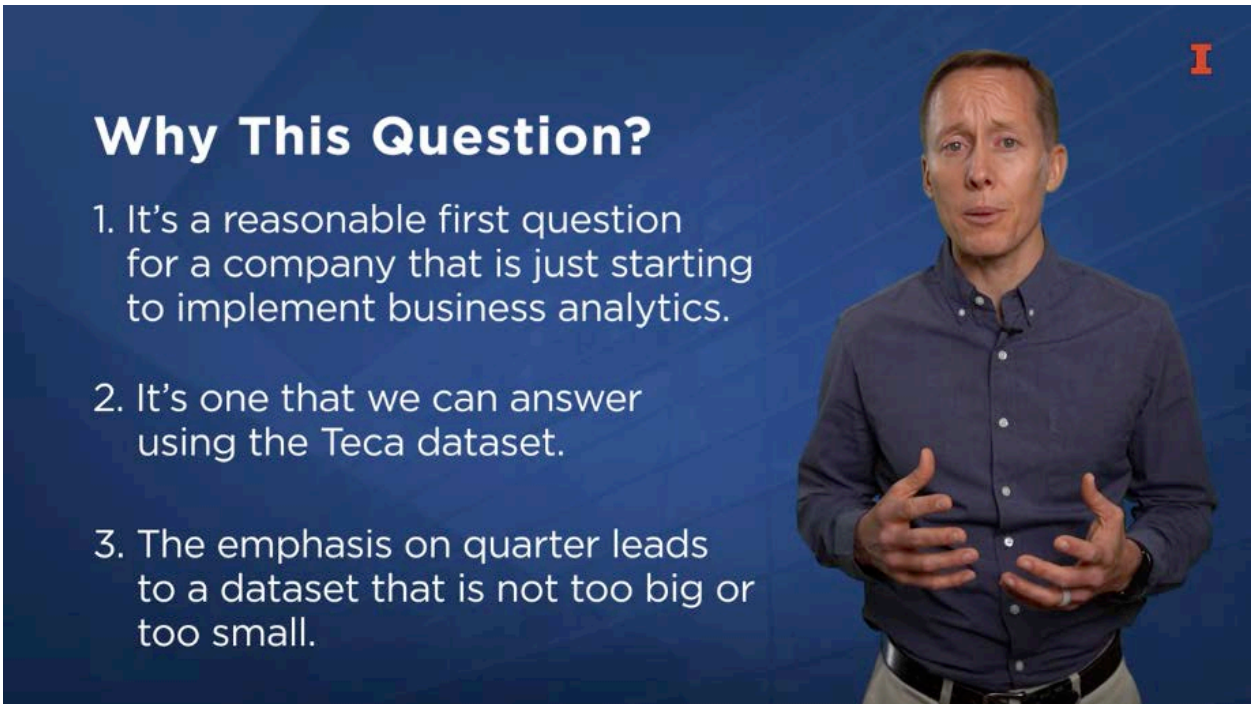

References:
Cipriano, S. (2019, August 15). Moon Full Moon the Moon Dark Universe Darkness
[Photograph]. https://pixabay.com/photos/moon-full-moon-the-moon-dark-4404540/
Hassan, M. (2018, May 17). Questions Question Mark Quiz Confused Mysterious
[Vector Graphic]. https://pixabay.com/vectors/questions-question-mark-quiz-3409194/

Lesson 1-3: Business Problem

Lesson 1-3.1 Business Problem



In this video, we want to briefly explain the business problem that we will be analyzing. Here's the question. How are a quarterly sales affected by quarter of the year, region, and product or category name? This is a broad question in the sense that we have not hypothesized any type of relationships such as our sales in the fourth quarter higher than in any of the other quarters or our quarterly sale is greatest when sales from the pop category exceed those from the lottery category.

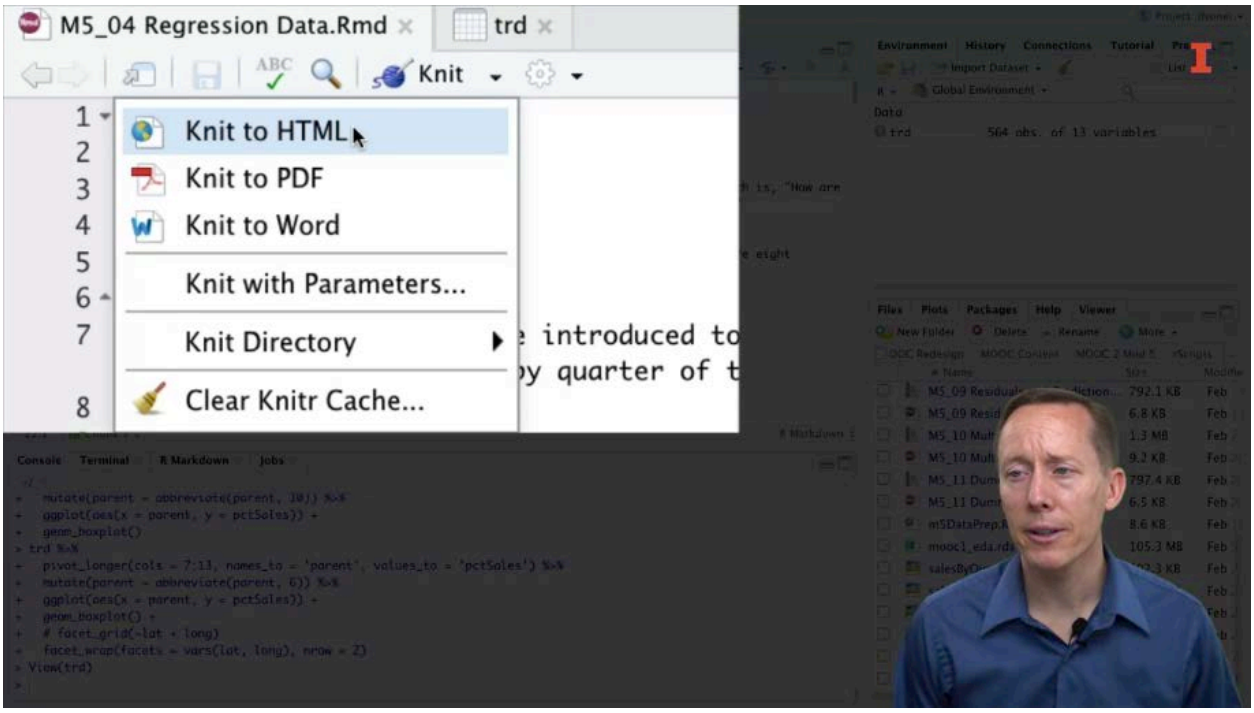


This broad question allows us to simulate elements of the data analytic process for a company that is just starting to use data analytic approaches. At this point, we want to gain some insight about what is driving our sales. But our ultimate hope is that this question will lead us to discover actionable insight. This actionable insight may be related to making predictions about sales so that we can order the right quantity of our product. In contrast, the actionable insight may be about better understanding the factors that drive quarterly sales, so that we can experiment with different strategies to increase sales. This question is one that we will be able to answer using the Teca dataset. This dataset is based on actual data from about 150 gas station and convenience stores in the Central United States. It's a rich dataset that includes information about what products are sold, the customers who buy the products, the location of each store, the time of day when the transaction occurred, and revenue and profit information. There are hundreds of different products, so each product is grouped into several categories that have a hierarchical relationship. Specifically, each product is one of many products in a category and each category is one of many categories in a parent category. We're going to focus on some of the parent categories. This question emphasizes quarter of the year rather than month, week, or day, because it is a time period that is not too big or too small. It's large enough to illustrate concepts related to correlation and regression by using visualizations. Using too many data points can make it difficult to visualize relationships with scatter plots, which is a visualization that we will use to illustrate these correlation and regression concepts. Thus, this dataset is small enough that it won't be too overwhelming. As we explore the data and search for results using correlation and regression, it will be easy to go off on a tangent to investigate deeper insights. That's not necessarily a bad thing, but since we don't have

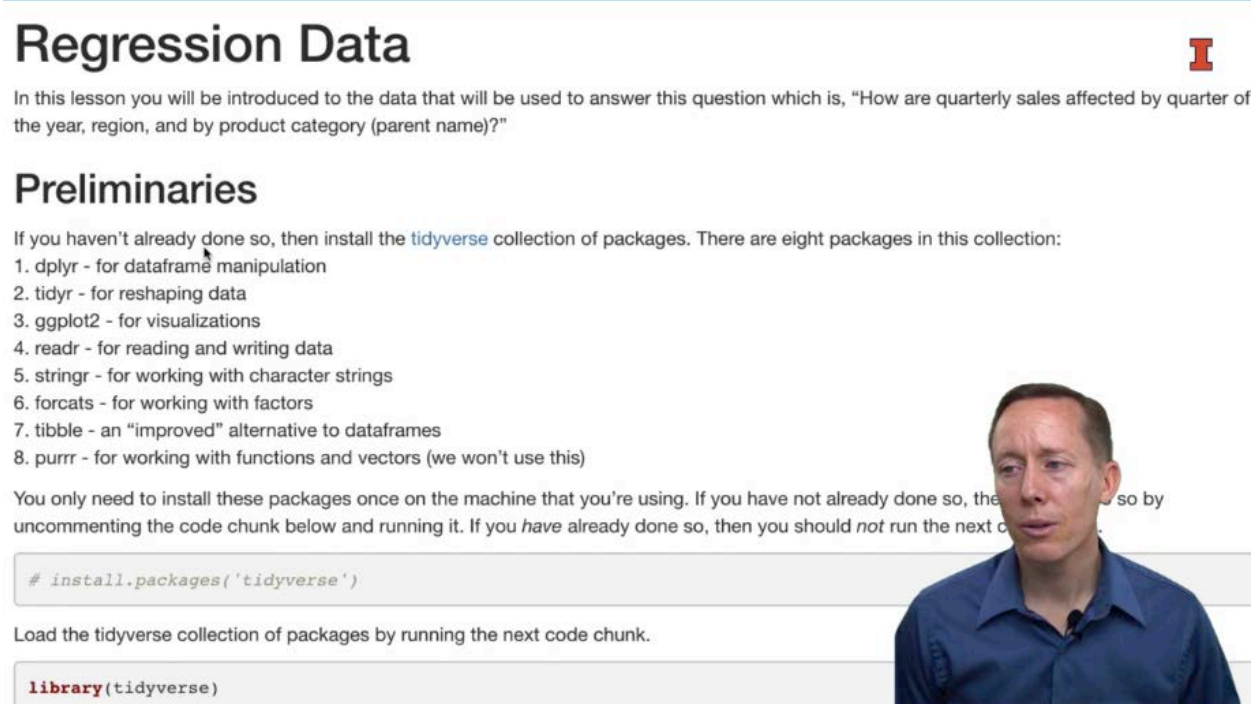
unlimited time, having a specific question like this will help us remember the main purpose of our analyses.

Lesson 1-3.2 Data

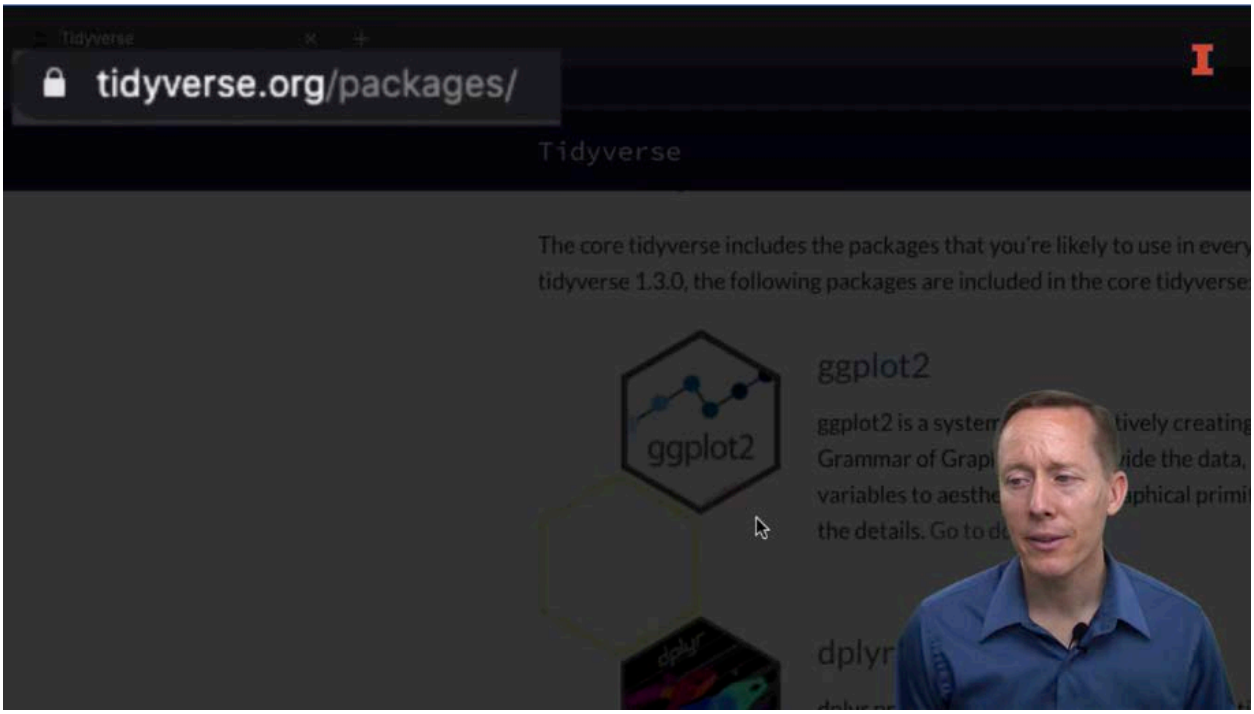
In this lesson, we will introduce you to the data set that we will use to answer our business question. Which is how our quarterly sales affected by a quarter of the year, geographic region and product category. So rather than just show you this data, we want to give you a process that you could use as a template for exploring the data before you use it in an advanced analytic algorithm regression. So there is a corresponding R markdown file that you can use and a data set that you can use to follow along with this video.



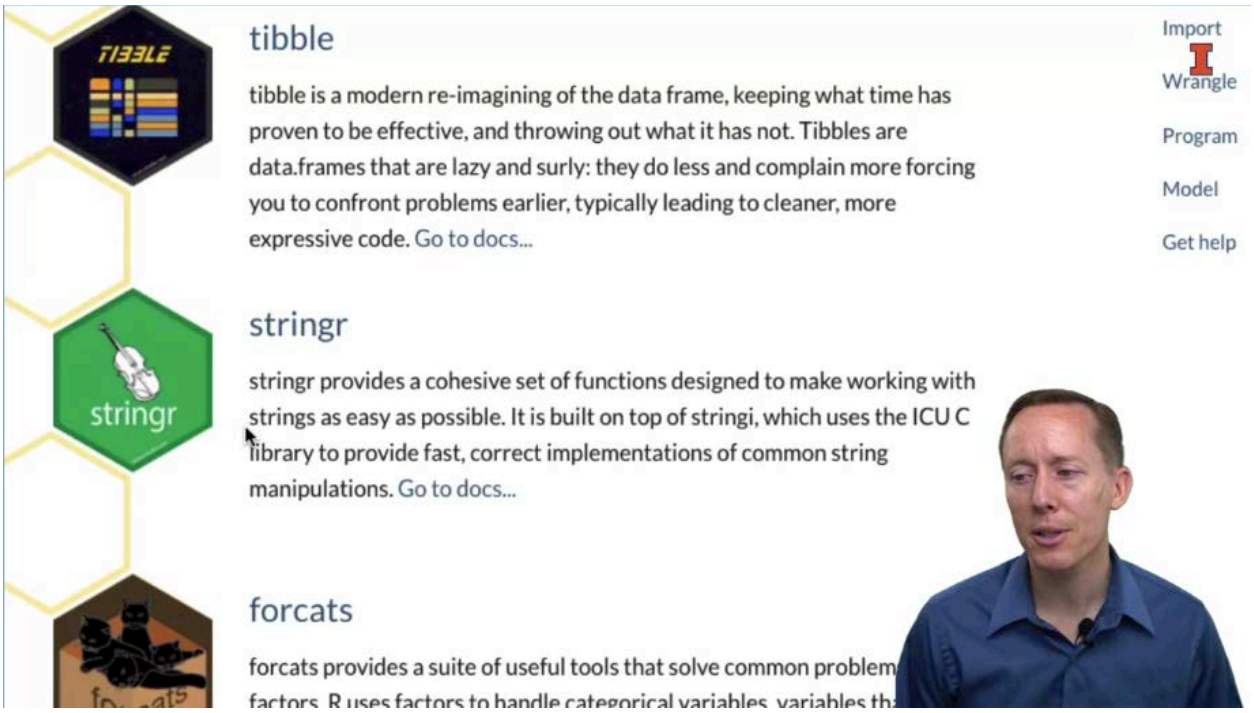
Now I've opened this R markdown file in our studio and then I've knit it together as an html file.



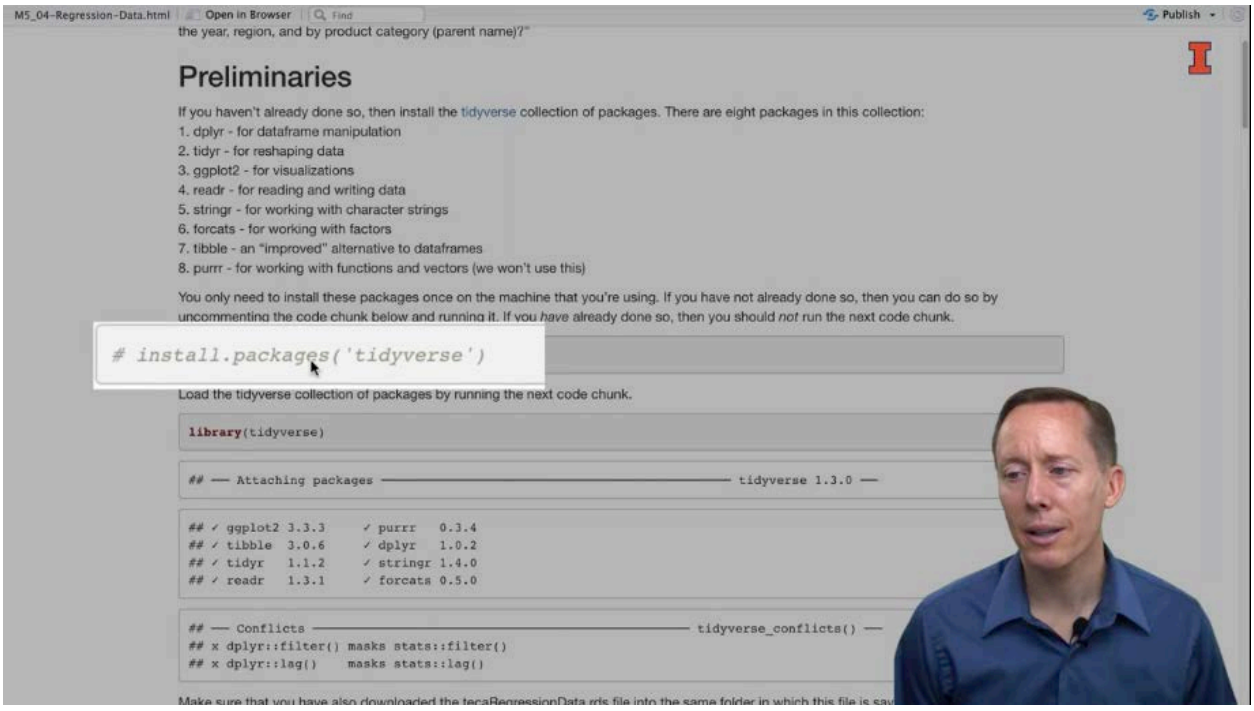
So I'm going to refer mostly to this html file. So some preliminaries are that you need to make sure that you have installed the tidyverse collection of packages. Now, as a quick reference to what those packages are they are listed right here.



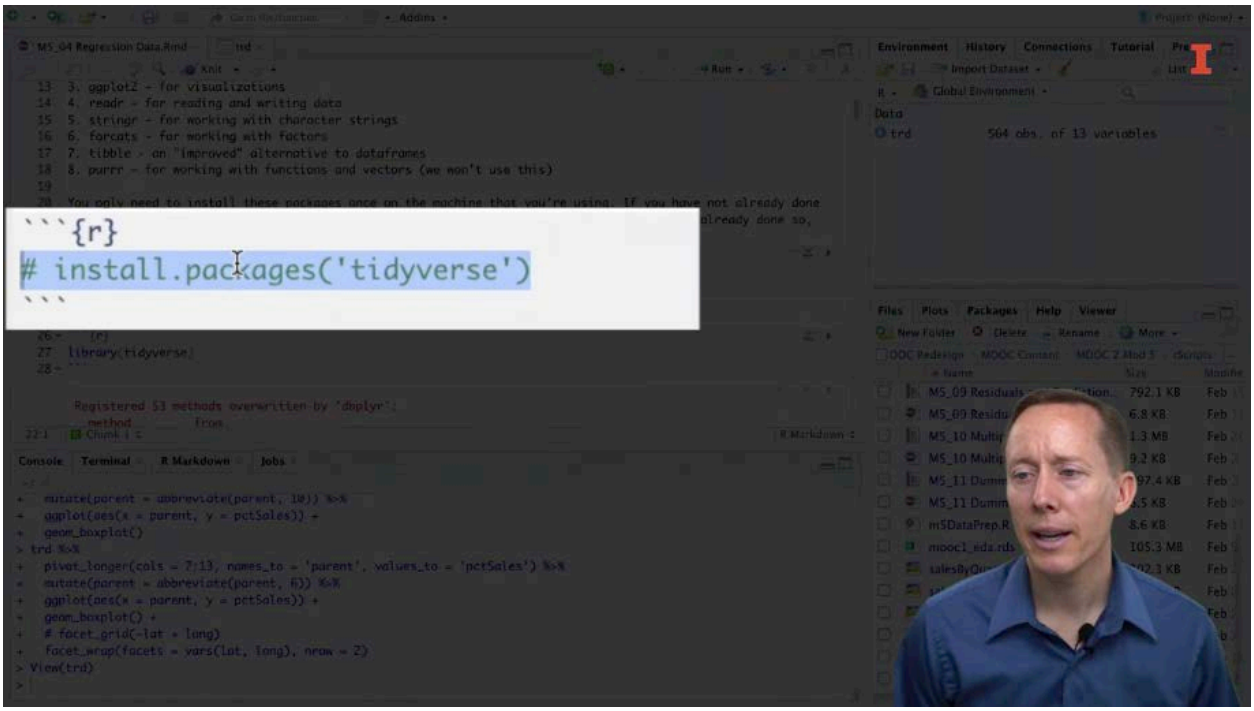
You can also click on this link which will take you to a website that will give you more detail of these packages real briefly, the ggplot2 package is for plotting dplry is used for data wrangling.

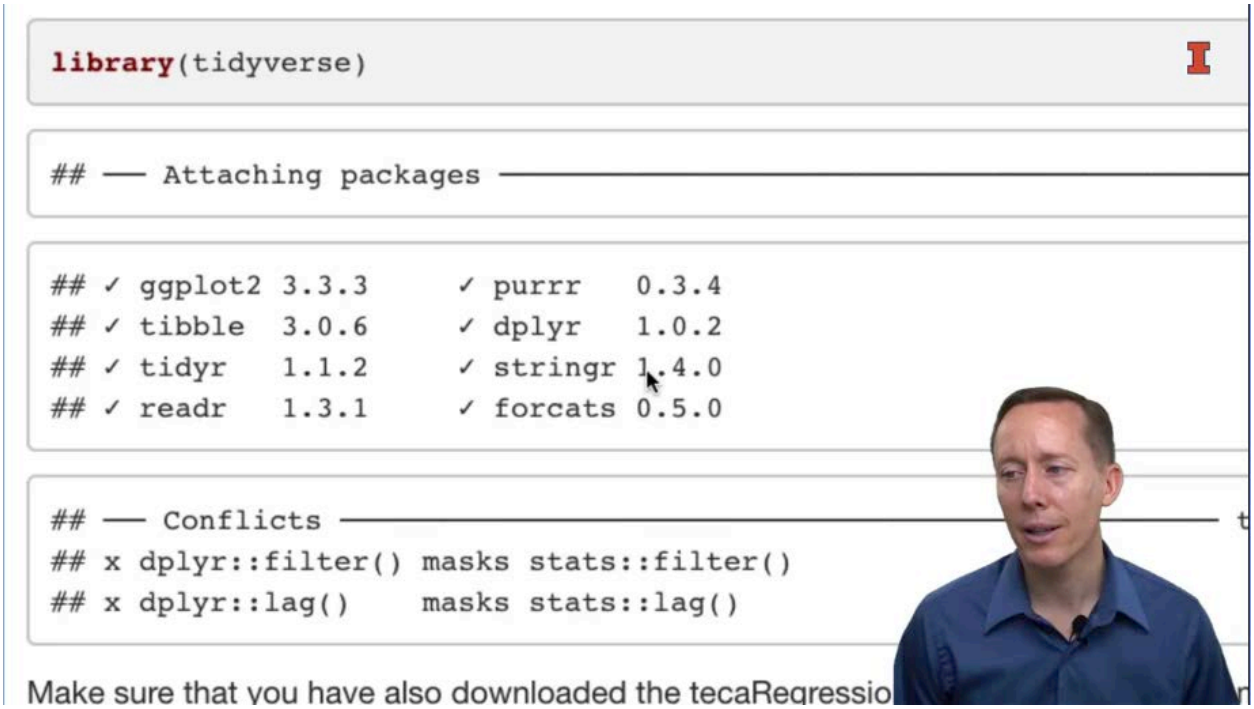


Tidyr is used for reshaping data. Readr is used for reading and writing data. Purr is used for working with functions and vectors. We won't use that one very much. Tibble is used for creating an improved data frames and you'll see this but we won't use it much really. Stringr is used for dealing with string data types and forcats is a collection of functions for dealing with factor data types.

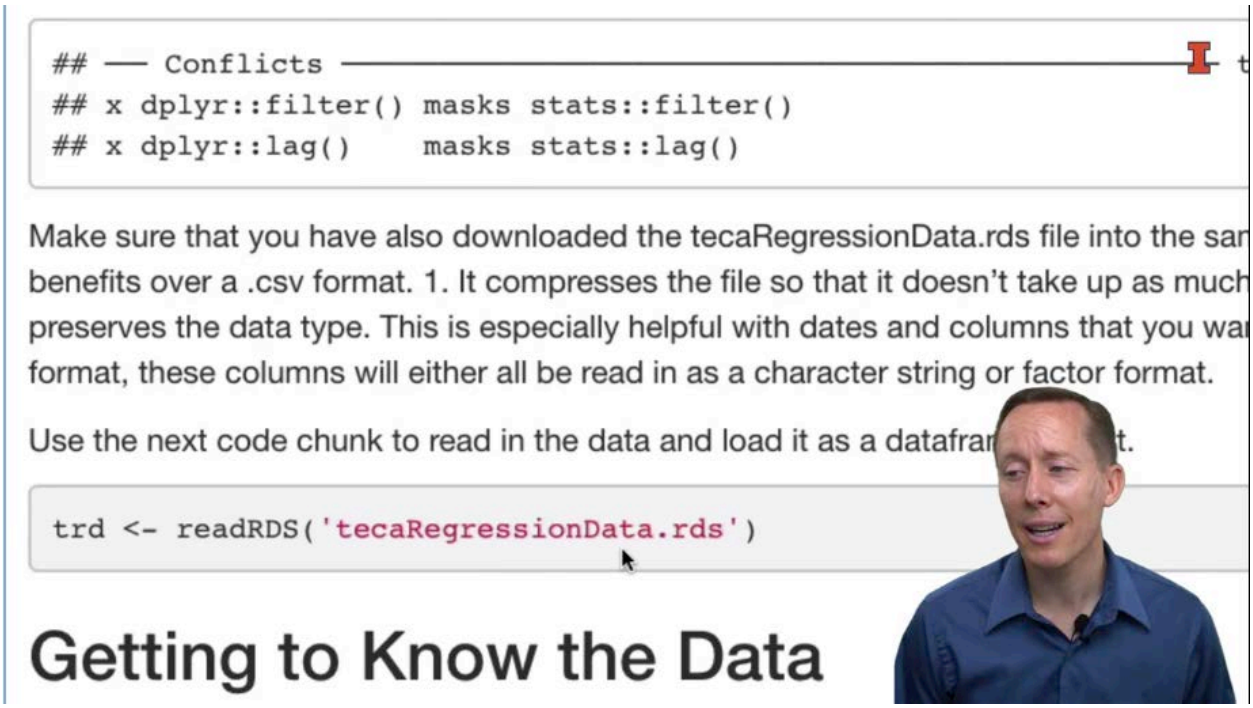


So if you have not already installed the tidyverse packages, you can do so by uncommenting out this code chunk in the R markdown file over.

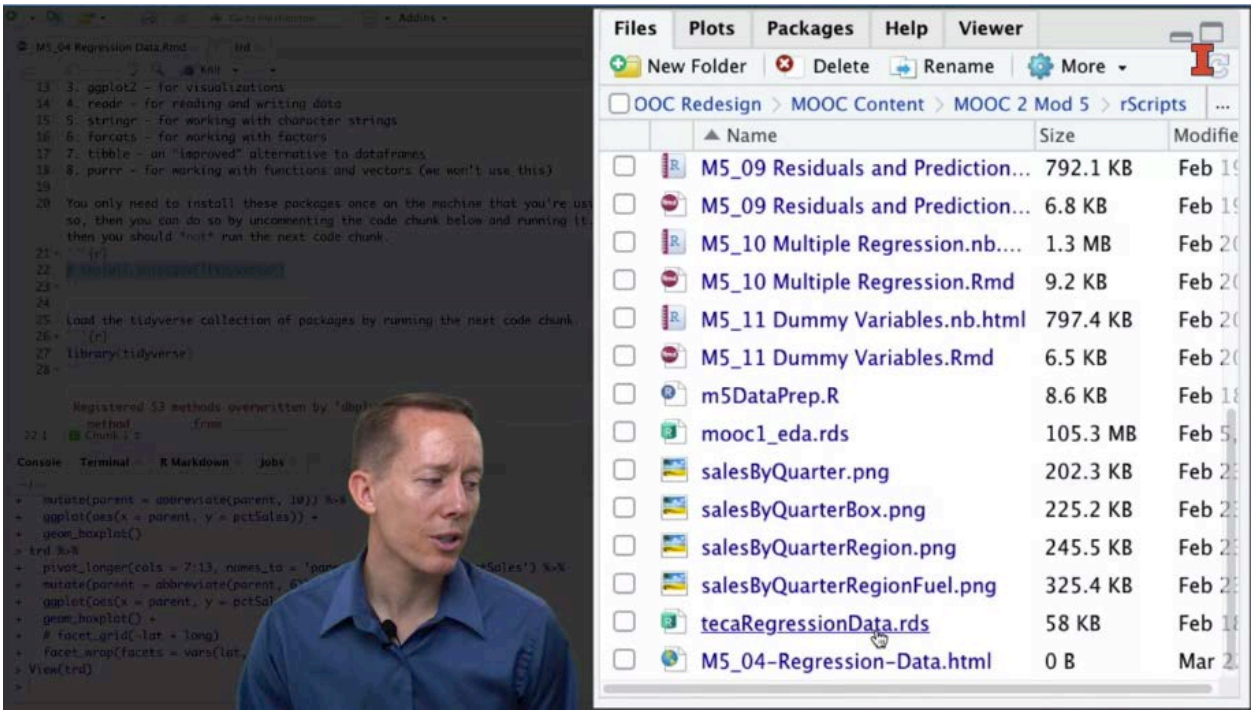




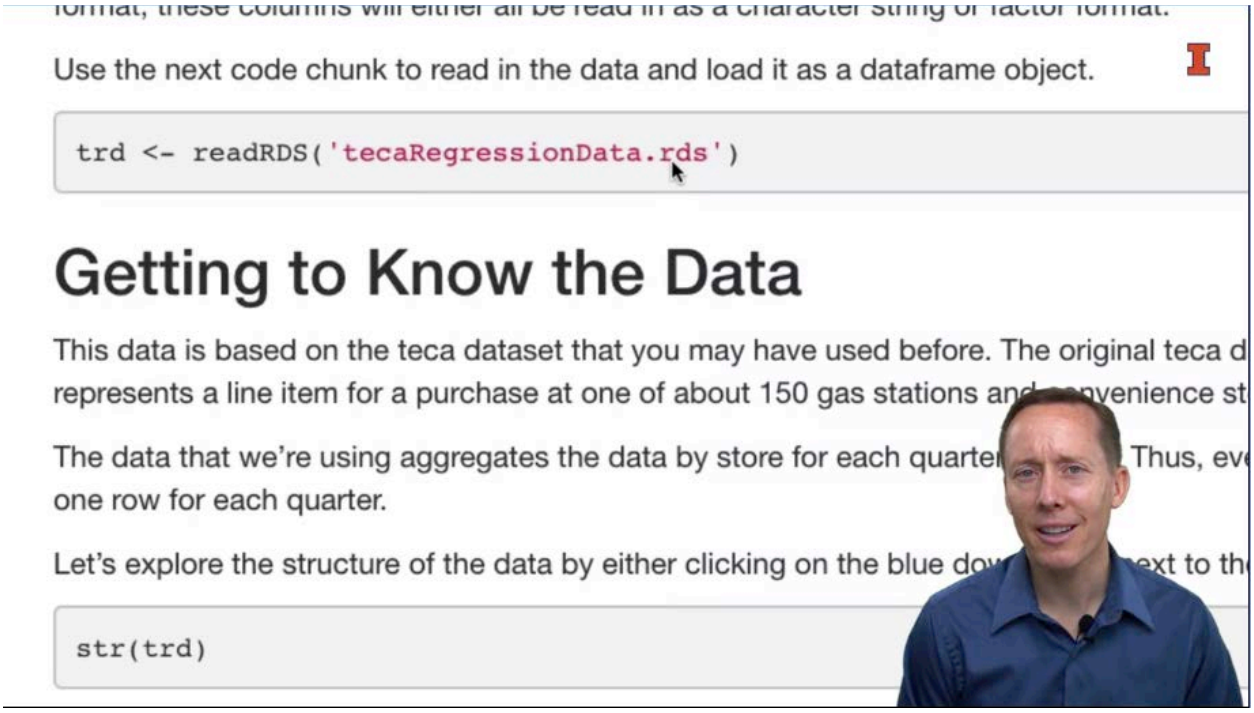
Here and then running that code chunk once you've installed that, you don't need to install it again. But you do need to load all of those packages and you can load all eight of those by simply using library (tidyverse). And you can see that the output indicates that they're all loaded and there are a few conflicts in different functions. But that shouldn't be much of a problem for us.



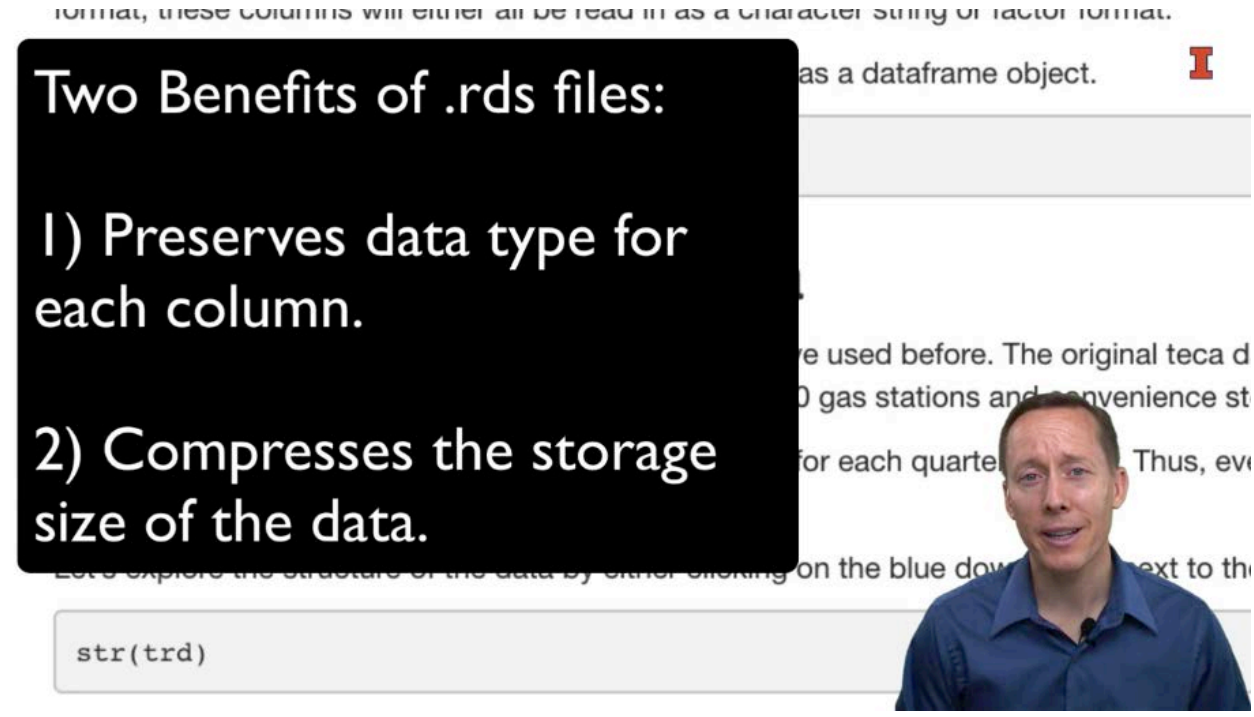
Now, also if you want to follow along, make sure to download the tecaRegressionData.rds file and store it in the same folder that the R markdown file is stored in.



So you can see on my machine, I've got the M504 regression Data.Rmd file in the same folder as this tecaRegressionData.rds file.

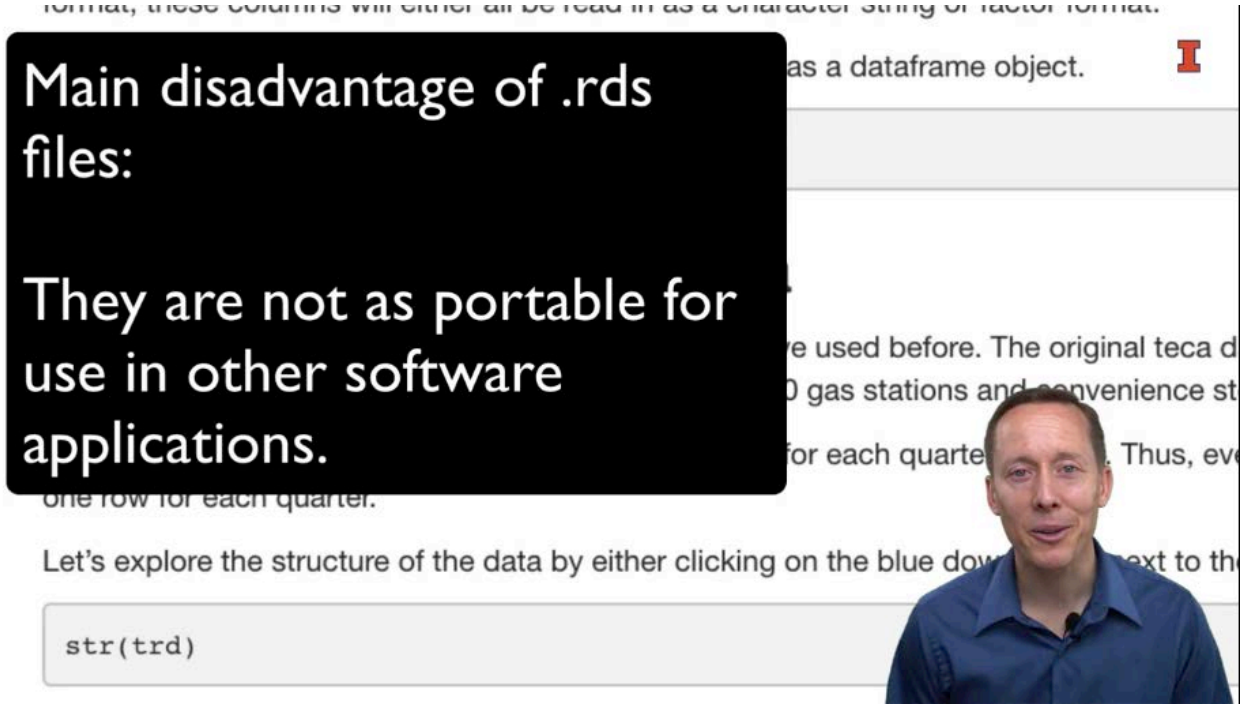


Now if you do that everything else you should be able to follow along in this video. So we need to read in this data and we'll use the readRDS function since it's an rds data set and by the way, what is an rds dataset? RDS is in R data structure and there are two main benefits over CSV files.



The first is that rds files preserve the column type. So the data type for each column is saved as either a character string or numeric or factor or date, all right. Rather than a

CSV file, which stores everything as a string basically. So we don't have to reconvert the columns to the correct data type every time we read in the data. Now, this rds file also compresses the data so it doesn't take up as much room as a CSV file.

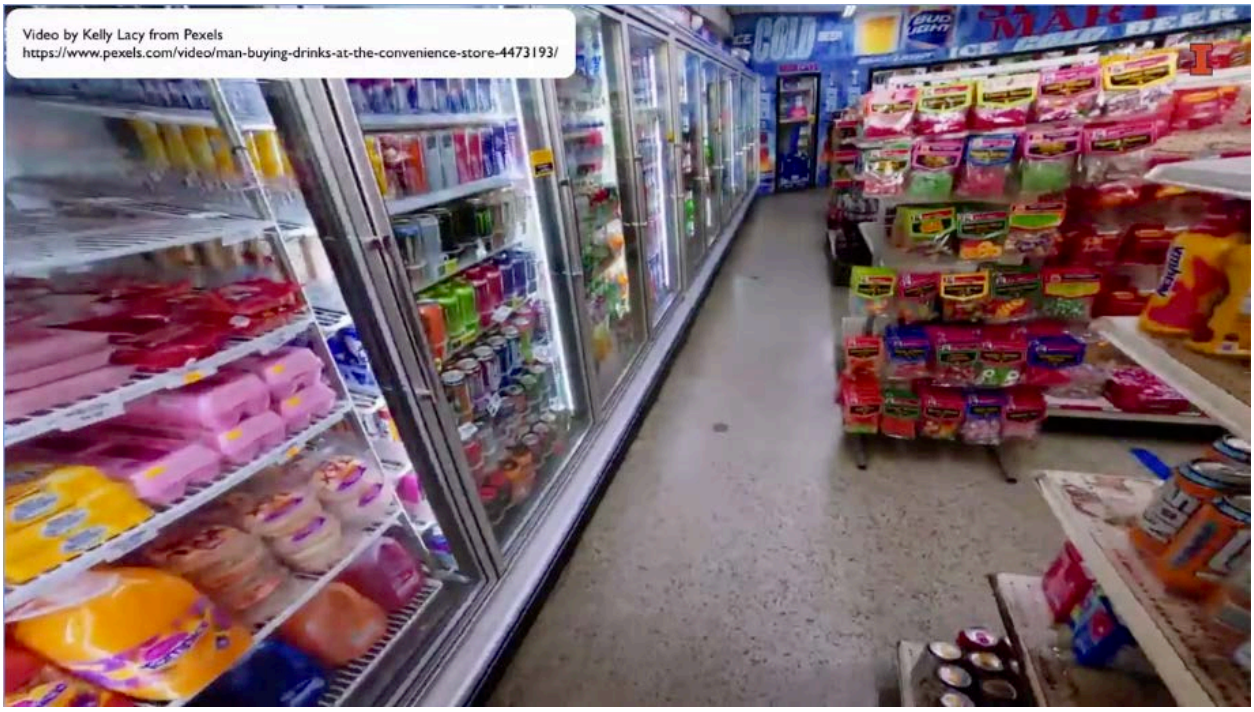


The main downside is that it is not as portable because it can't be used by python or power bi or something else, for instance.

```
str(trd)
```

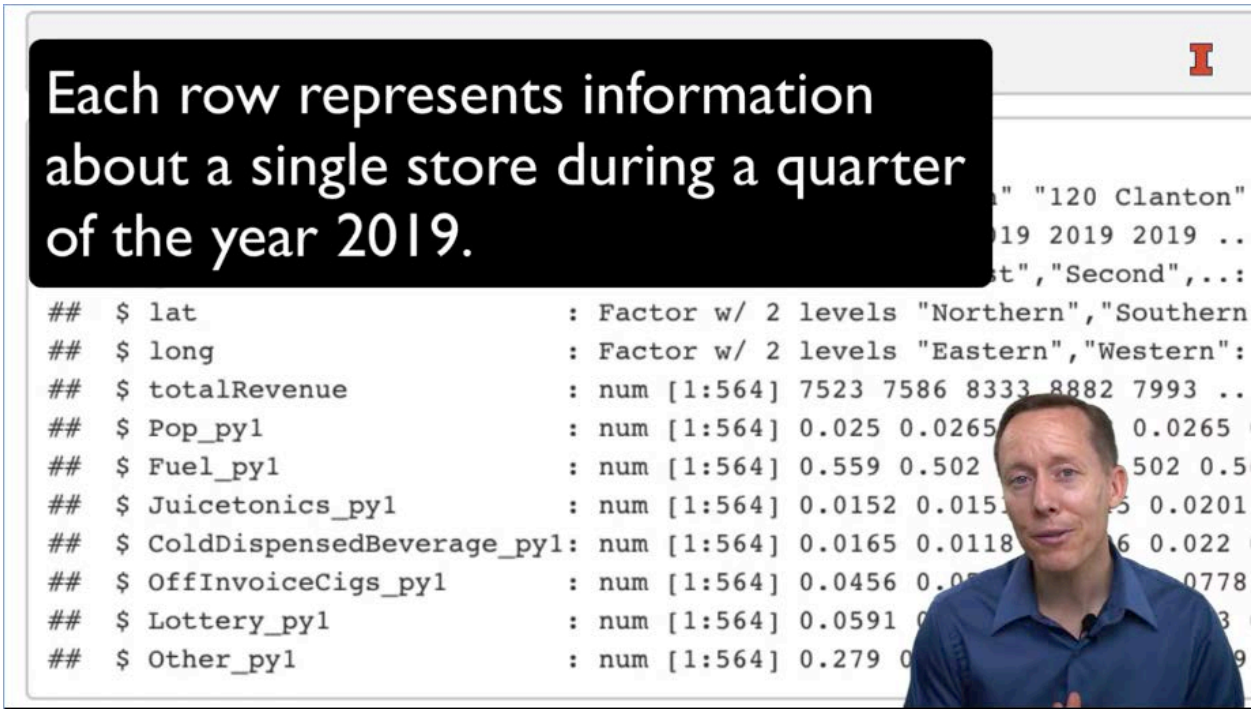
```
## tibble [564 × 13] (S3: tbl_df/tbl/data.frame)
##  $ site_name                 : chr [1:564] "120 Clanton" "120 Clanton"
##  $ quarter                   : num [1:564] 2019 2019 2019 2019 2019 ..
##  $ quarterNoYear             : Factor w/ 4 levels "First","Second",..:
##  $ lat                       : Factor w/ 2 levels "Northern","Southern
##  $ long                      : Factor w/ 2 levels "Eastern","Western":
##  $ totalRevenue              : num [1:564] 7523 7586 8333 8882 7993 ..
##  $ Pop_py1                   : num [1:564] 0.025 0.0265 0.0265
##  $ Fuel_py1                  : num [1:564] 0.559 0.502 0.502 0.5
##  $ Juicetonics_py1           : num [1:564] 0.0152 0.015 0.0201
##  $ ColdDispensedBeverage_py1: num [1:564] 0.0165 0.0118 0.022
##  $ OffInvoiceCigs_py1        : num [1:564] 0.0456 0.0778
##  $ Lottery_py1               : num [1:564] 0.0591
##  $ Other_py1                 : num [1:564] 0.279
```

But anyway, we will read in this data and save it as trd. So that's a data frame trd. And now let's go ahead and explore the data. Now first I'll just give you an explanation, a verbal explanation and then we'll use some functions in our studio.

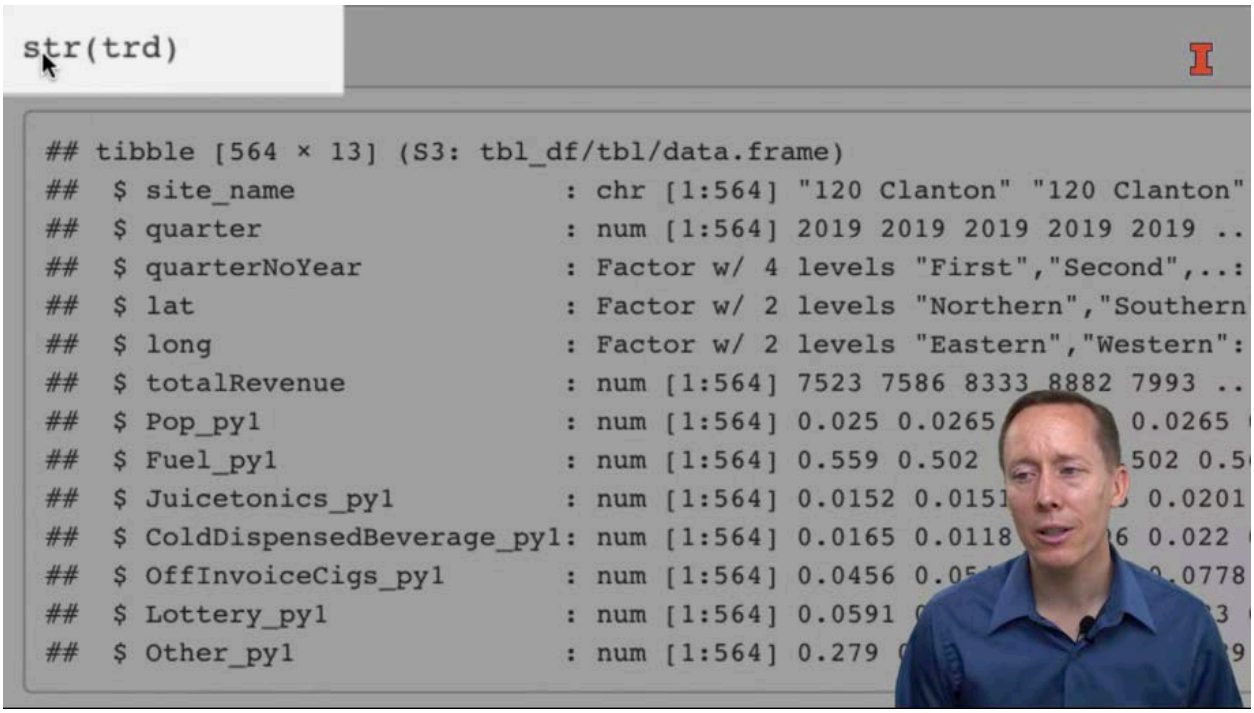


So Teca is the fictitious name of an actual company that operates about 150 convenience stores throughout the center of the United States.

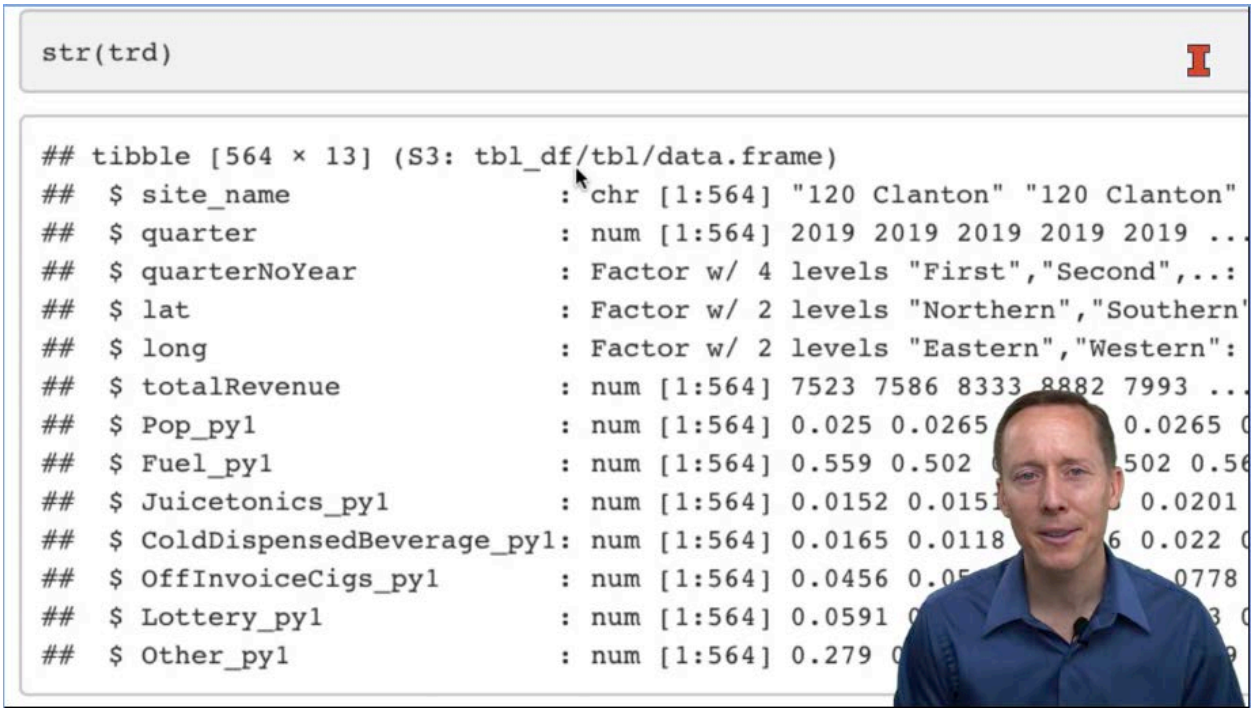
Now the original teca data is point of sale data and it has one row for every item of a transaction.



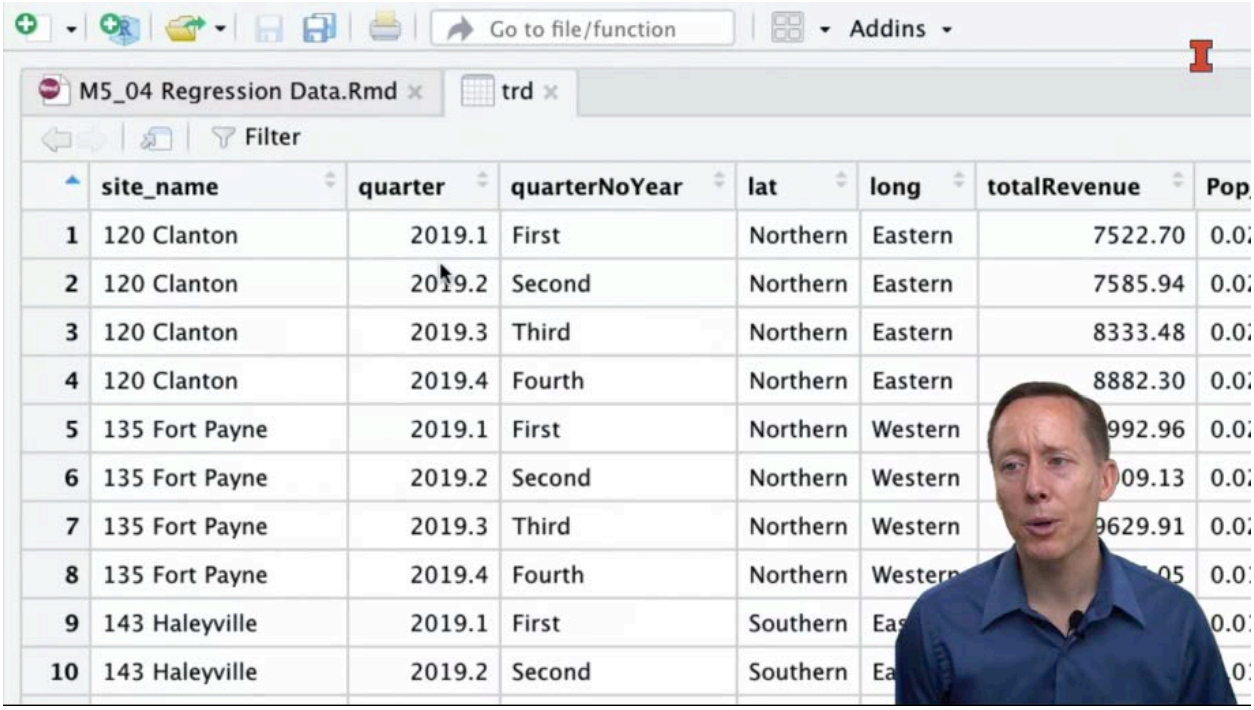
Now what we're looking at here is based on a sample of data which has been aggregated such that there is one observation for each store for every quarter of the year 2019, all right. So there's a quick verbal explanation.



Let's let's explore it using some functions in our so let's look at the structure of this data frame by using the str command.



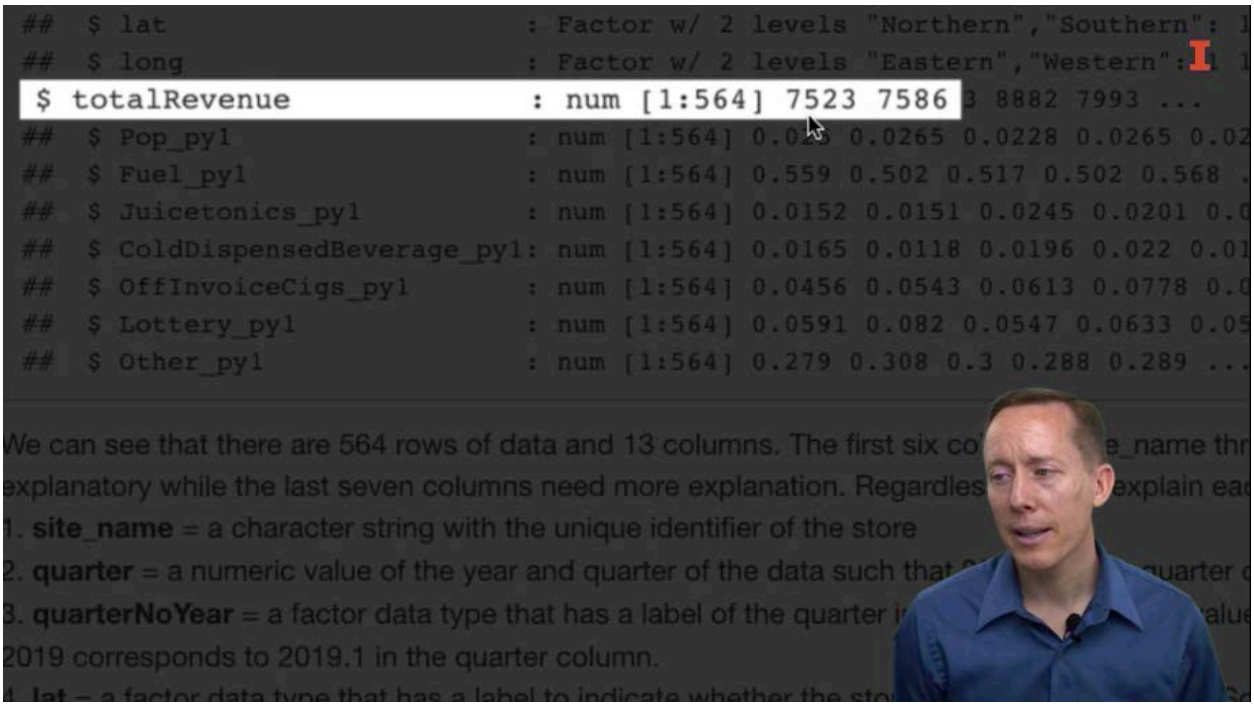
We can see that it's stored as a table, which is an advanced data frame and it has some other ways that it could be used as well, but not important for us at this point. The first six columns of data are pretty self-explanatory. So we've got site name, which is a name for the store quarter, which is actually not an integer it's a numeric value.



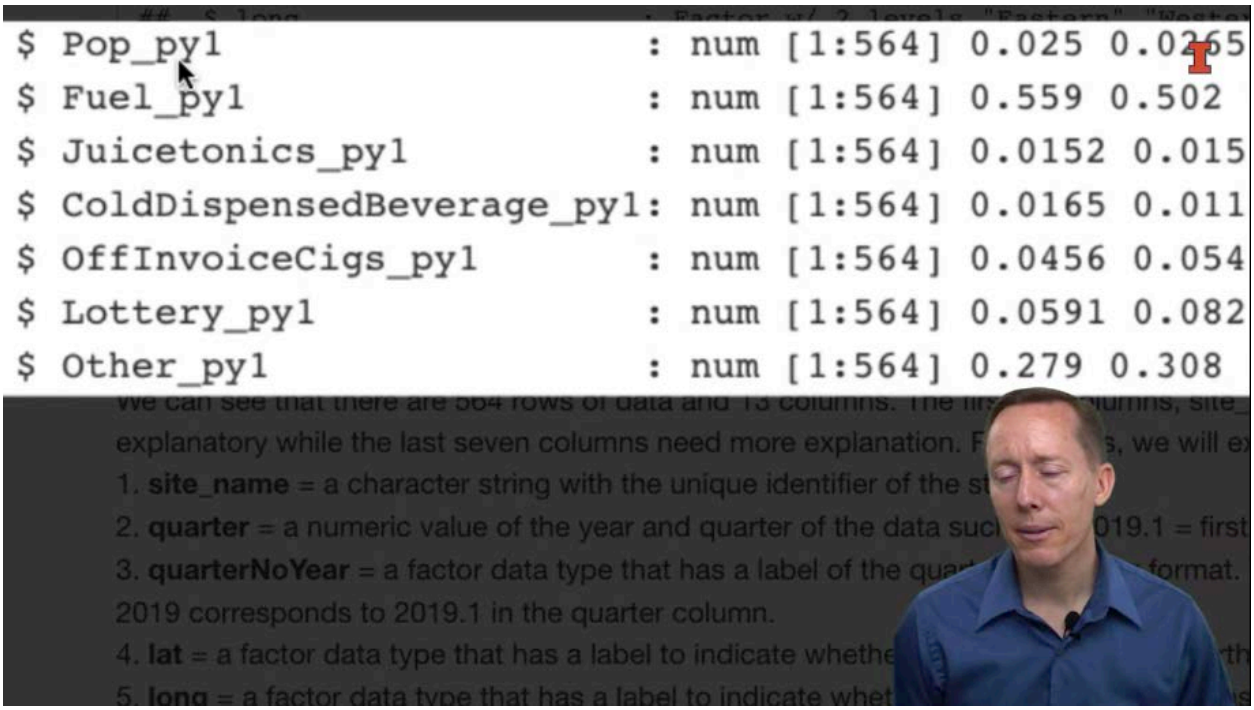
	site_name	quarter	quarterNoYear	lat	long	totalRevenue	Pop
1	120 Clanton	2019.1	First	Northern	Eastern	7522.70	0.0
2	120 Clanton	2019.2	Second	Northern	Eastern	7585.94	0.0
3	120 Clanton	2019.3	Third	Northern	Eastern	8333.48	0.0
4	120 Clanton	2019.4	Fourth	Northern	Eastern	8882.30	0.0
5	135 Fort Payne	2019.1	First	Northern	Western	992.96	0.0
6	135 Fort Payne	2019.2	Second	Northern	Western	09.13	0.0
7	135 Fort Payne	2019.3	Third	Northern	Western	9629.91	0.0
8	135 Fort Payne	2019.4	Fourth	Northern	Western	05	0.0
9	143 Haleyville	2019.1	First	Southern	Eas		0.0
10	143 Haleyville	2019.2	Second	Southern	Ea		0

And I will actually switch over to our studio and show you some of the rows of data here. You can see that quarter, we've got the year and then a period and then an integer to represent the quarter of the year. So 2019.1 corresponds to the first quarter of 2019, 2019.2 corresponds to the second quarter and so on, all right. And then quarterNoYear is a factor data type. That is first second third or fourth.

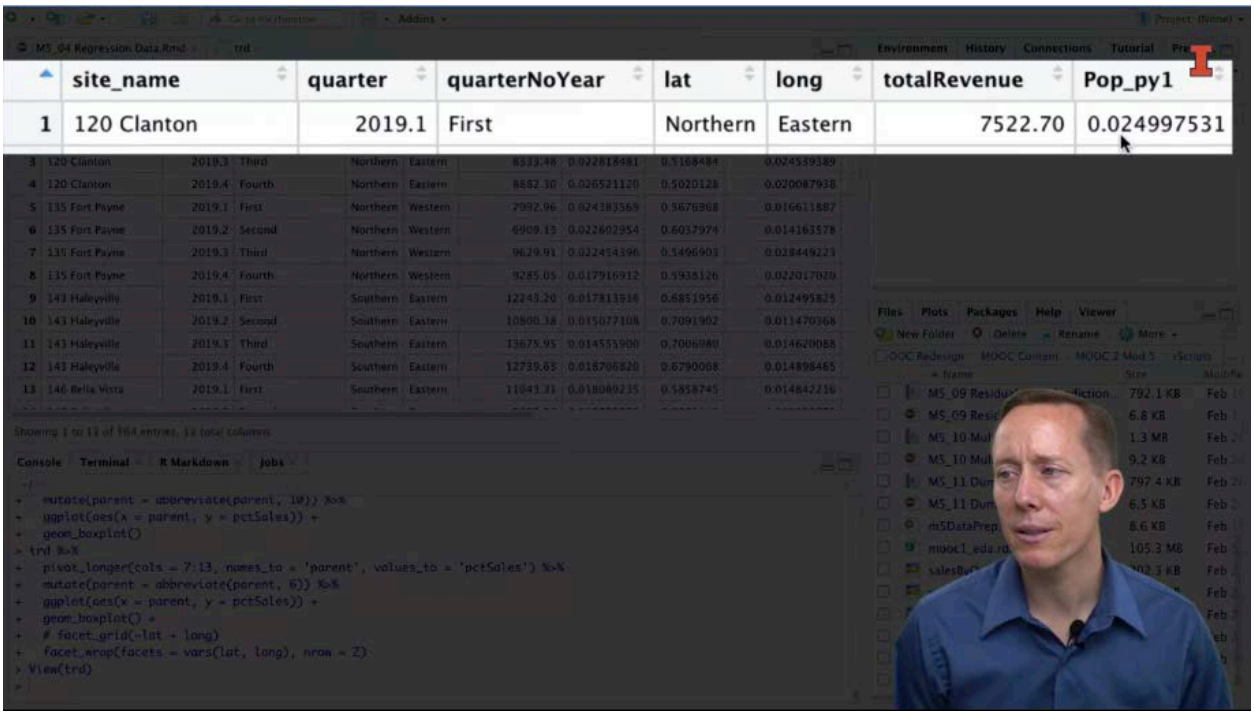
Alright, then we've got the lat and long columns and these are not actually coordinates their factors. And you can see that lat only has two levels to it Northern or Southern. So we've divided the data in half based on half of the data that in the northernmost area, half in the southernmost area. And similarly with long half the data is in the eastern region, half as in the western region, all right.



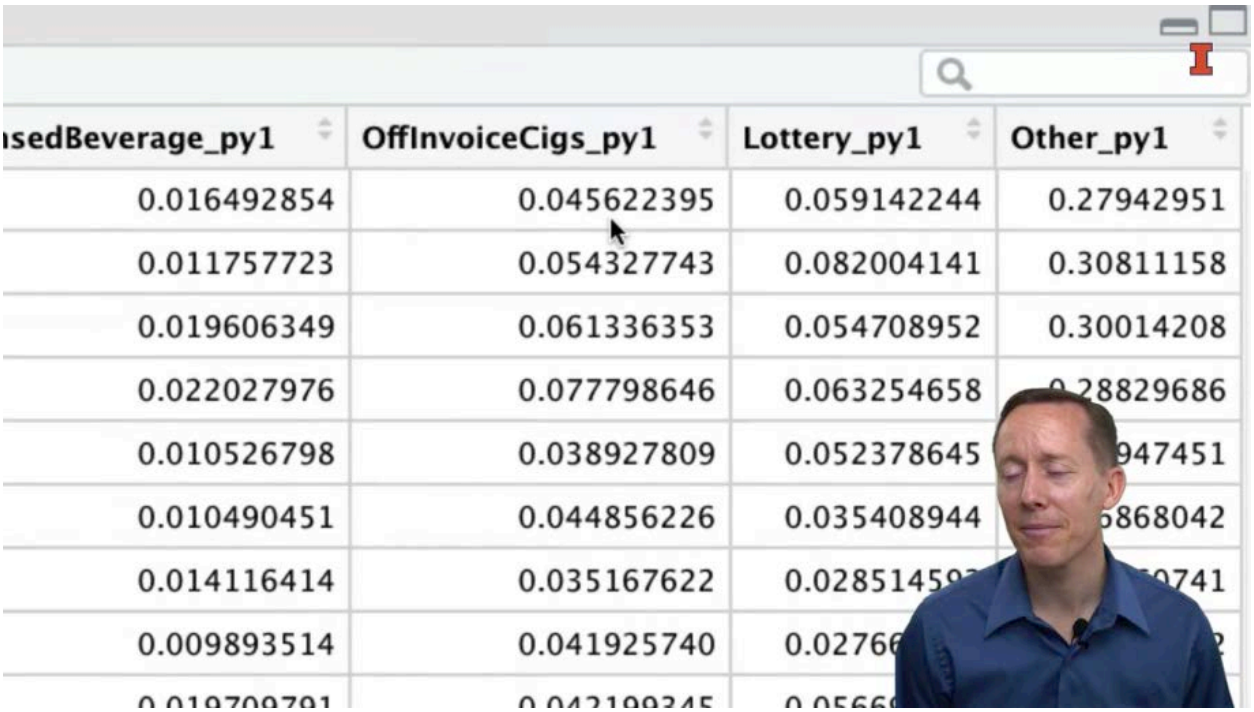
And then we've got total revenue, which is the total amount of revenue for that quarter and that store. And this is based on a sample of all of the observations. So that's why these numbers seem pretty low.



Now these last seven columns may not be as self evident, but pop _ py 1 refers to the proportion of sales that came from pop during one year prior to the same quarter, one year prior.



So actually let's go over to the data frame itself. This 0.025 means that for this store on 120 Clanton during the first quarter of 2019. Last year, during the first quarter of 2018 2.5% of the total revenue came from pop.

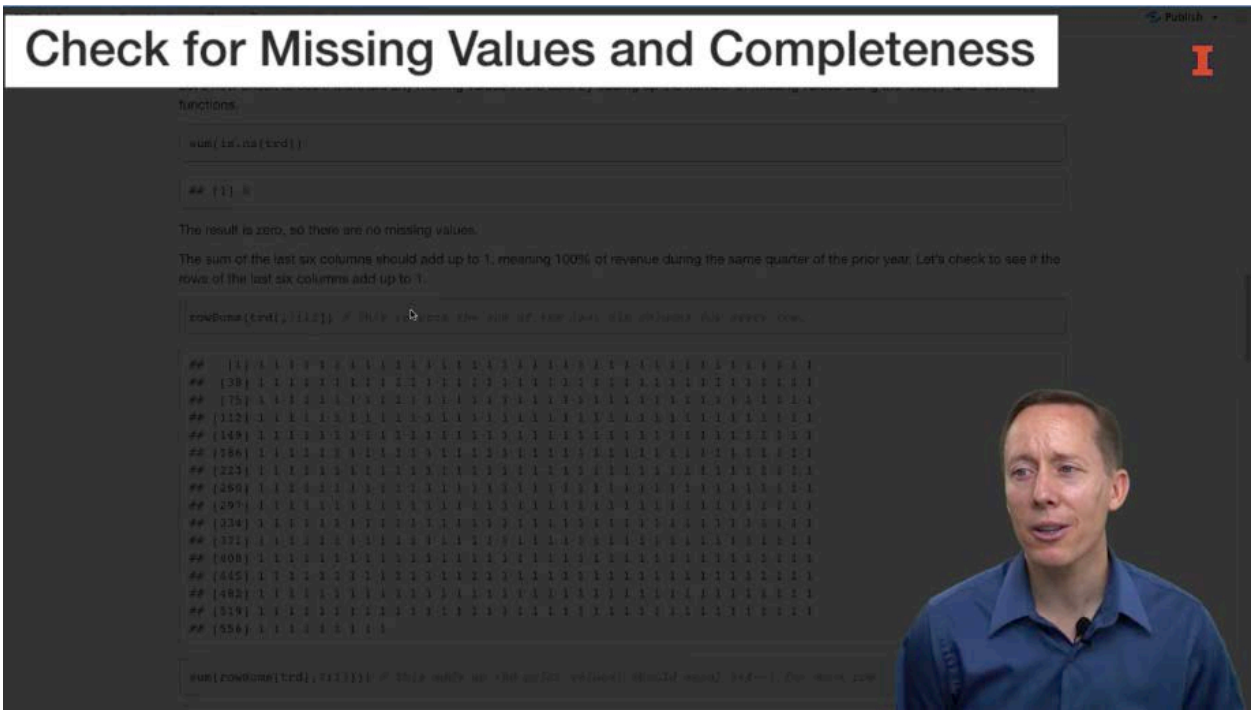


Versus 55.9% from fuel, 1.5% from Juicetonics, 1.6 from coal, dispense beverages, 4.6% from offinvoiceCigs, 5.9% from lottery and then 27.9% from other everything else, All right.

```
str(trd)
```

```
## tibble [564 × 13] (S3: tbl_df/tbl/data.frame)
##  $ site_name                 : chr [1:564] "120 Clanton" "120 Clantc
##  $ quarter                   : num [1:564] 2019 2019 2019 2019 2019
##  $ quarterNoYear             : Factor w/ 4 levels "First","Second",.
##  $ lat                       : Factor w/ 2 levels "Northern","Southe
##  $ long                      : Factor w/ 2 levels "Eastern","Western
##  $ totalRevenue              : num [1:564] 7523 7586 8333 8882 7993
##  $ Pop_py1                   : num [1:564] 0.025 0.02 228 0.026
##  $ Fuel_py1                  : num [1:564] 0.559 0.50 0.502 0
##  $ Juicetonics_py1           : num [1:564] 0.0152 0.0 245 0.02
##  $ ColdDispensedBeverage_py1: num [1:564] 0.0165 0.01 0196 0.02
##  $ OffInvoiceCigs_py1        : num [1:564] 0.0456 0 13 0.07
##  $ Lottery_py1               : num [1:564] 0.059 063
##  $ Other_py1                 : num [1:564] 0.27 .2
```

So, there's a quick overview the structure of the data set, 564 rows 13 columns. Now, let's go ahead and just check the quality of this data set.



So let's check for missing values and completeness.

```
sum(is.na(trd))
```

```
## [1] 0
```

The result is zero, so there are no missing values.

The sum of the last six columns should add up to [illegible]eaning rows of the last six columns add up to 1.

```
rowSums(trd[,7:13]) # This returns [illegible]
```

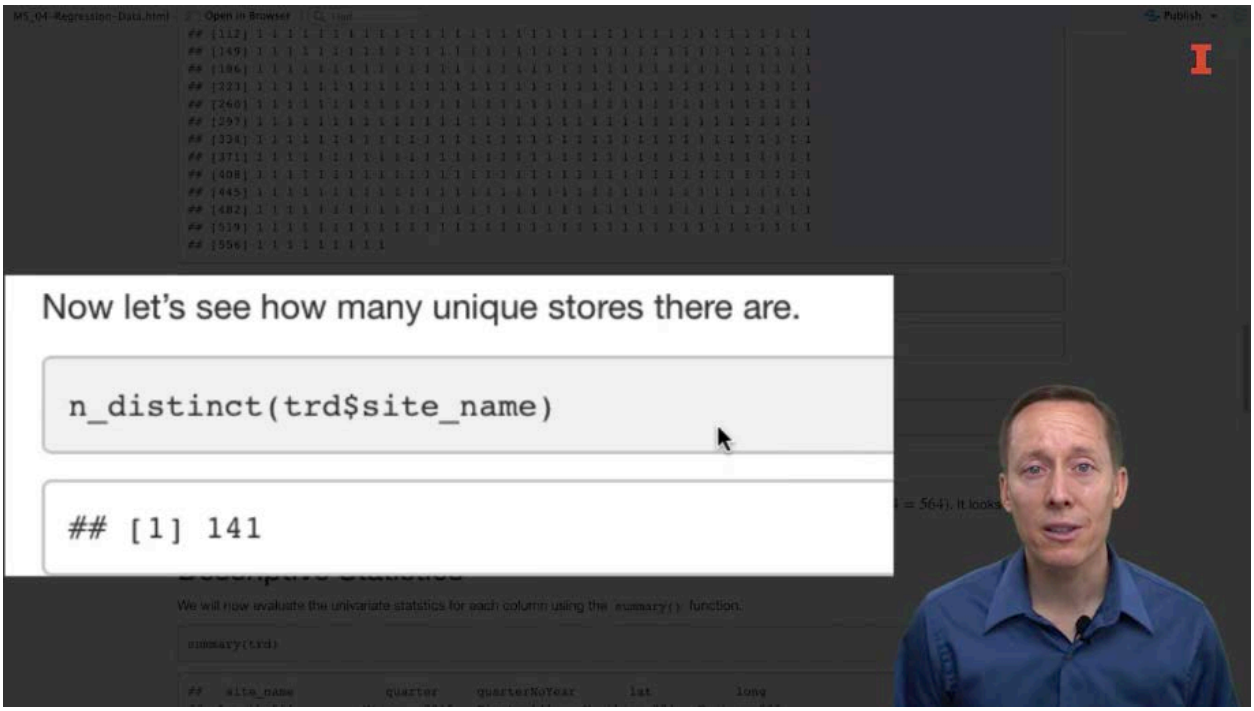
One way that we can see how many missing values there are. Is to use the is.na function on this data set. And it would just tell you a list of true and false whether there are missing values. If we sum up that actually vector of trues and falses, we can see how many are missing and in this case is 0. So obviously this data has already been cleansed for us. There are no missing values, which is great.

6. **totalRevenue** = the total amount of revenue for that store
This is the main variable that we would like to predict and exp
7. **Pop_py1** = the percentage of totalRevenue from the same
8. **Fuel_py1** = the percentage of totalRevenue from the same
9. **Juicetonics_py1** = the percentage of totalRevenue from th
10. **ColdDispensedBeverage_py1** = the percentage of totall ColdDispensedBeverage parent name
11. **OffInvoiceCigs_py1** = the percentage of tot[illegible]nue fr name
12. **Lottery_py1** = the percentage of totalRe[illegible]e s
13. **Other_py1** = the percentage of totalRev[illegible]a

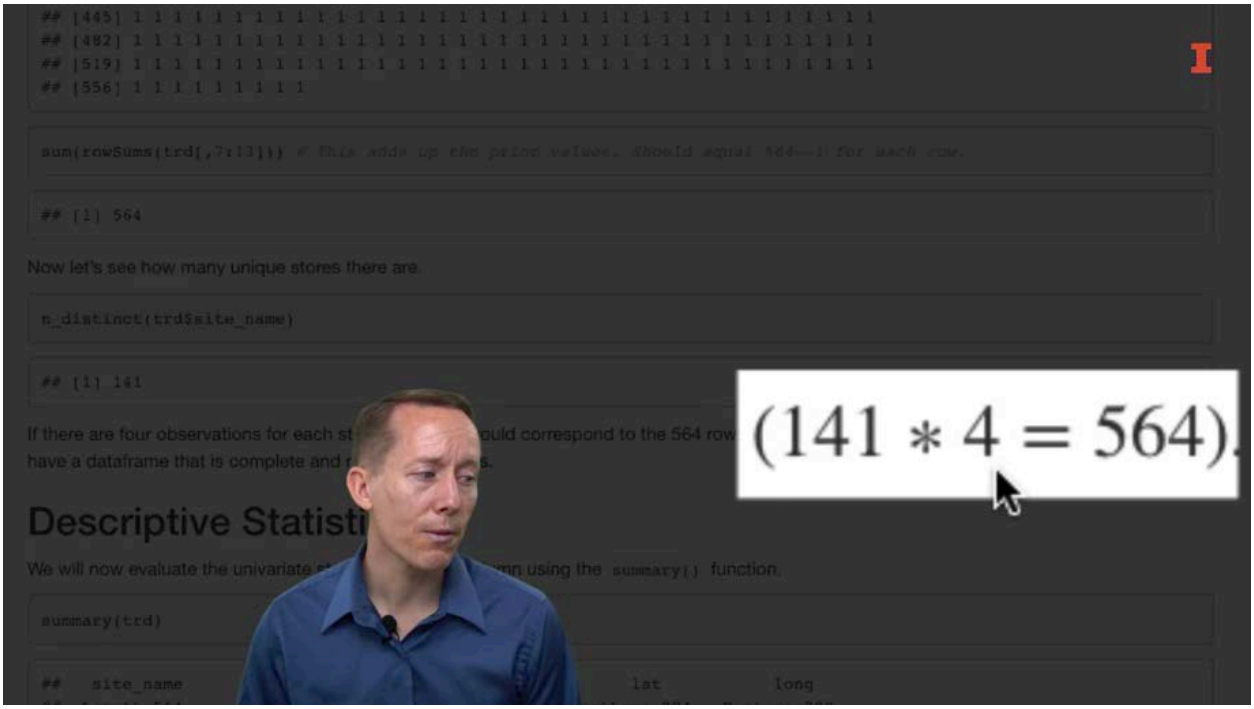
Now remember that I said each of these last six or seven columns of data are the percentage of cells from the prior year. So if we sum them up for each row, it should give us a value of one. Let's make sure that's the case to make sure this data is complete.

Now, if we use this row sums function on the last columns of data called seven through 13, it will report back the total amount of those numbers and they all add up to one. It's pretty easy to see that we could actually sum up this by wrapping it in a some function and see that this adds up to 564.

So this is helpful just to verify that we've accounted for all of the sales during the same quarter of the prior year.

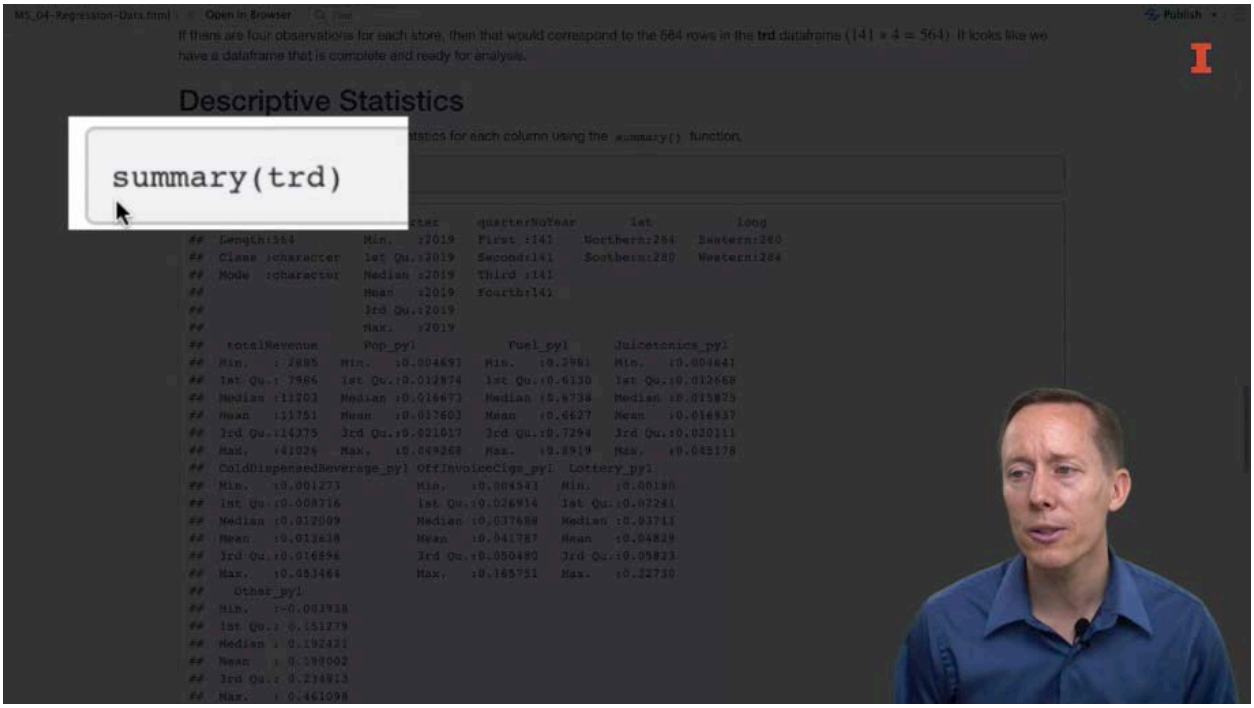


All right now, let's see how many unique stores are. So we can use the n distinct function on the site, name, column of the trd data frame and we get 141. So there's not 150, there's 141 stores.

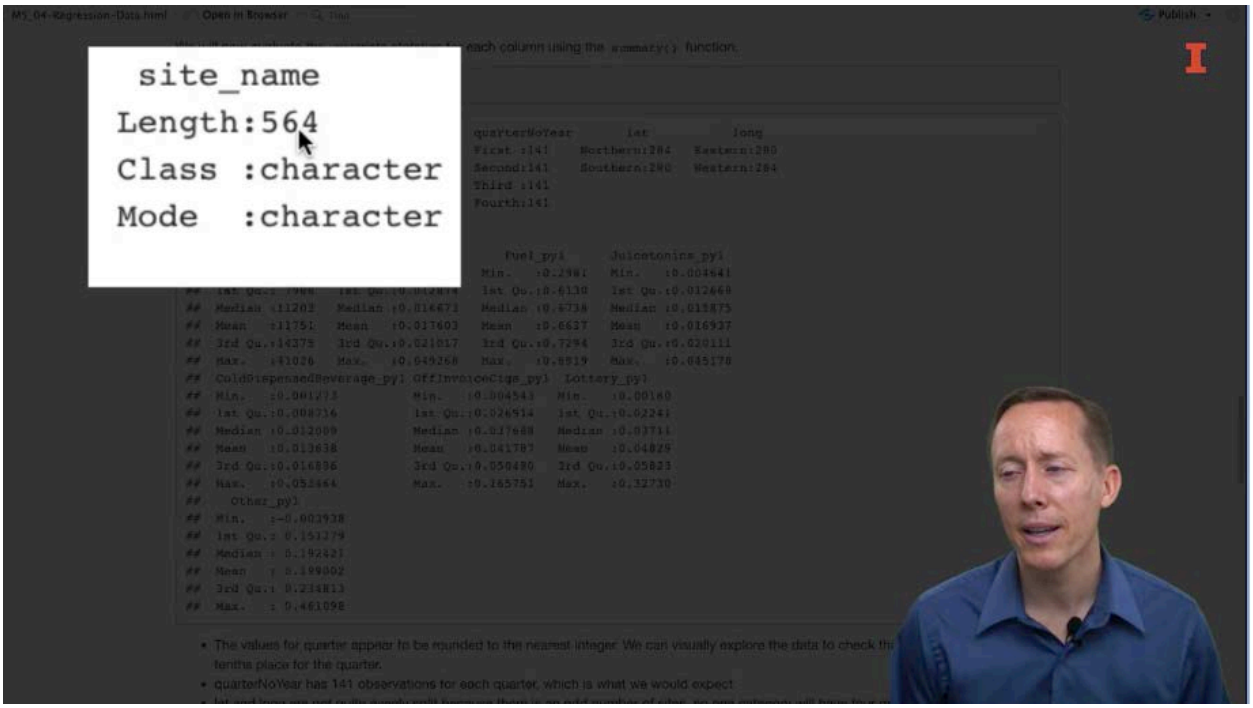


And if we want to make sure that we've got an observation for every quarter of two 2019 we could take that 141* 4 = 564. So, since there's also 564 rows in the data set, it looks

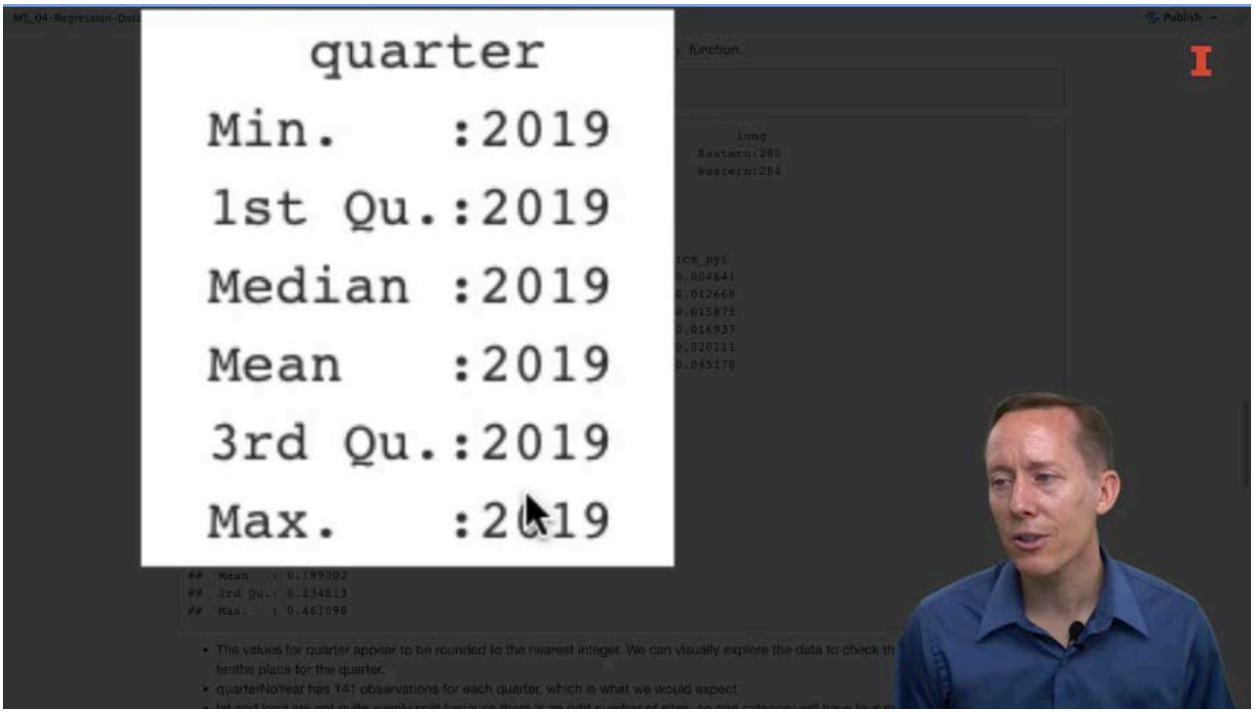
like we've got the complete data set here. We've got four observations for each of those 141 stores.



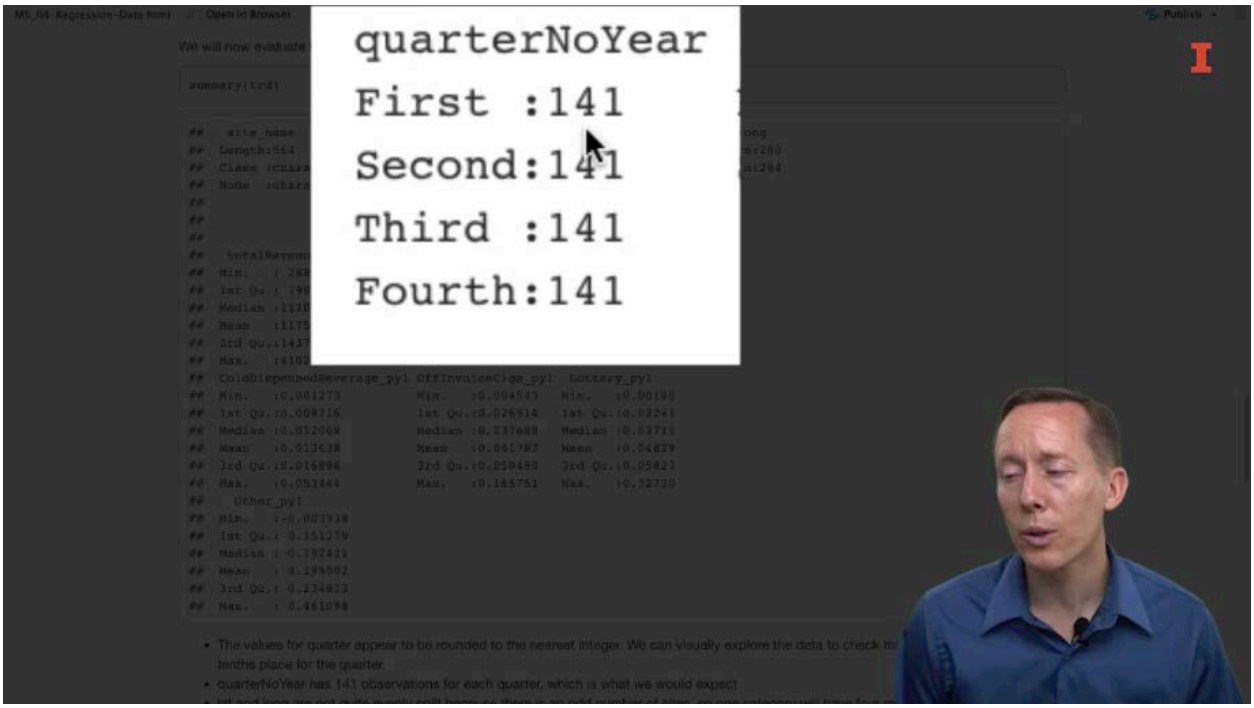
Now that we've done that, let's go ahead and understand the columns of data in this data set so we can do that using the summary function on the trd data frame.

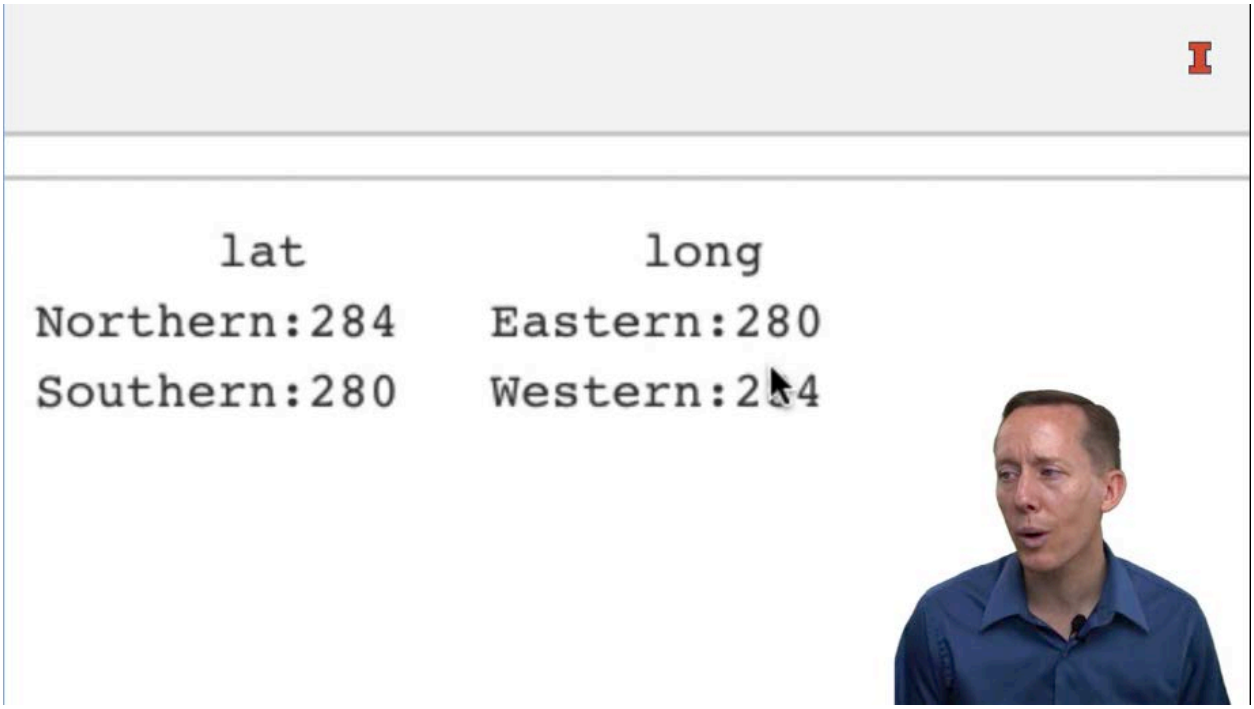


And real quickly, let's go over what this is telling us. We've got the site named column, which is a character column. And so it doesn't give us much here. It's other than there are 564 observations,

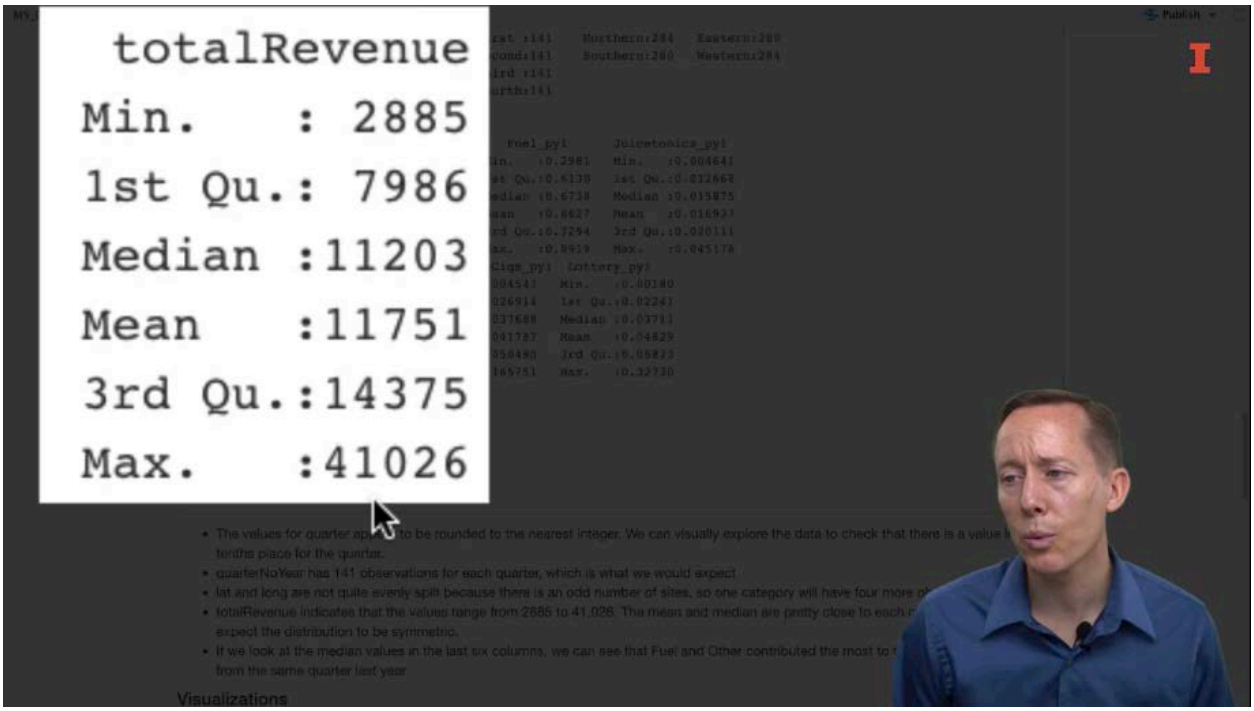


we can see that all of the observations are for 2019 quarter, no year.

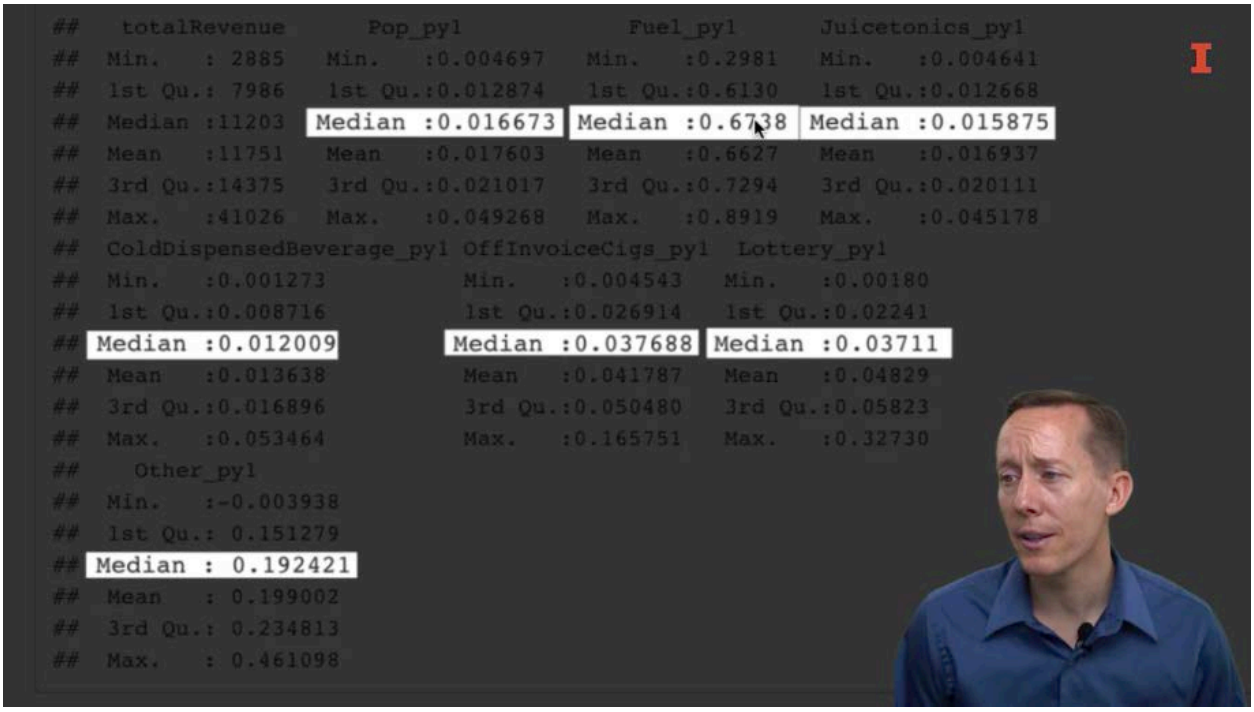




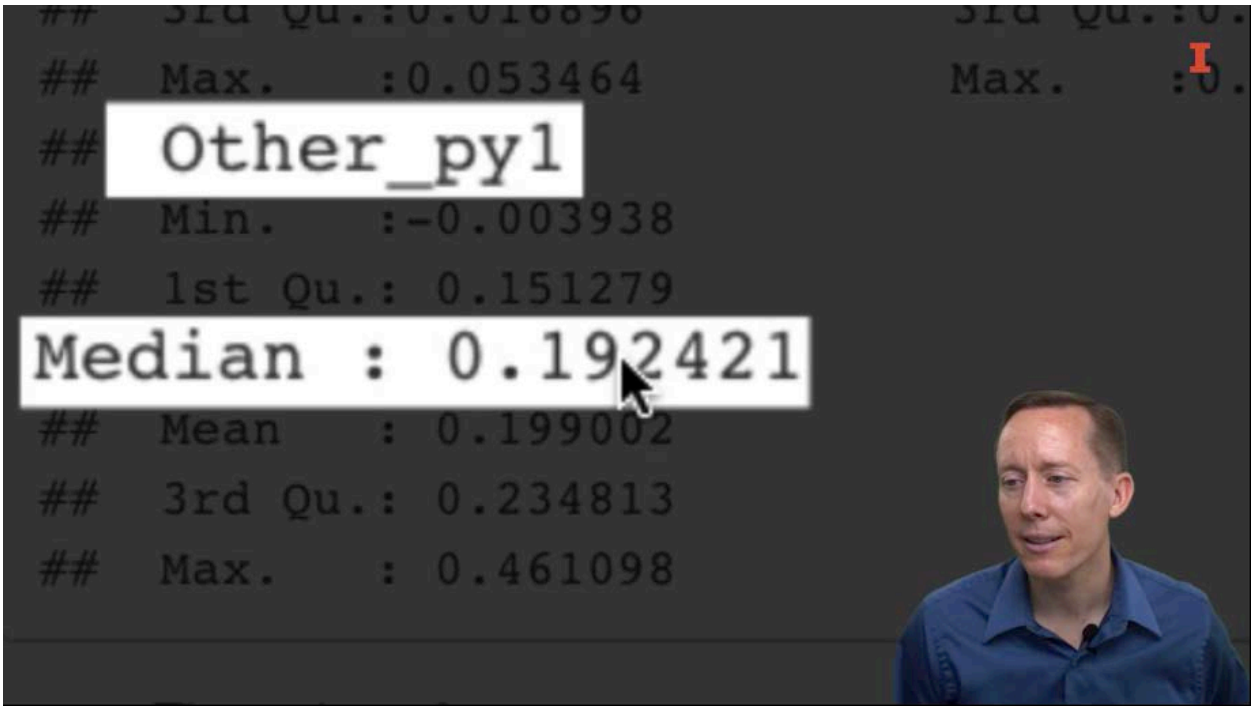
It tells us that there are a 141 observations for each of those four quarters. So that's important. It adds up to 564, so that's what we would expect lat and long these numbers are not quite equal because we've got an odd number of stores 141 stores.



You can see the descriptive statistics for the range of revenue, the minimum is 2885, the max is 41026.



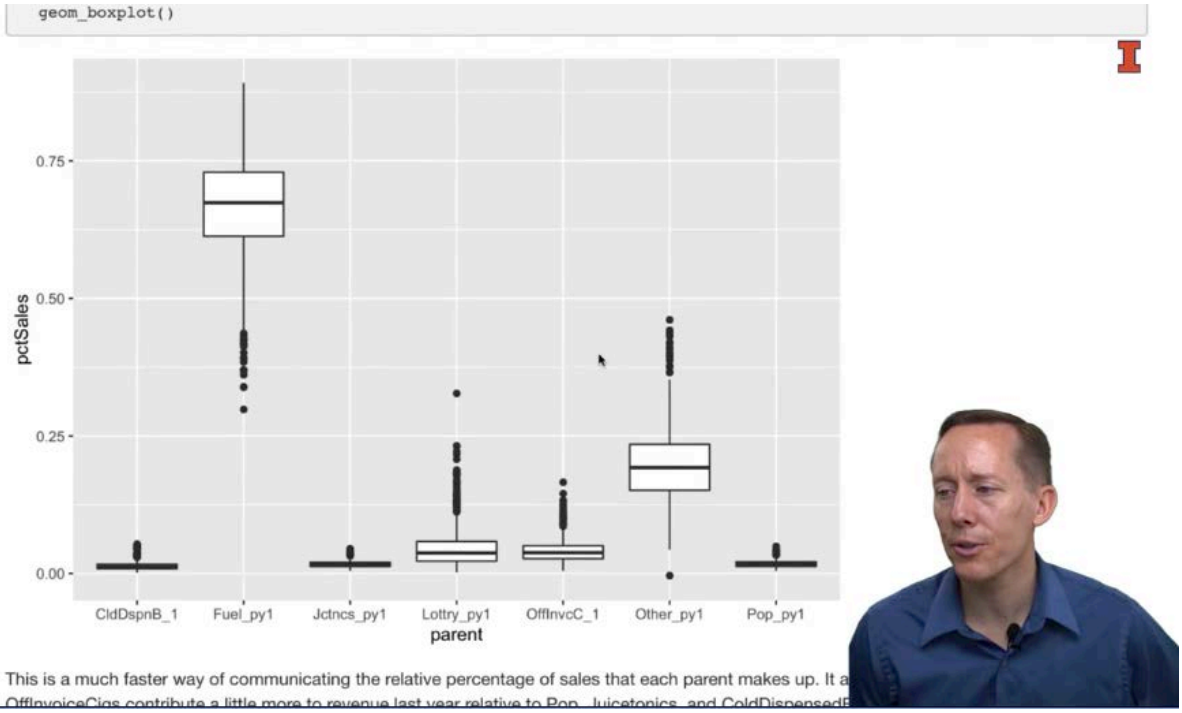
And then we can see the percentage of sales for each of these seven categories of product for the same quarter of the prior year. So if you look at the mediums, you can real quickly see that fuel makes up the majority of the sales.



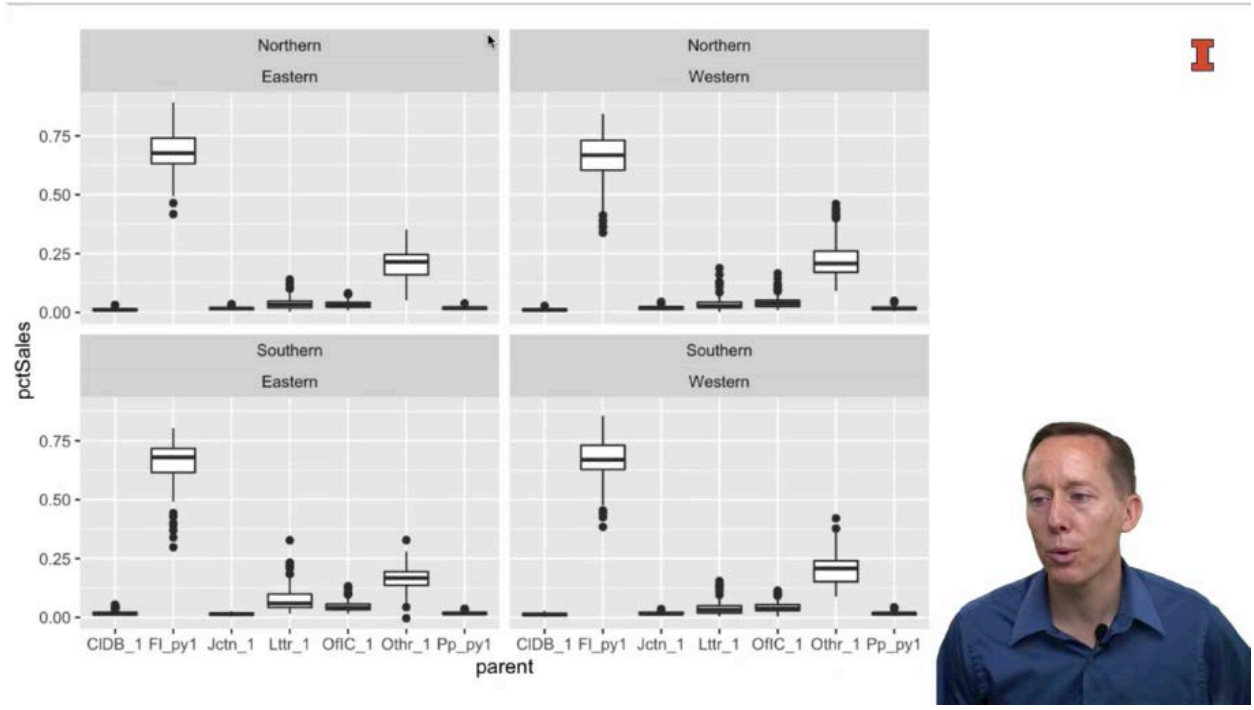
All these others are pretty low except for other, which is 19% of the total sales. Now let's go ahead and explore that information visually. Oftentimes these visual plots make it much easier to see what's going on.

So we're going to take the trd data frame, pivot it to a longer data frames. So pivot longer and we're going to relabel the parent. We're going to abbreviate it so it's only 10 characters long. That's a really useful function. And then we'll put it into the ggplot

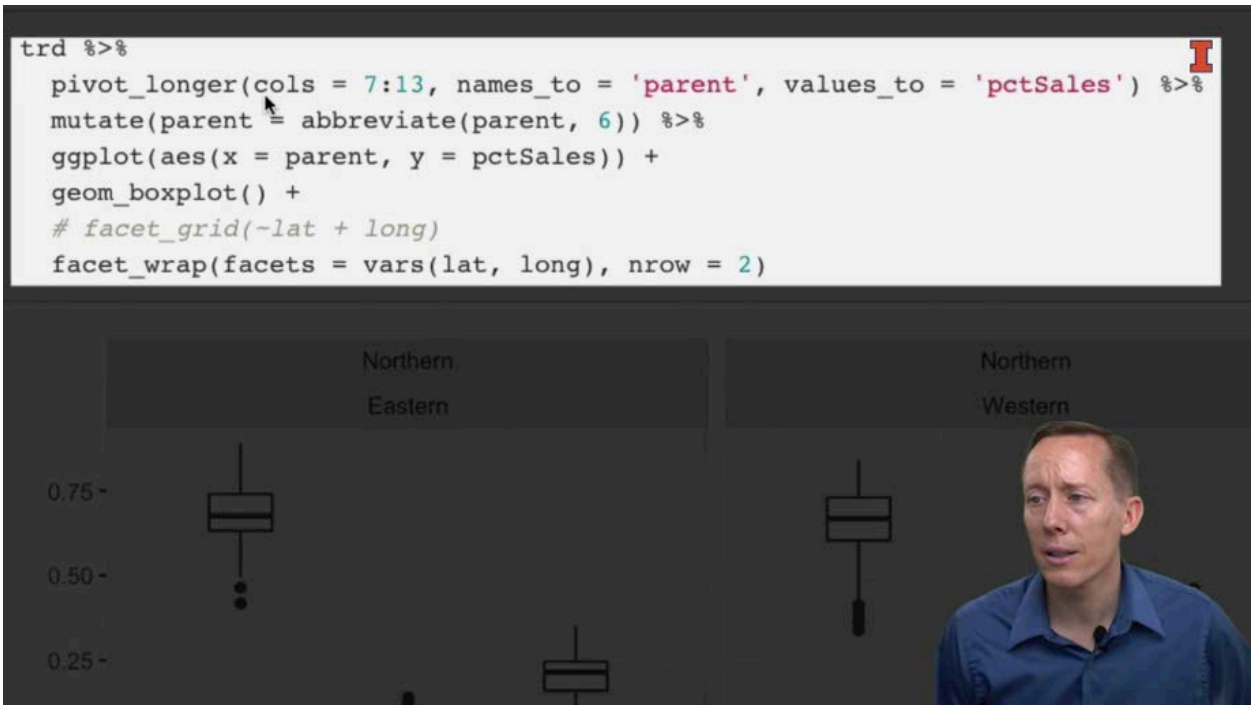
function for plotting, we'll put parent on the X. Axis percent, sales on the Y axis and then create box spots and it results in this visual right here.



This is really helpful for quickly recognizing that fuel makes up most of the percentage of sales. The other category is the second highest. And then these other remaining categories are all pretty low, looks like lottery and offinvoiceCigs are a little bit higher than popjuice and tonics and cold dispense beverages. So that's helpful. Now what if we want to evaluate if this is the same for every region that we're looking at since region is one of our main variables of interest, I want to see how this affects our total sales.

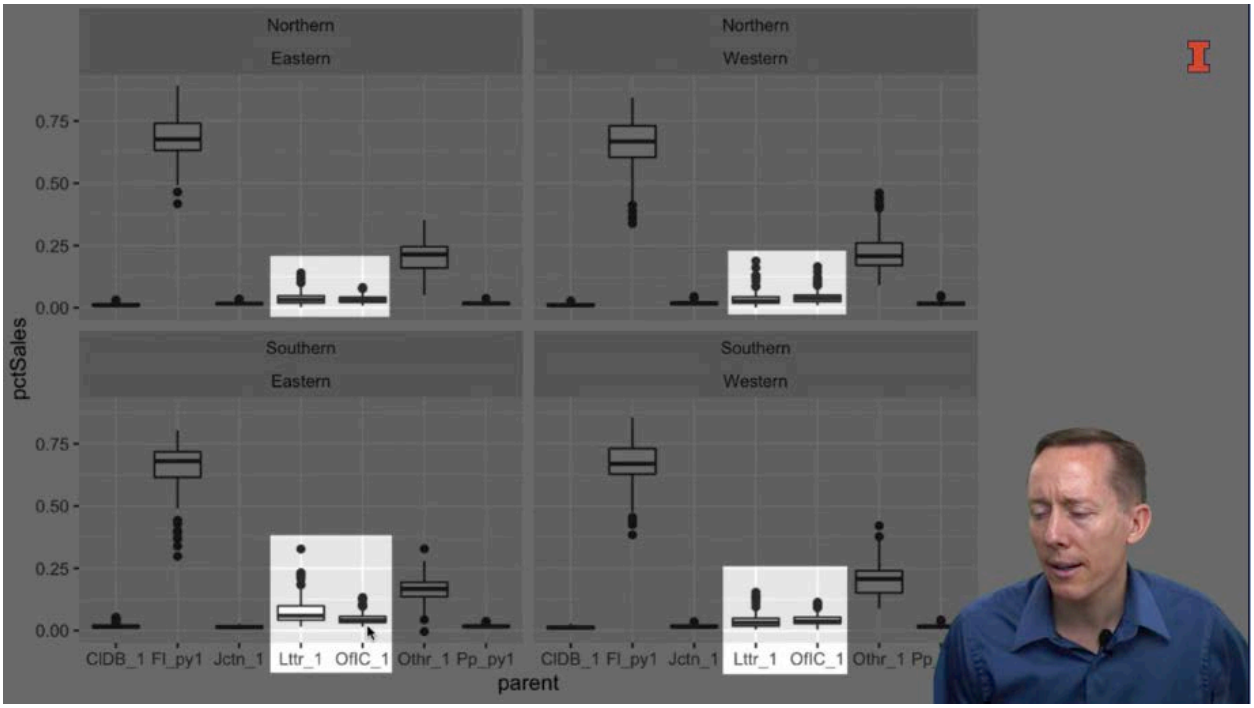


So we can look at these distributions based on the four different regions we have.



So we'll take that trd data frame pivot longer. We will abbreviate the parent name. So it's only six characters this time. And then we'll use the ggplot function put X, the parent on the XX. Axis, percent sales on the Y axis use a box plot. And then there's a couple of functions that are helpful for creating tiny multiples. And we're going to use facet wrap.

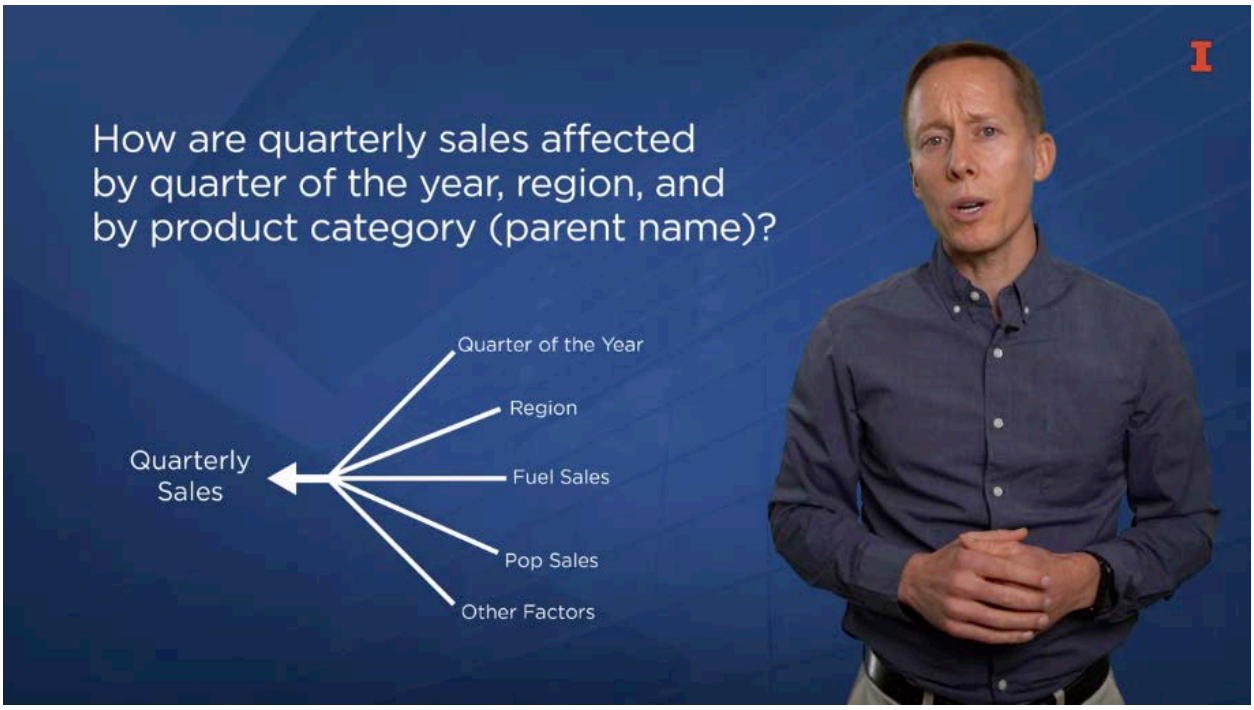
You could also use facet grid and we're going to basically say, hey, we want to divide up these box plots based on the lat and long categories and we want to rows of data.



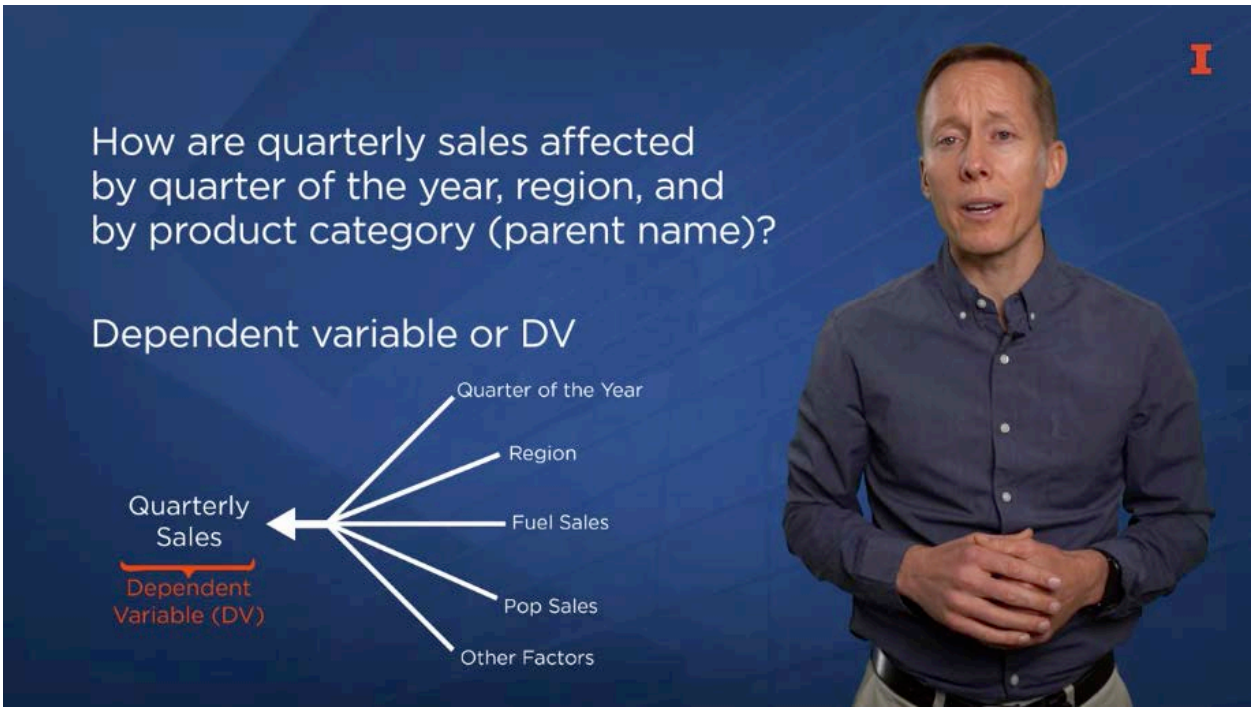
And so that presents this nice set of four box plots here. And this allows us to look at the four quadrants and we can see that for the Northern, it looks real quickly. These these boxes look pretty similar. The Southern, they look pretty similar to each other. The one difference might be comparing Southern to Northern, where the lottery and offinvoicecigs appeared to be a little bit higher. The distributions are a little higher relative to the Northern regions, otherwise looks pretty similar, especially this Southeastern region is higher. All right, so there's a quick overview of the data that will be looking at. Hopefully this process is a good reminder of how you can explore the data on your own and how visualizations are very helpful for exploring the data.

Lesson 1-3.3 What Problems Can Regression Answer?

Regression is a powerful statistical technique. It is the workhorse of machine learning. It is essentially an algorithm that is used to take a bunch of data and create a model that we can use for explanation, inference, and prediction. In this lesson, we want to introduce you to regression by describing the types of questions that it can answer. We will also introduce you to some common terminology associated with using regression.



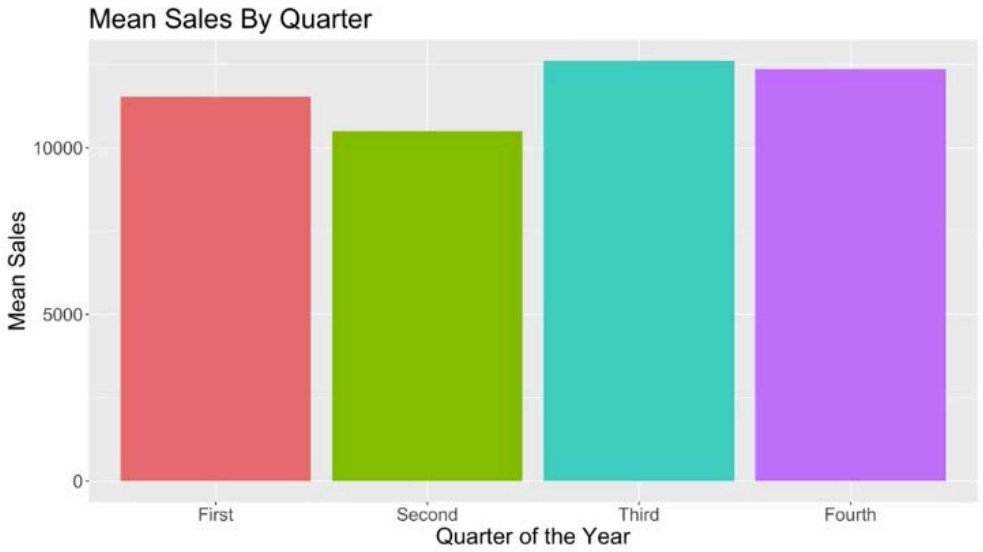
Let's start by discussing how regression can be used to explain how two or more variables are related. Let's assume that our business problem is how our quarterly sales affected by quarter of the year, region, and buy a product category or parent name? This type of question can be difficult to answer because quarterly sales is likely to be affected simultaneously by quarter of the year, region, product category, as well as other factors that aren't even a part of our question. In data analytic parlance, the variable that we're trying to explain quarterly sales in our business problem is known as the dependent variable or DV for short.



This is because it's value is thought to depend on the value of the other variables in our question, quarter of the year, region, and product category.

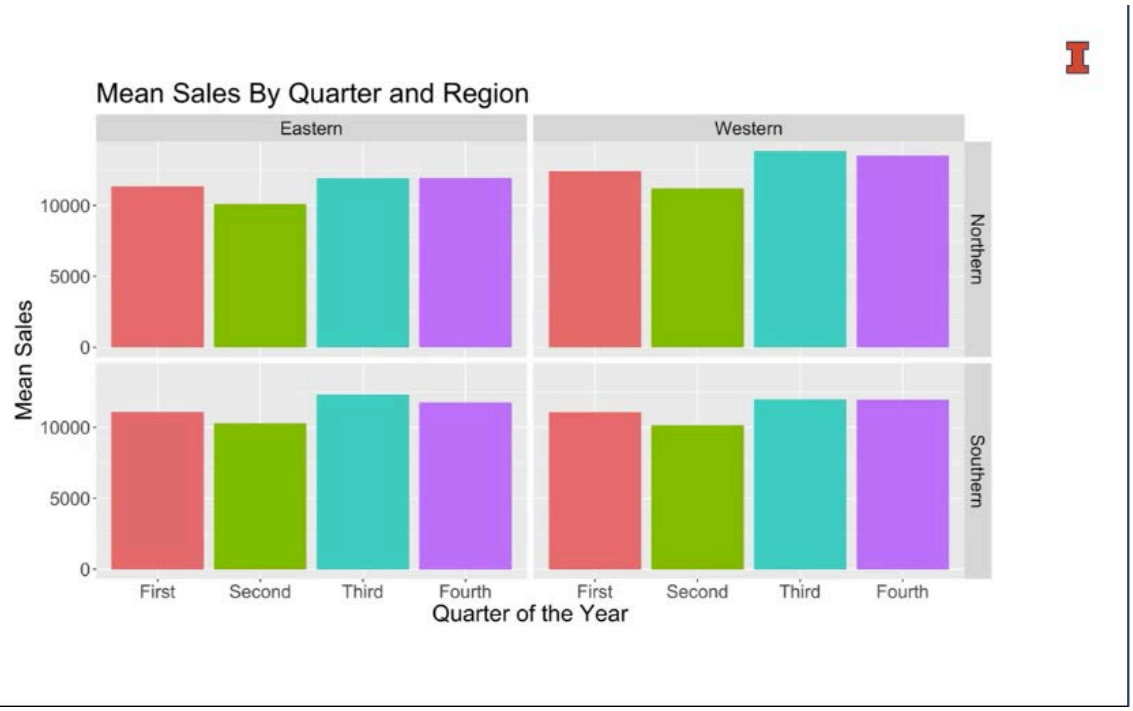
These other variables that are thought to affect the dependent variable are known as independent variables or IVs. That is because their values are independent of our model. These other variables are also known as explanatory variables because they are expected to explain the level of the dependent variable.

Regression allows us to separate and quantify the direction and magnitude of the individual effects for each of these independent variables. It does this by evaluating various combinations of these variables and quarterly sales and then comparing the averages.



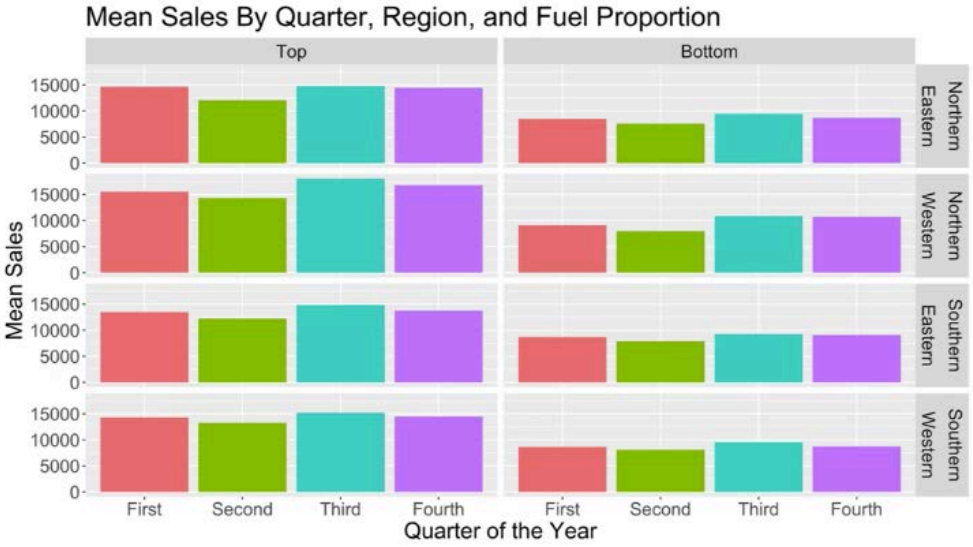
For instance, if you were to compute the average quarterly sales and compare them to each other, then you would be able to see that the quarterly sales are on average

lowest for the second quarter of the year and highest for the third quarter of the year. A simple bar chart can help you quickly identify this pattern, and a table can help you quantify the difference.

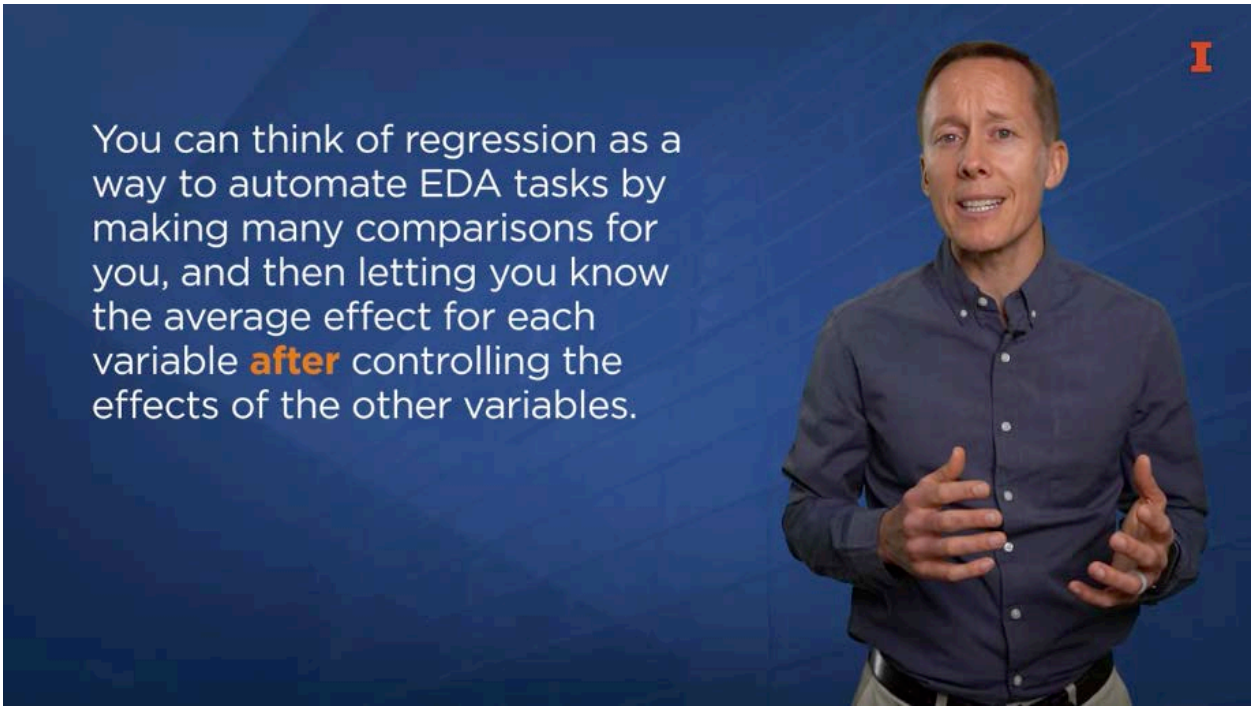


You might wonder if that pattern is consistent across various geographic regions. You could also compare quarterly sales across the regions and quarters of the year. This would result in 16 different averages that you'd have to compare. It looks like that general trend would be consistent, but perhaps not as strong in the southeastern region as it was in the northwestern region.

Quantifying the effect of region and quarter is doable, but would take a bit more effort.

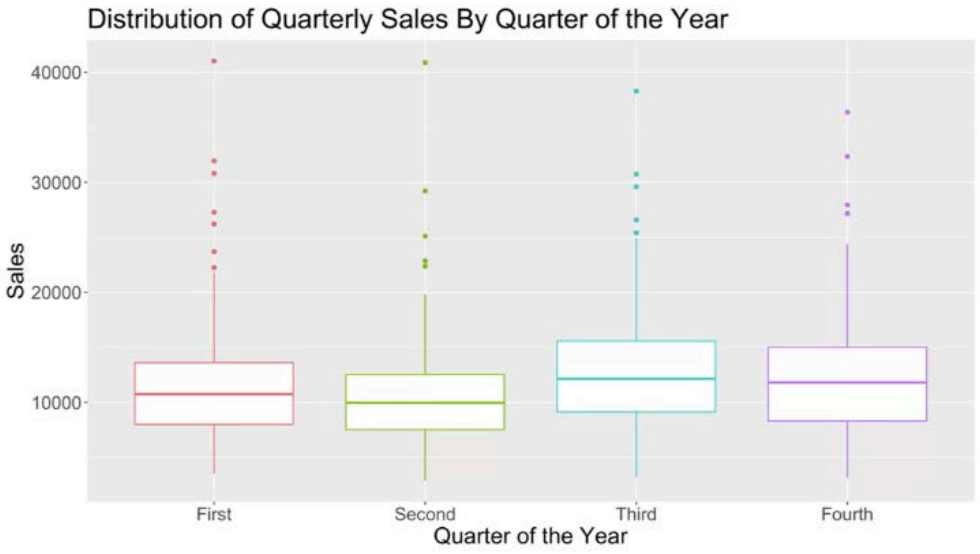


Now what if we compared the average quarterly sales for those stores that were in the top half of values for fuel PY1 versus the bottom half. This would result in 32 different averages to keep track of and quantify. You can see how this would become increasingly difficult as the number of dimensions increases.

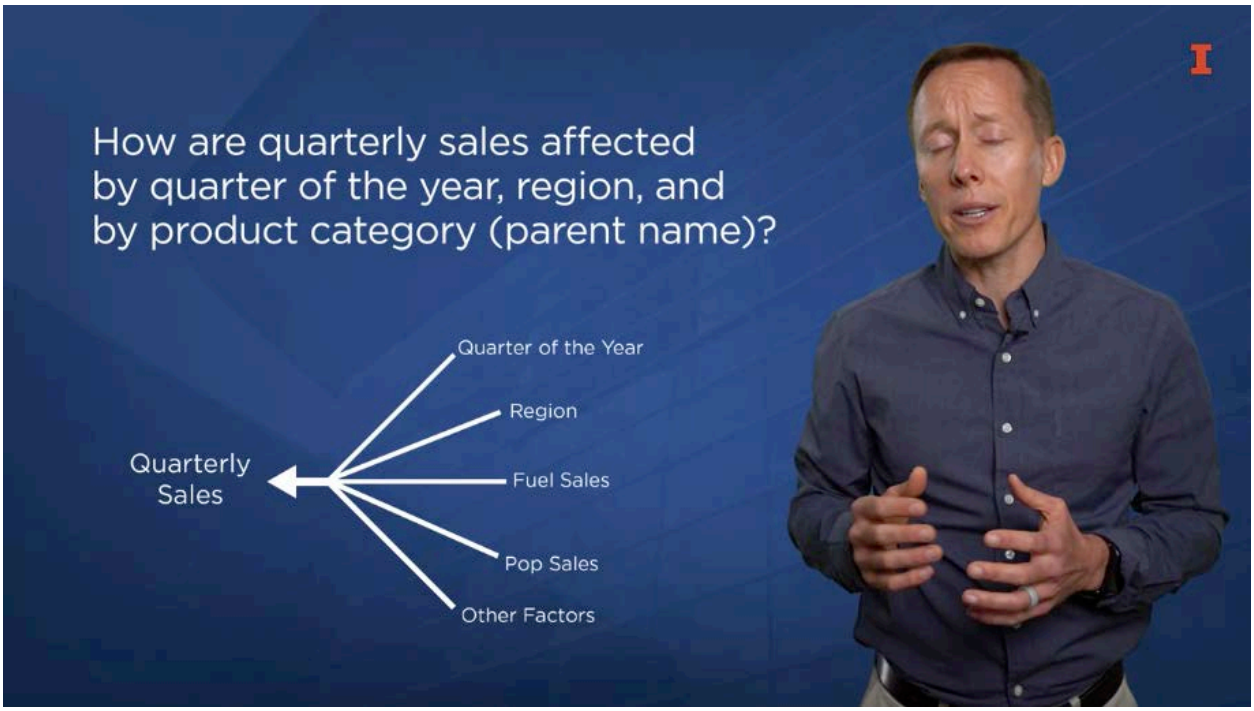


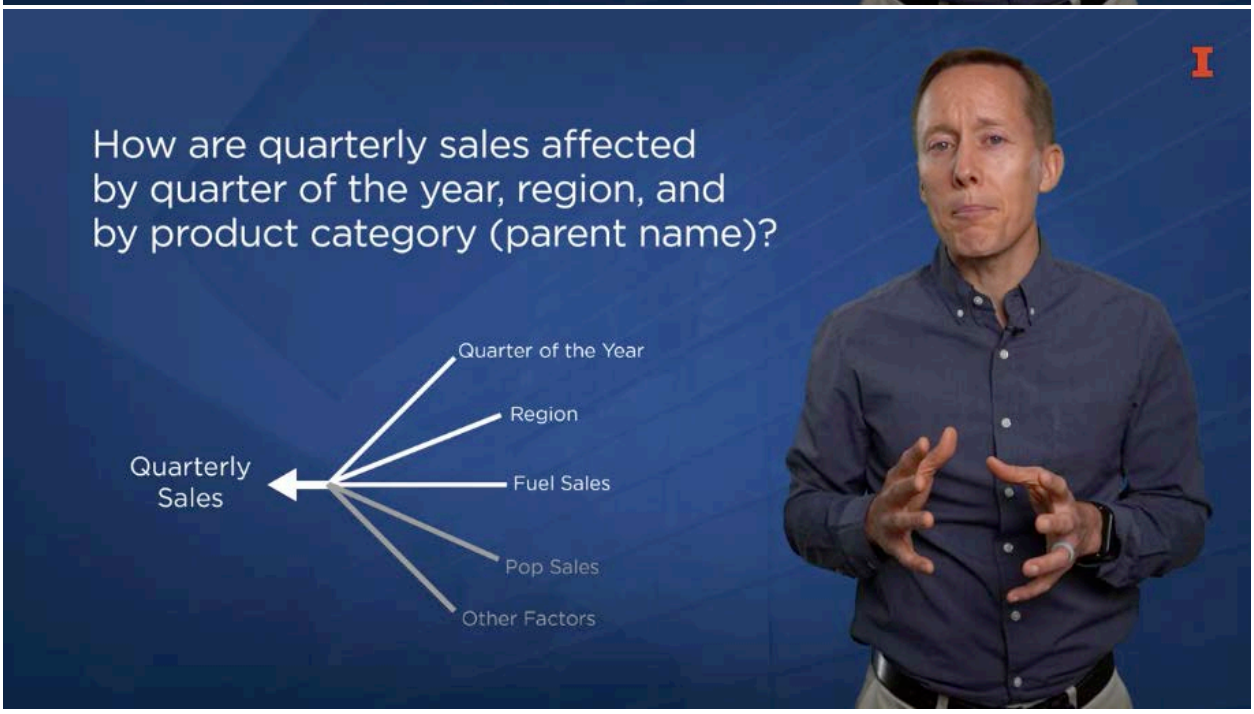
In one sense, you can think of regression as a way to automate EDA tasks by making many comparisons for you, and then letting you know the average effect for each

variable after controlling for the effects of the other variables. Isn't that great to think that regression is an exploratory data analysis robot. Now let's talk about how regression can help with inference. Inference is all about using evidence to arrive at a conclusion. Inferences are rarely clear cut. There's typically some amount of uncertainty. It's important to communicate the amount of confidence that we have in a conclusion. Let's refer back to our question about whether sales fluctuate by quarter of the year. We already saw the averages are different from each other, but how confident are we that the second quarter sales will always be less than the third quarters?



To help answer this question, we can look at boxplots which indicate that at least sometimes sales during the second quarter are higher than during the third. How would you express your confidence that sales during the second quarter are lower than sales during the fourth quarter. You might be inclined to use phrases like most of the time, or I'm really confident that sales during the second quarter will be less than sales during the fourth quarter. The problem is that many people may interpret those terms differently. Based on properties associated with t-distributions, regression allows us to quantify in statistical terms the extent to which we can be confident that observe differences in means are different from each other. Thus, we can avoid ambiguous terms like pretty confident, and instead use more precise terms like 95 percent confident.





After running several regressions to identify the relationship between sales and quarter of the year, region, and product category, we may end up eliminating independent variables from the model if we're not confident that they have a reliable effect on quarterly sales. The idea is to create a parsimonious model, meaning a model that has no more complex than it needs to be.

This brings us to our last type of question that regression can be used to answer, which is prediction questions. Regression models can be used to make predictions if we can get a reliable estimate of the levels of the independent variables before knowing the dependent variable. For instance, if we want to predict quarterly sales one year in advance, and our regression model is based on the percentage of sales from fuel sold during the same quarter of the year, then the model will not be useful for prediction purposes because we will not know the percentage of sales from fuel until we find out our total quarterly sales. This is why we are using percentage of sales from the same quarter during the prior year. It will probably have less explanatory power, but it is more timely and will allow us to make predictions one year in advance. Ideally, most businesses would like to make reliable predictions many years into the future, but there's often a trade-off between how far into the future you can make predictions and the accuracy of those predictions. For this reason, domain knowledge is really important. If you think that you only need one quarter lead time to make predictions, then perhaps you can use independent variables based on the prior quarter rather than only on the same quarter from last year.

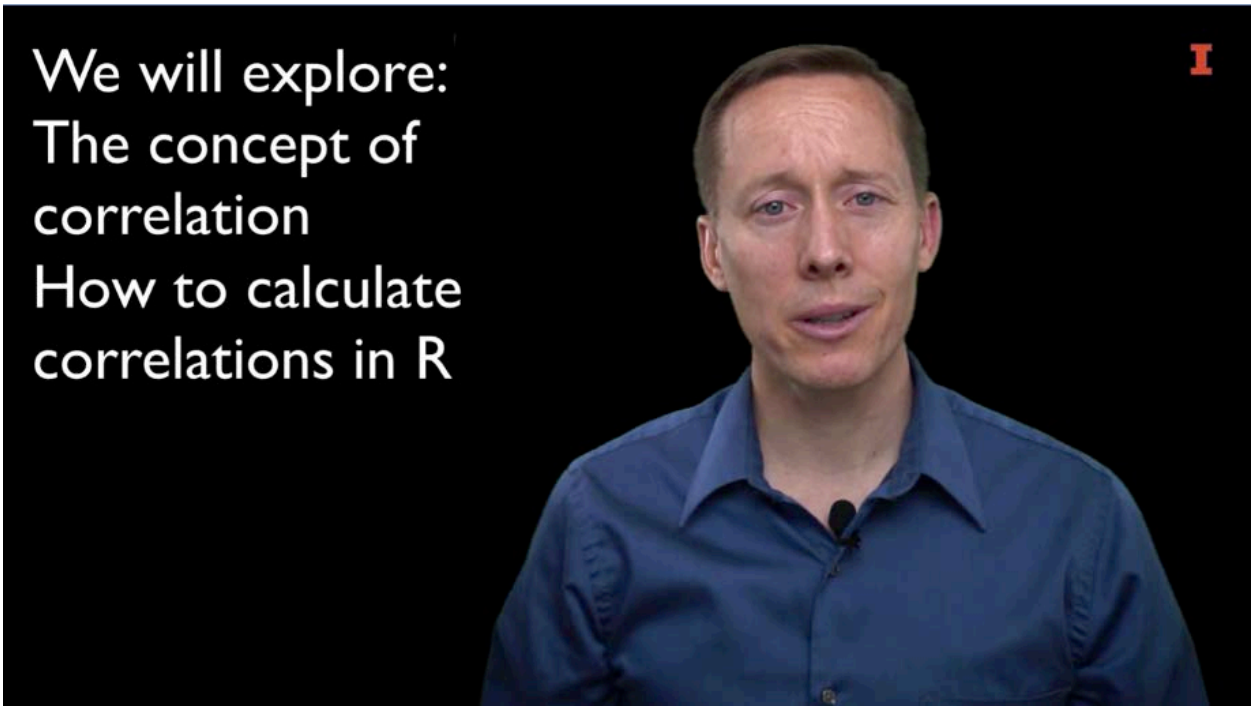
In conclusion, regression can be used to answer business questions related to explaining relationships, making inferences, and making predictions. Although, we're going to focus on how to explain, infer, and predict quarterly revenue of a convenience store,

it's important to recognize that regression can be used for many types of business decisions, such as predicting the price at which a house will sell, explaining cell phone call performance, estimating the success of a new product,

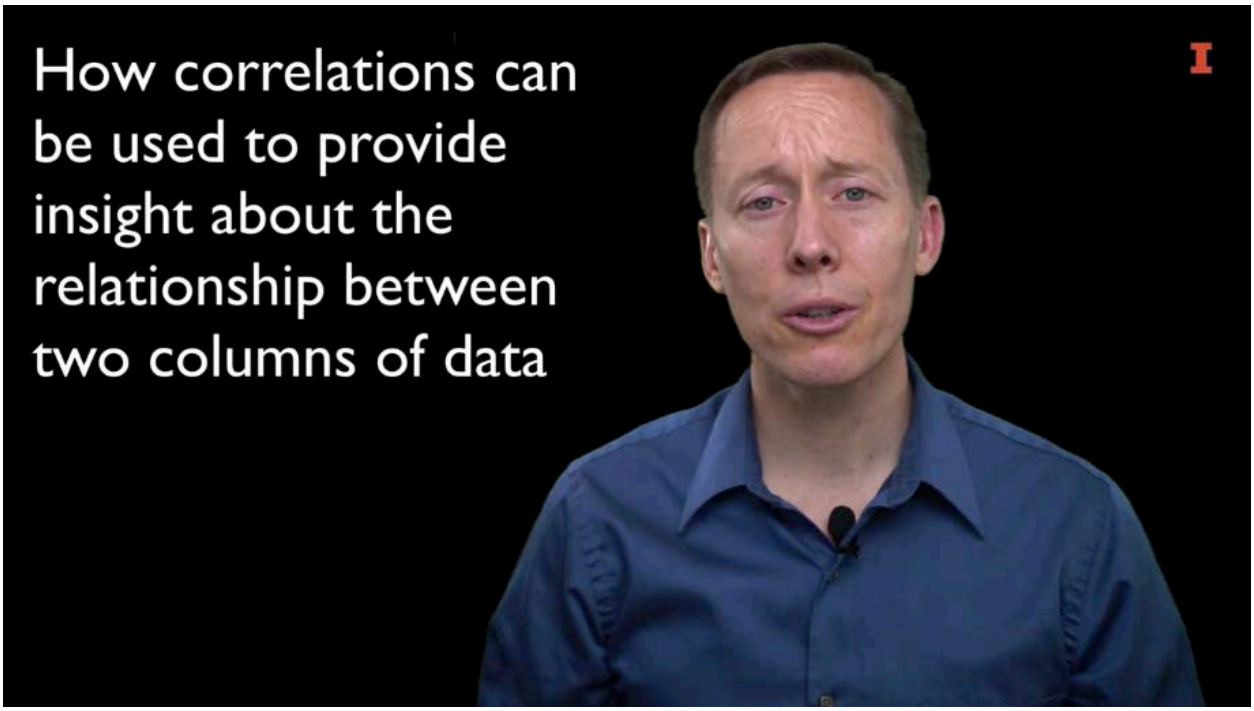
predicting whether customers will repay a loan, predicting the reliability of a supplier, and making inferences about factors that lead to the success of stores or branches.

Lesson 1-4: Regression

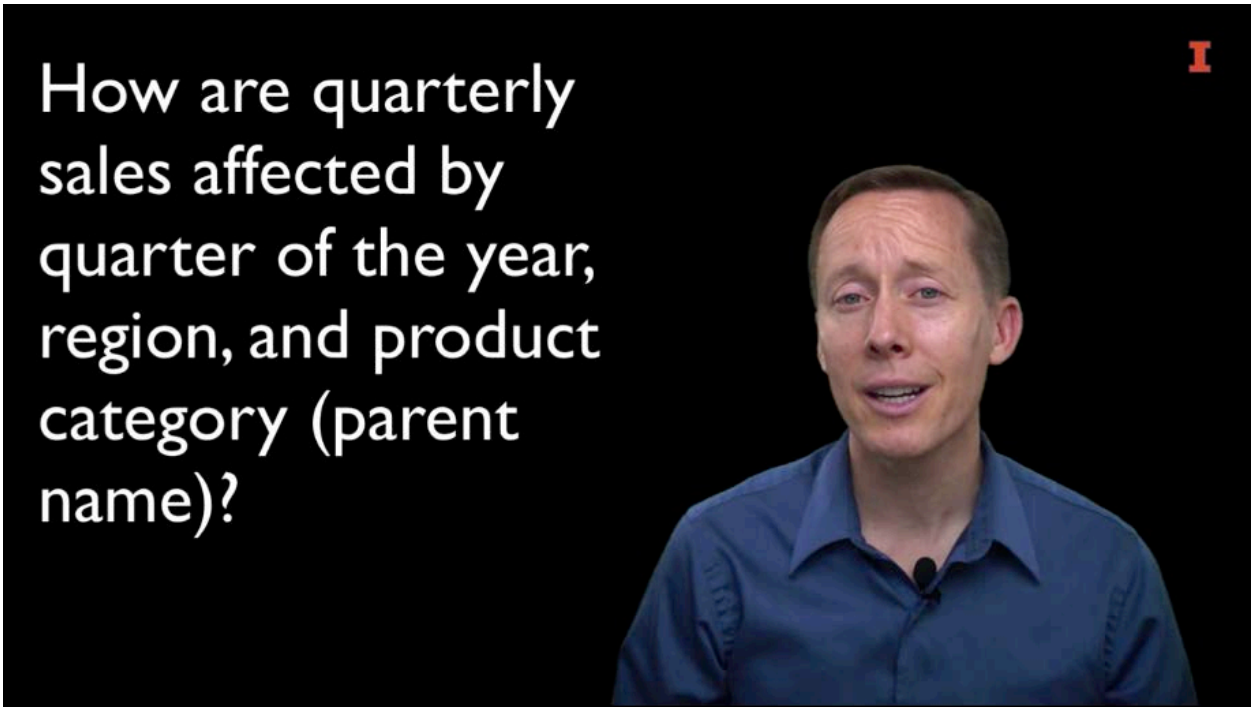
Lesson 1-4.1 Correlation



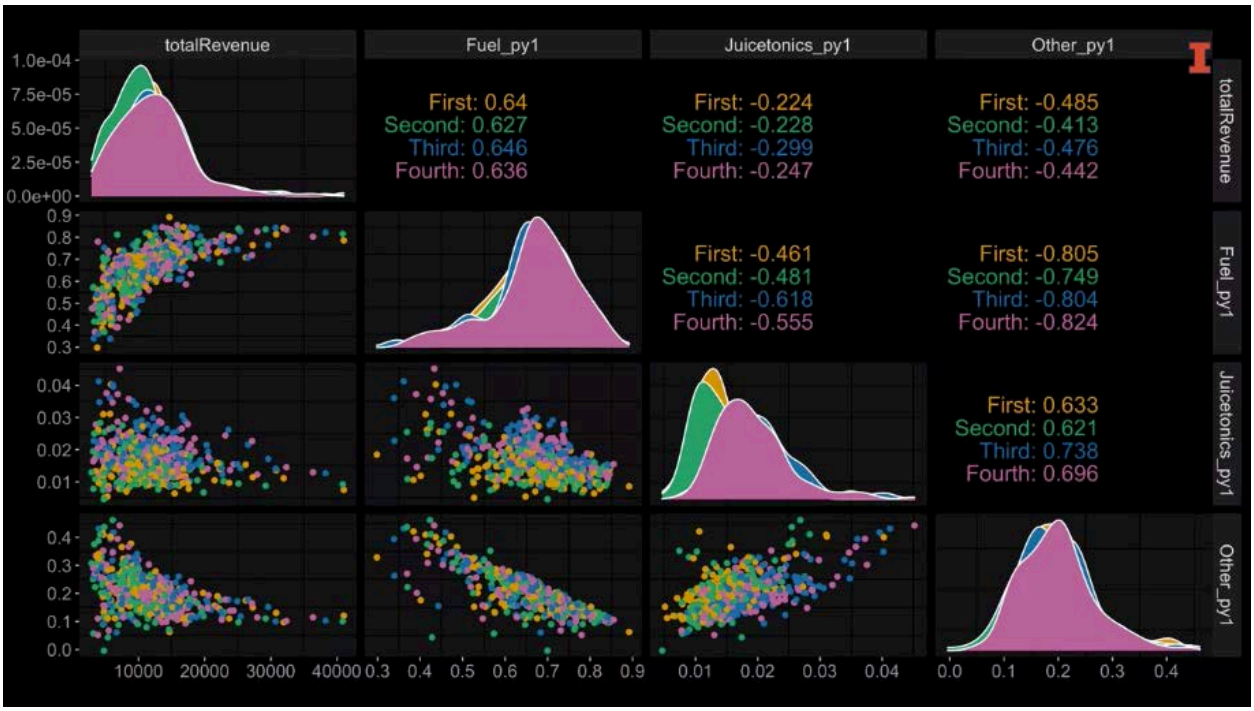
In this lesson, we will explore the concept of correlation, how to calculate correlations in R.



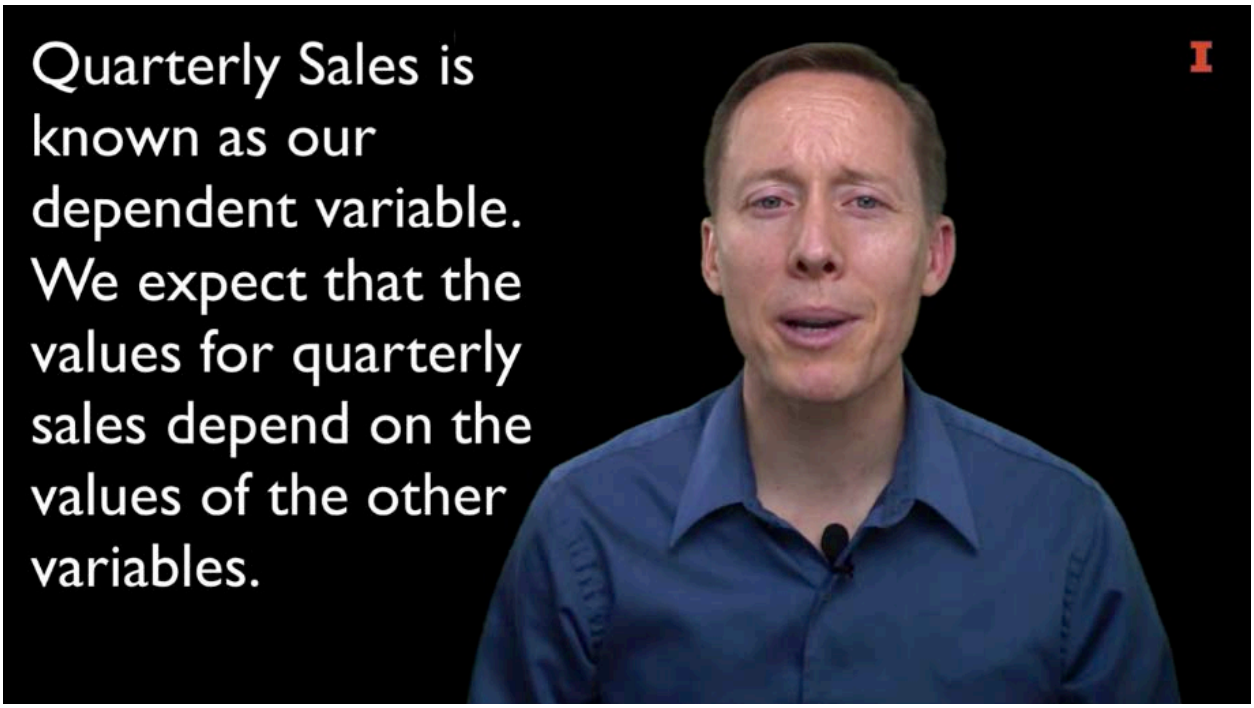
And how correlations can be used to provide insight about the relationship between two columns of data. At this point, you should have an understanding of the tech and regression data.



And we will use this data to provide insight to our business question, which is how our quarterly sales affected by a quarter of the year; region and product category.

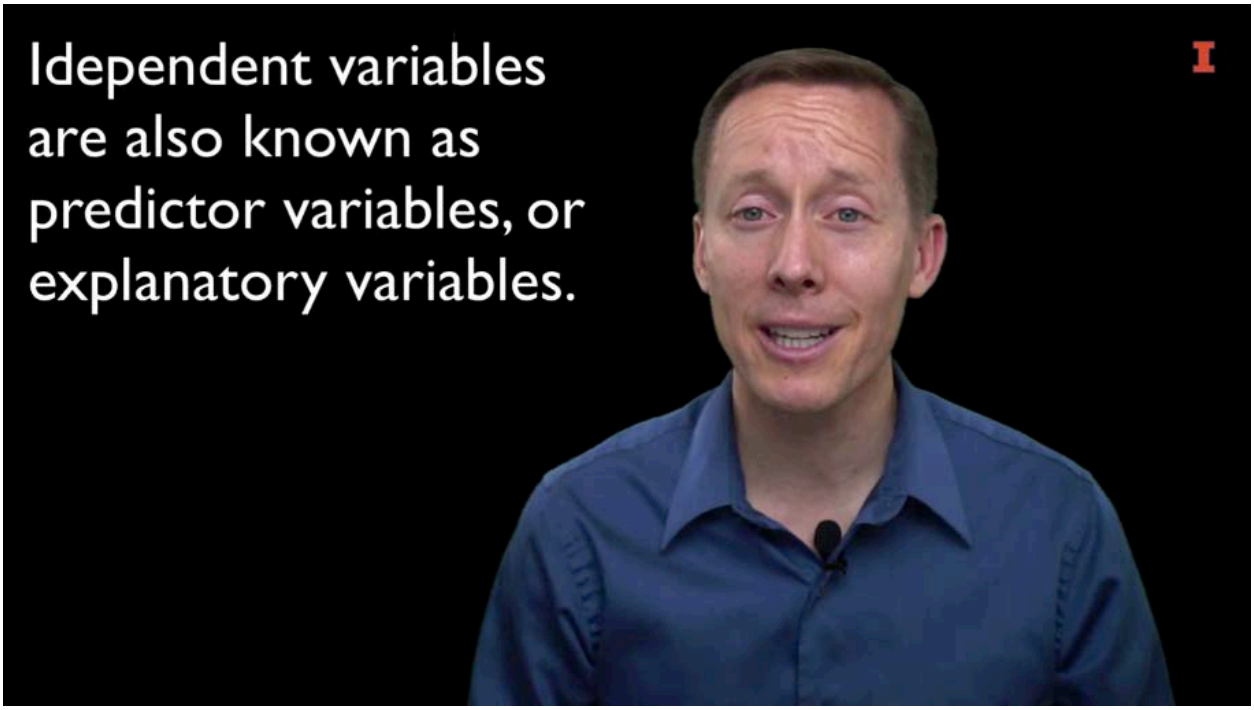


Now, correlation can be used along with scatter plots to investigate to a relationships between quarterly sales and the other variables. Scatter plots help us to see a relationship, whereas correlations help us to quantify a relationship. Because we want to better understand how quarterly sales are affected by the other variables.



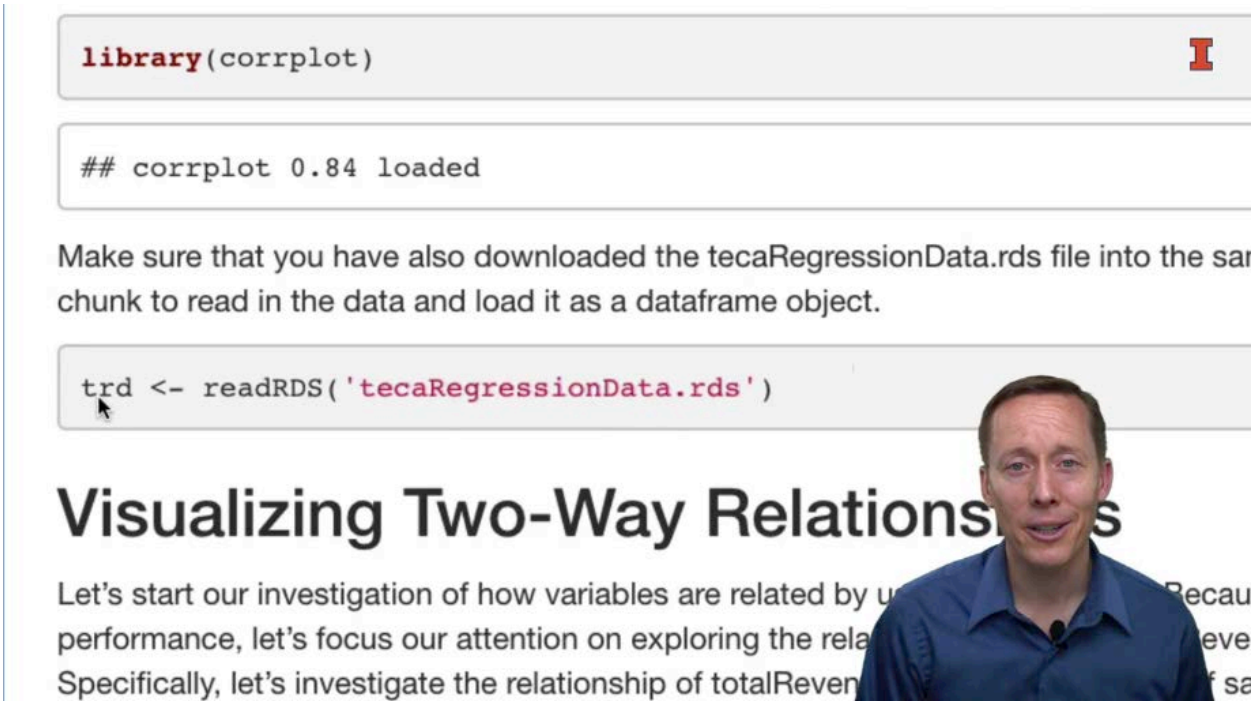
Quarterly Sales is known as our dependent variable. In other words, we expect that the values for quarterly sales depend on the values of the other variables. In contrast, the

other variables are known as independent variables. Because we expect that their values are determined independently of the other variables.



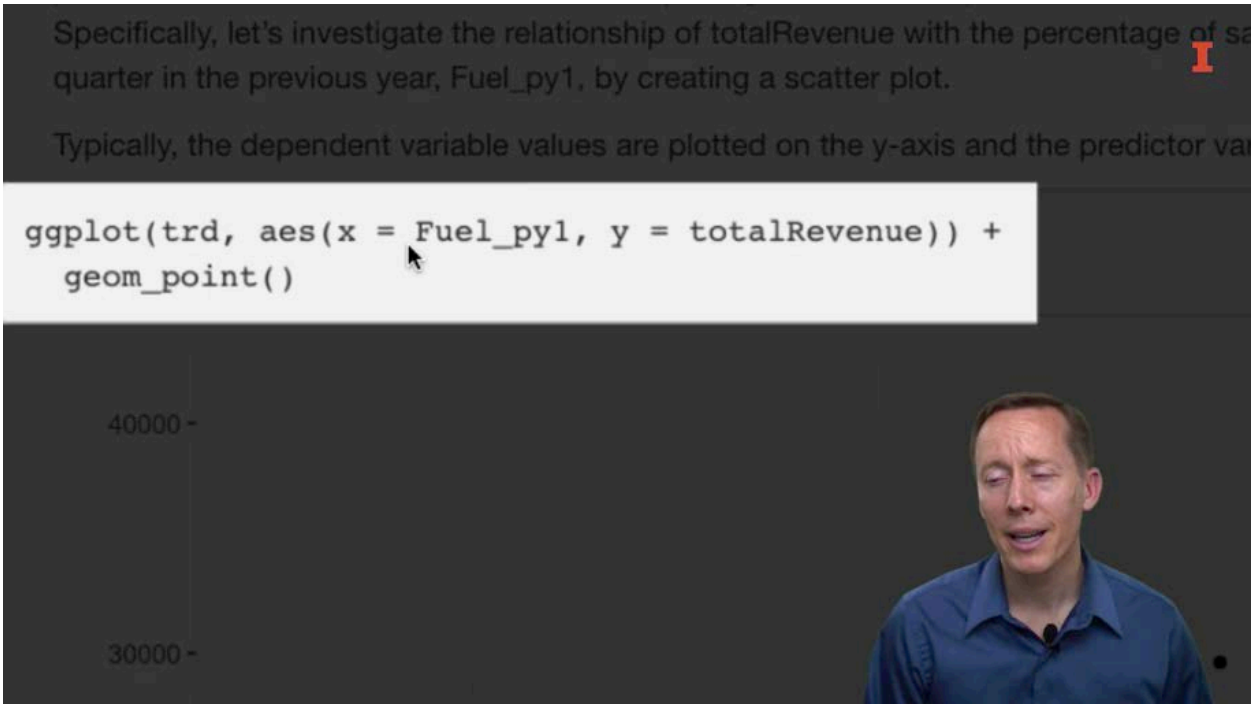
These independent variables are also known as predictor variables, or explanatory variables. Because they are being used to explain and predict the value of quarterly sales.

If you'd like to follow along with this exploration, you can do so by downloading the associated R markdown file as well as the tecaRegressionData. Now you'll also need to make sure that you install and load the tidyverse collection of packages and the core plot package.

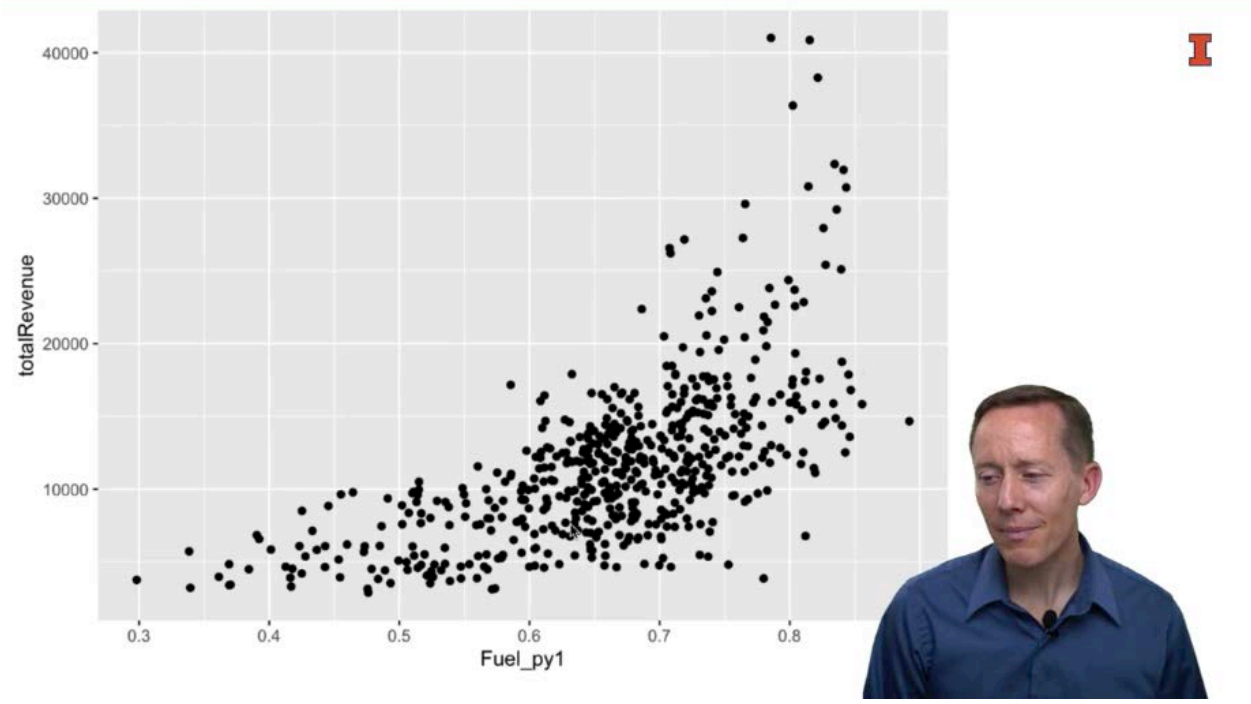


You also need to make sure that you read in the tecaRegression.rds data file as a data frame. Now once you've done that, let's go ahead and start visualizing relationships

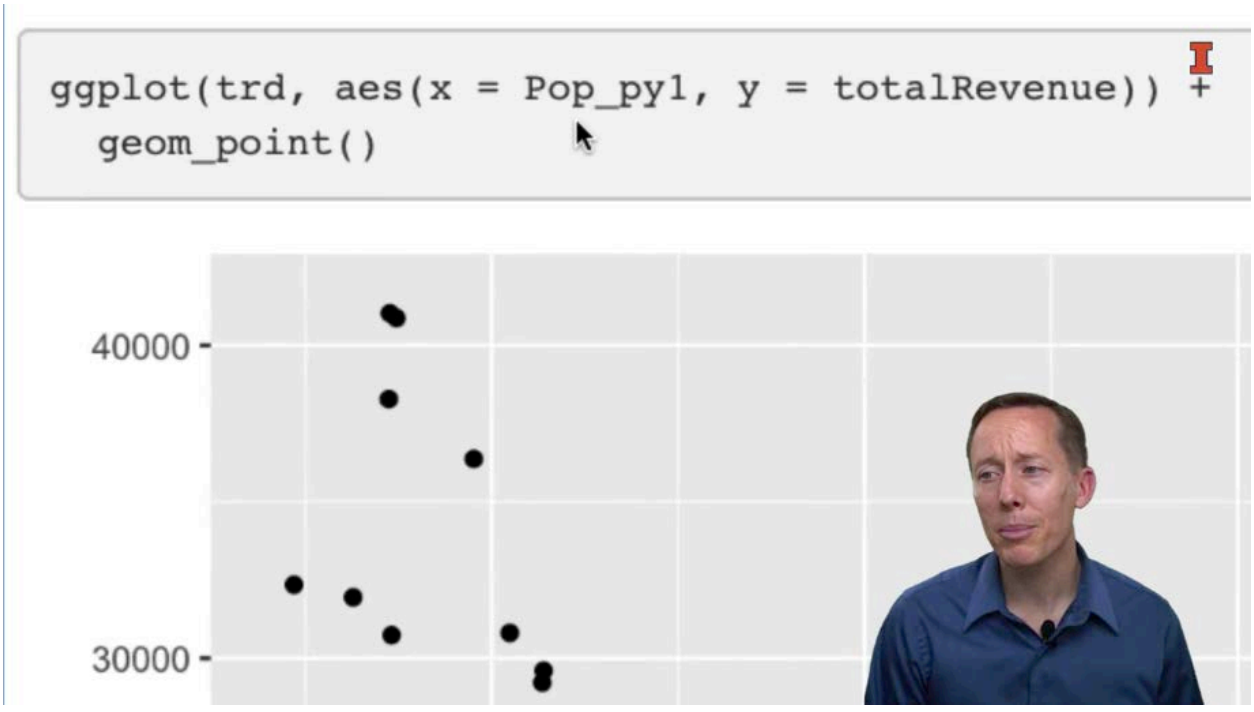
between our key dependent variable of interest. Which is total revenue and that represents quarterly sales. And we'll create scatter plots with that and some other variables, some of our other independent variables. And by tradition, the dependent variables are plotted on the y axis and independent variables on the x axis.



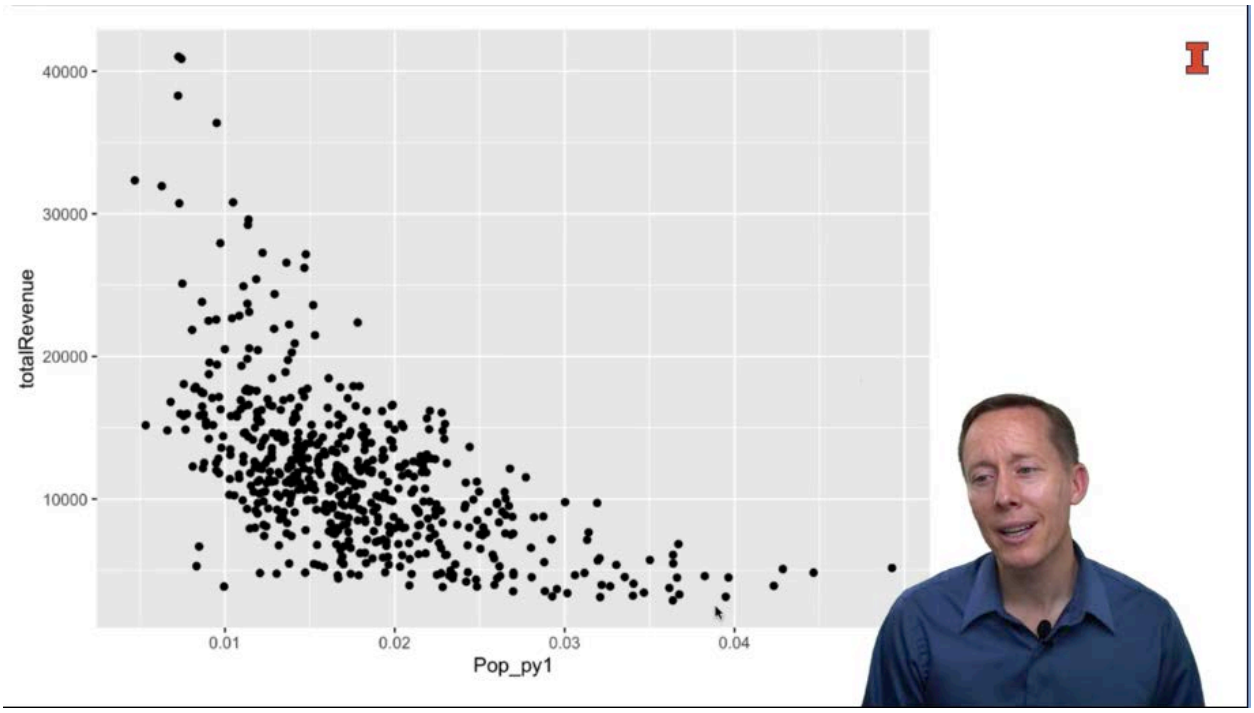
So you can see here that I've used the ggplot function on that trd data frame. And I've put Fuel py1 on the x axis and totalRevenue on the y axis. And then use the geom point to create a scatter plot. Now more important than creating the scatter plot in R, is understanding what it's telling us.



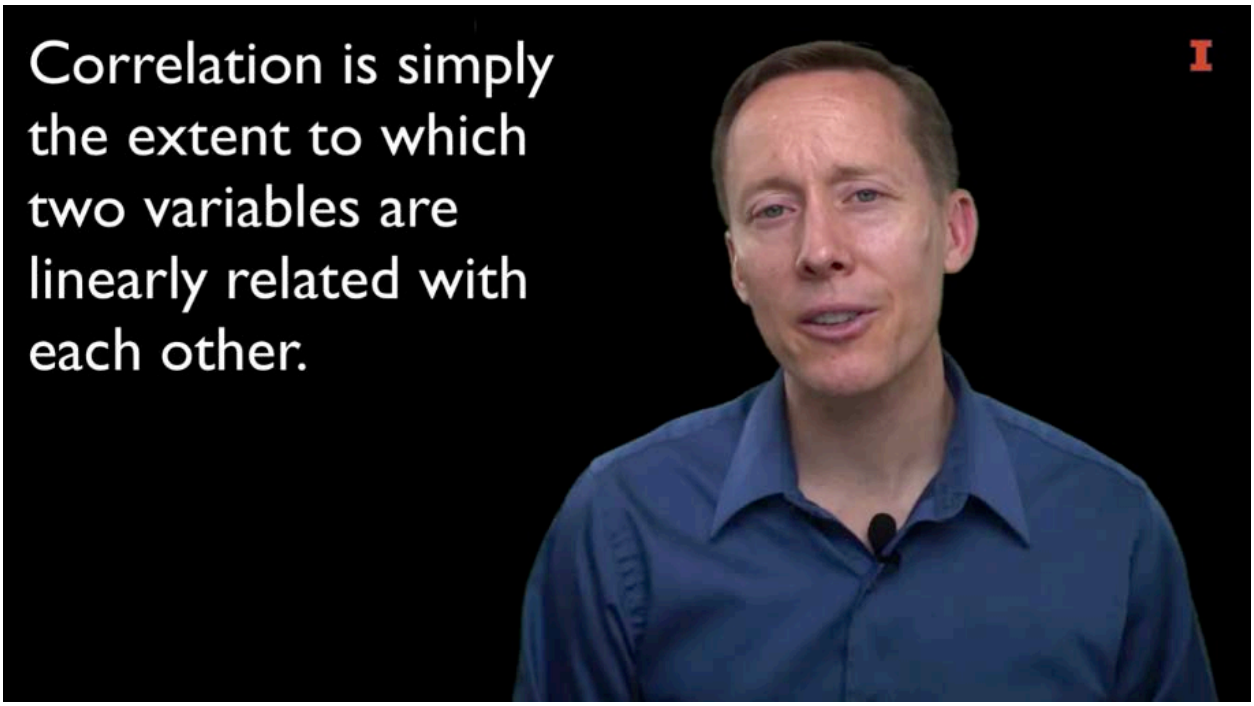
So the main thing that stands out to me is that, as values for fuel py1 increase so do values for total revenue in general, not all the time. But there's definitely kind of an upward trend here. Alright, an upward relationship or positive relationship between these two. And the scatter plot is helpful because it shows us that that's not always the case, but in general that is the case.



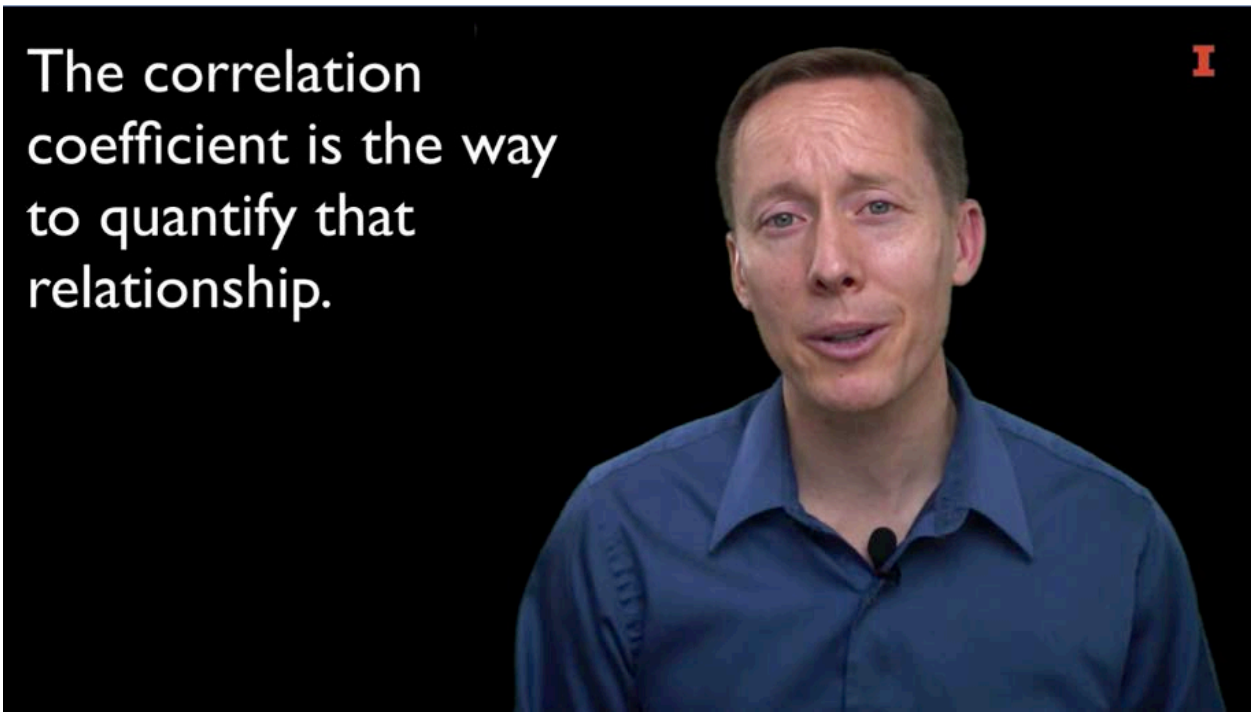
Now let's compare that scatter plot to scatter plot in which we put Pop py1 on the x axis.



And as we look at this relationship there is a clear downward slope, alright? So as the values for Pop py1 increase, the values for total revenue decrease. While these scatter plots are very helpful for communicating a nuanced relationship between two variables. Wouldn't it be great if we could quantify that relationship somehow? Well, we can and that's what correlation is all about.

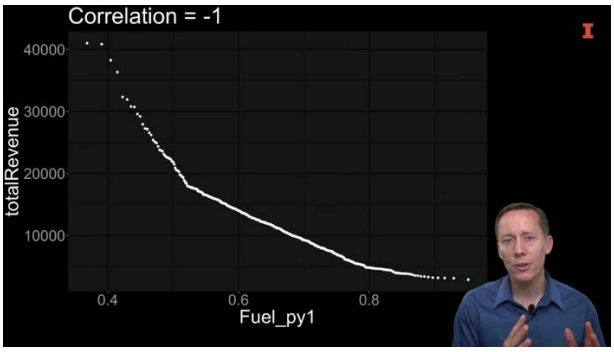


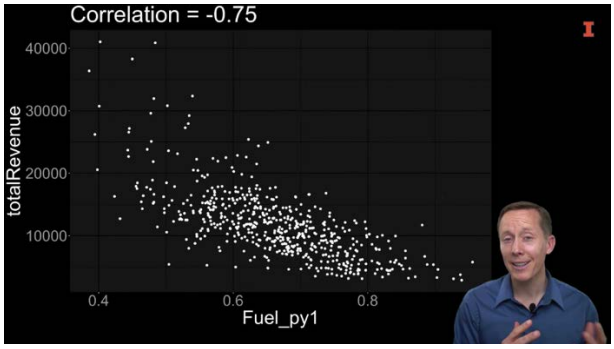
So correlation is simply the extent to which two variables are linearly related with each other.

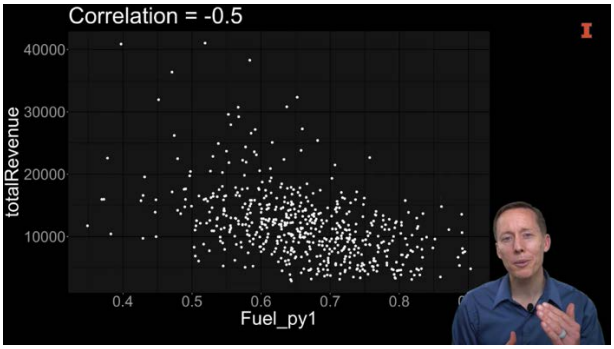


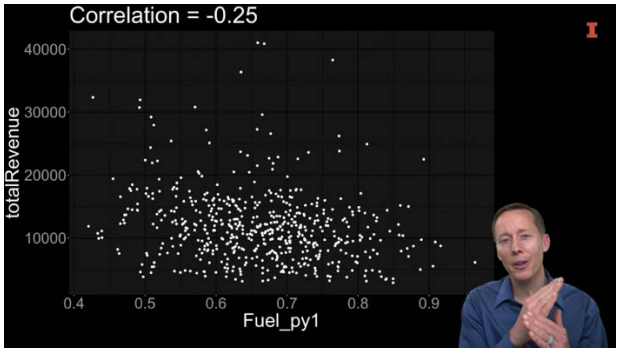
And the correlation coefficient is the way to quantify that relationship.

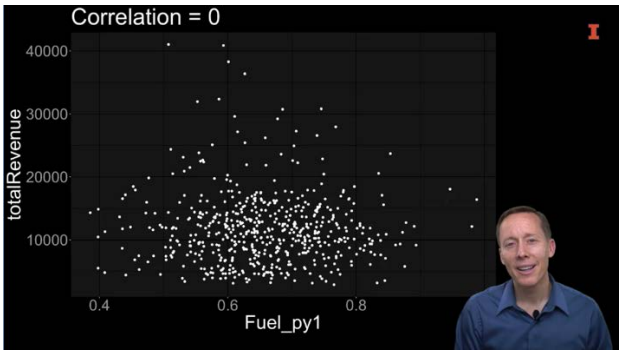
Correlation coefficients can range between -1 and 1, negative values mean that there is a negative relationship.

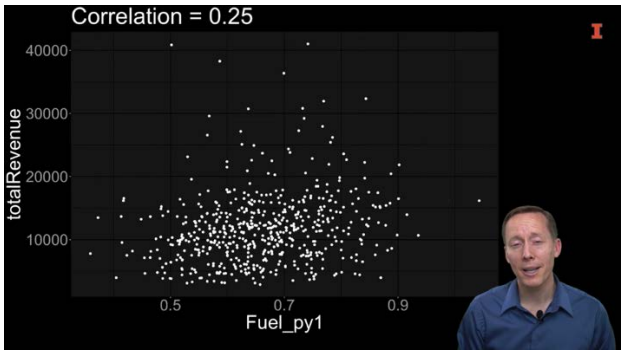

Correlation = -1
totalRevenue
Fuel_py1

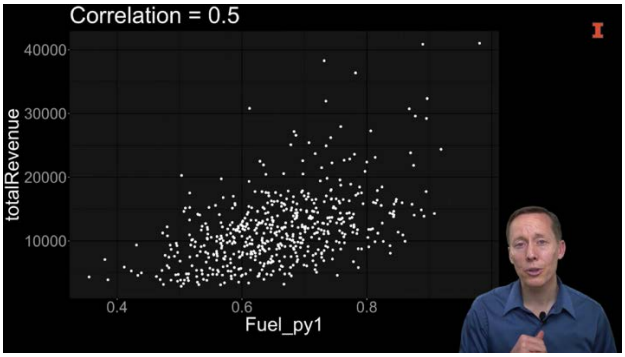

Correlation = -0.75
totalRevenue
Fuel_py1

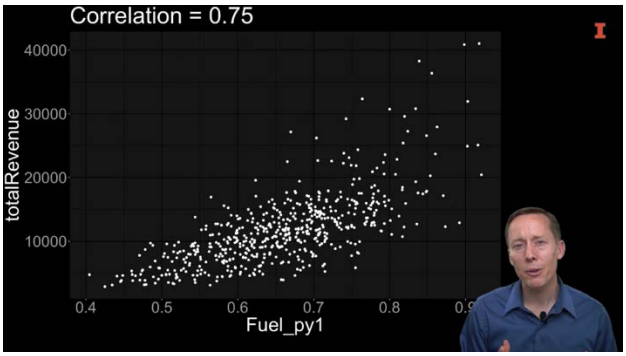

Correlation = -0.5
totalRevenue
Fuel_py1

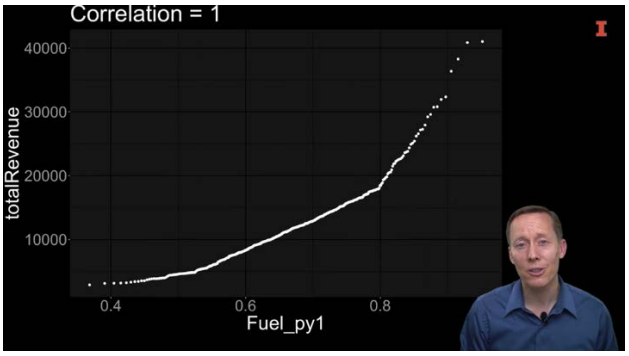

Correlation = -0.25
totalRevenue
Fuel_py1


Correlation = 0
totalRevenue
Fuel_py1

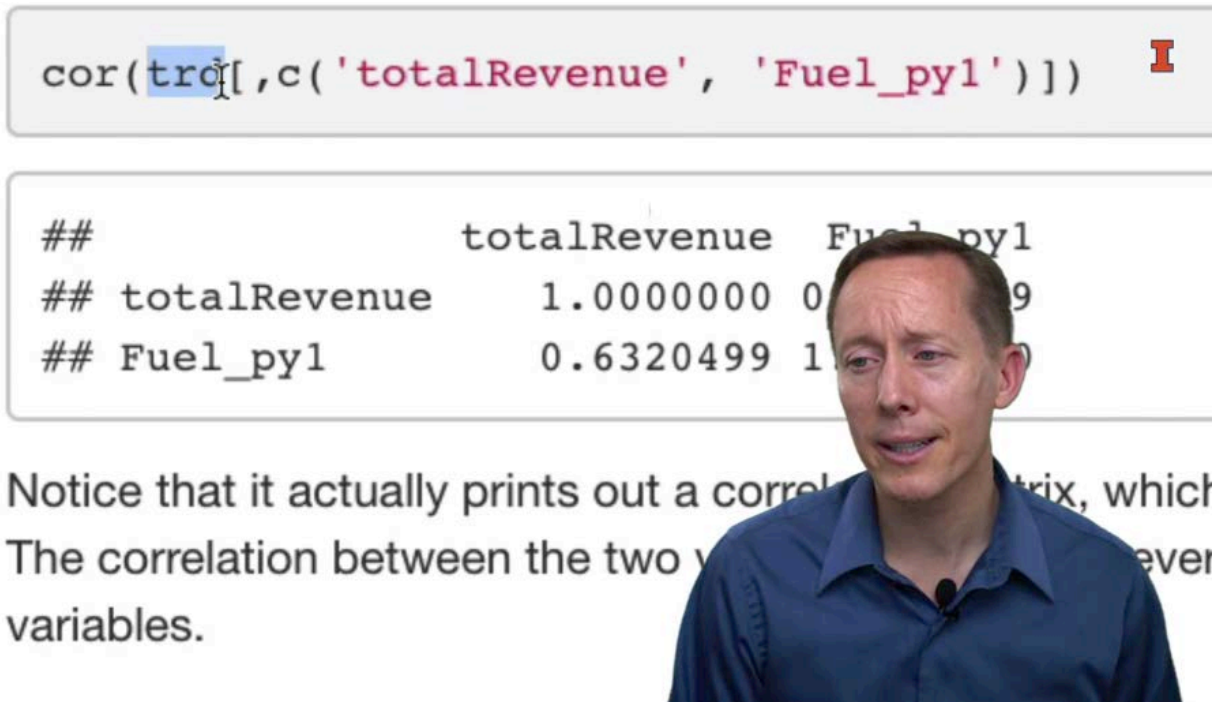

Correlation = 0.25
totalRevenue
Fuel_py1


Correlation = 0.5
totalRevenue
Fuel_py1

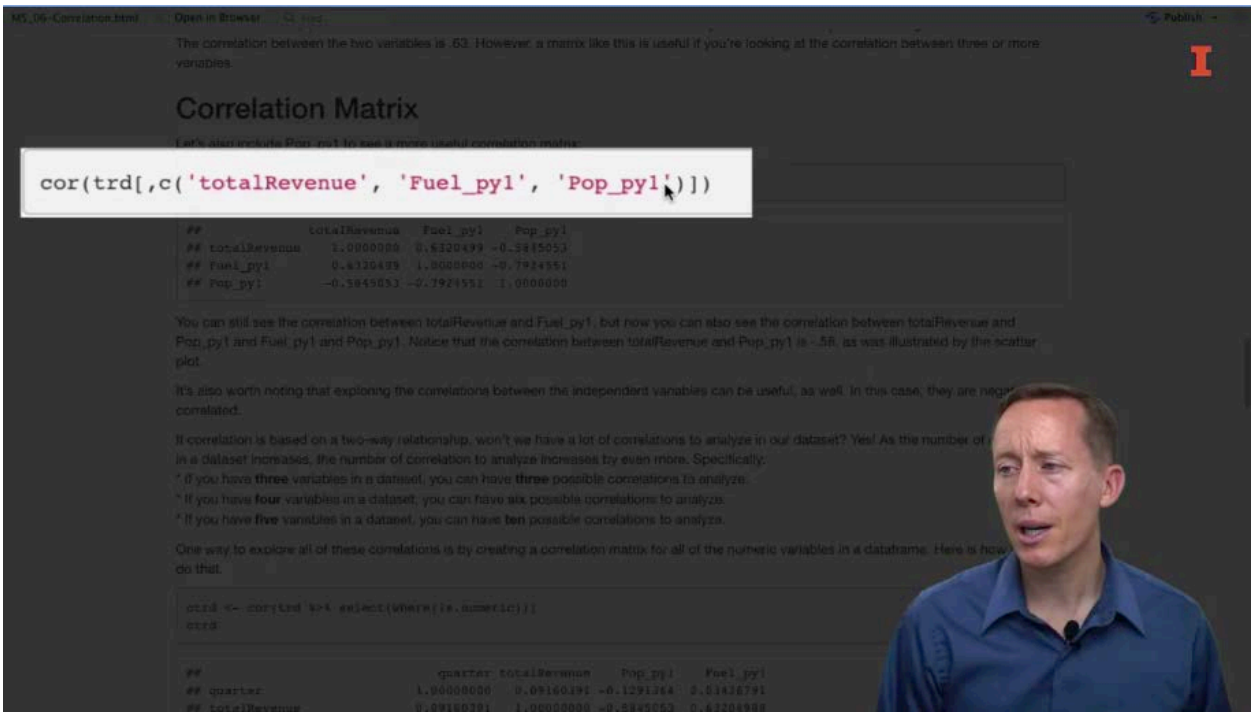

Correlation = 0.75
totalRevenue
Fuel_py1



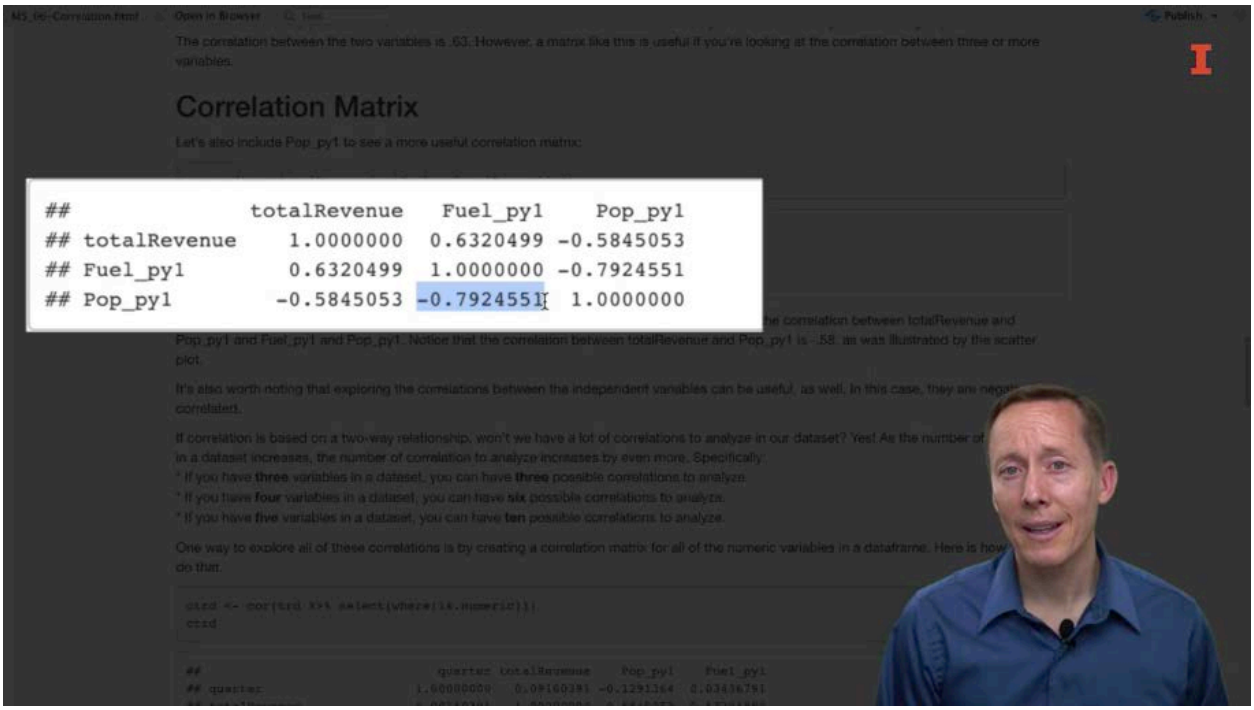
Or in other words, as one variable goes up, the other variable goes down. Positive correlation coefficients mean that the variables go up and down together. Values of zero means that there is no correlation or no linear relationship with each other.



Now, how can we calculate the correlation coefficient in R? It's very easy to do using the cor function, in this case, what I've done is I've taken the trd dataset. And I've subset it to only two columns, totalRevenue and Fuel py1. And it has created this correlation matrix which has a lot of unnecessary information with just these two variables. Really, the correlation coefficient is this 0.63, and you can see it's in there twice. It doesn't matter if you'll comes before total revenue or not, it's going to be the same. Now a correlation matrix can be very helpful when we're exploring a pattern of correlations between three or more variables.

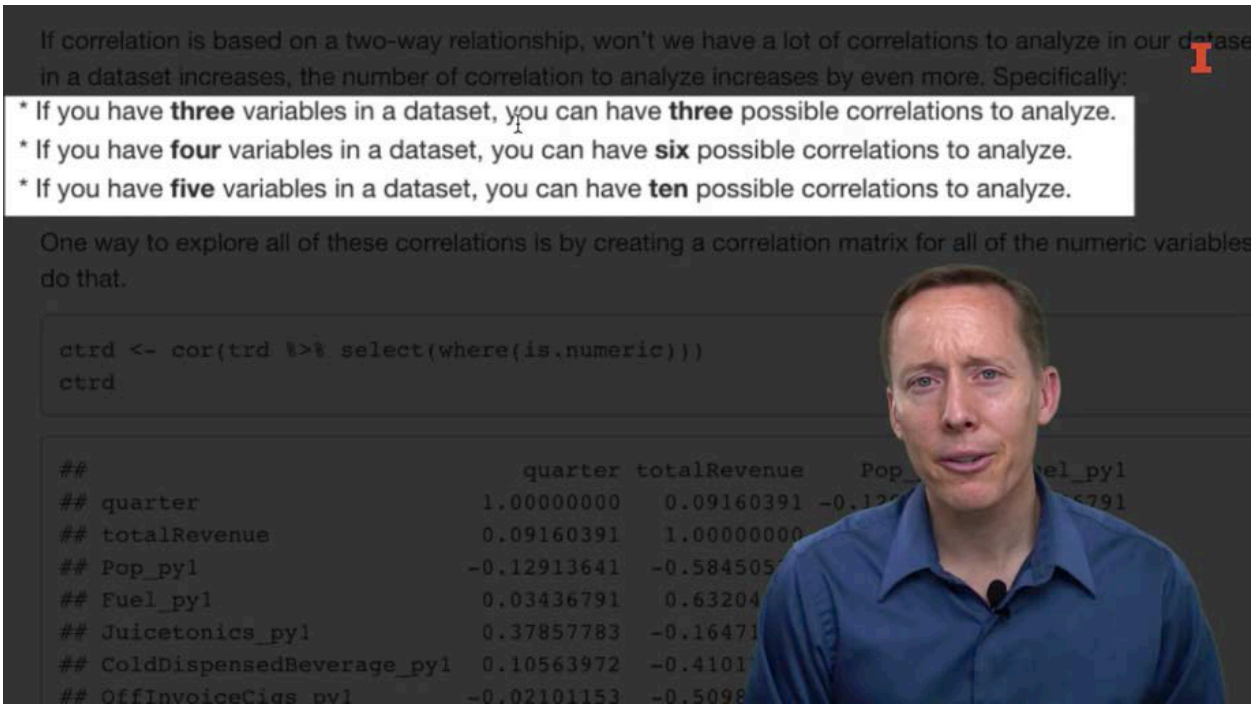


So let's go ahead and add Pop py1 to these other two columns and create a correlation matrix.

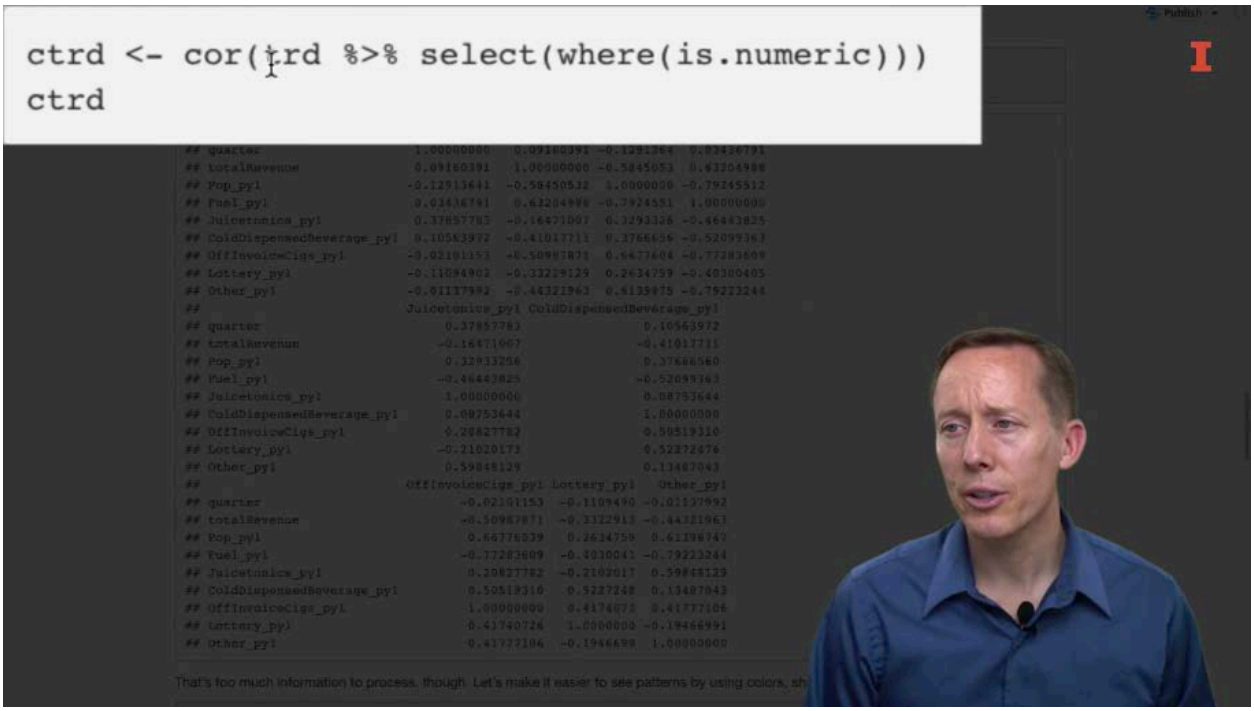


Now we can still see the correlation between totalRevenue and Fuel. Which is still .63, but we can now also quantify that negative relationship that we saw in the scatter plot between Pop and totalRevenue as- 0.58. So it's still a pretty strong relationship, but it's a -1. Now, it's also worth exploring the relationships between the other independent

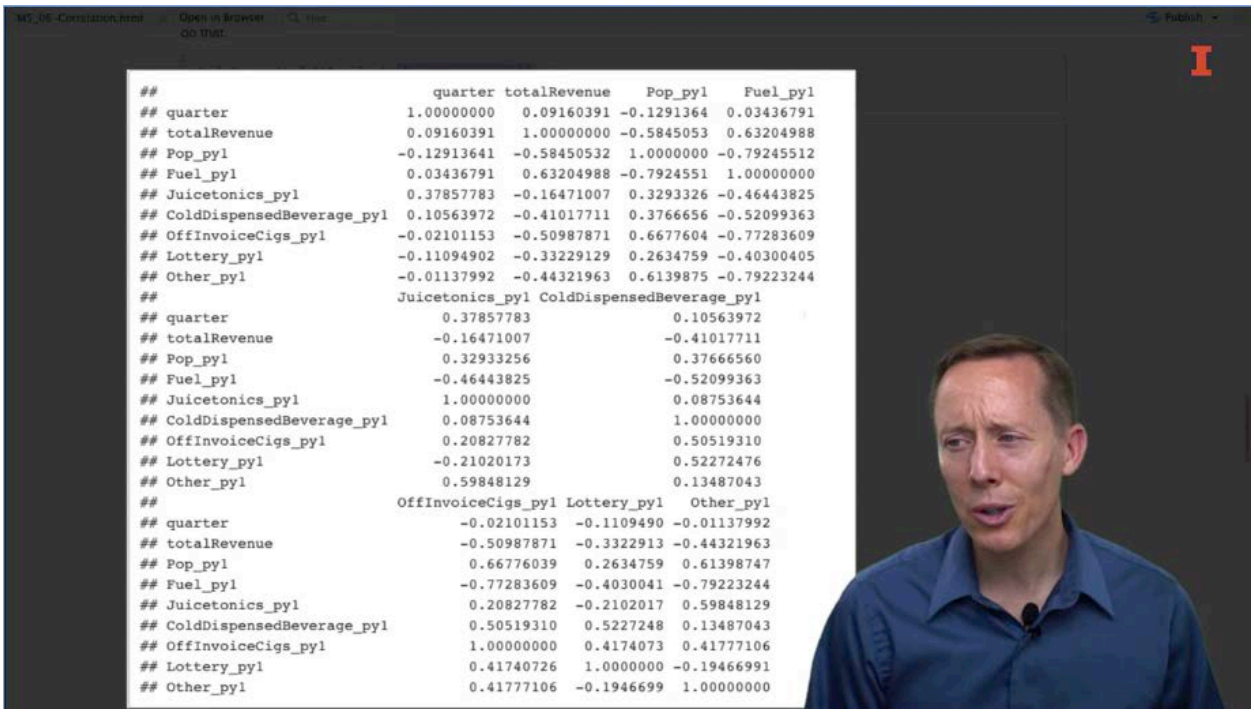
variables. In this case Fuel py1 and Pop py1 and it's a very strong negative correlation. So that's helpful because if we're trying to explore what is really driving totalRevenue, we may not need to include both of those variables.



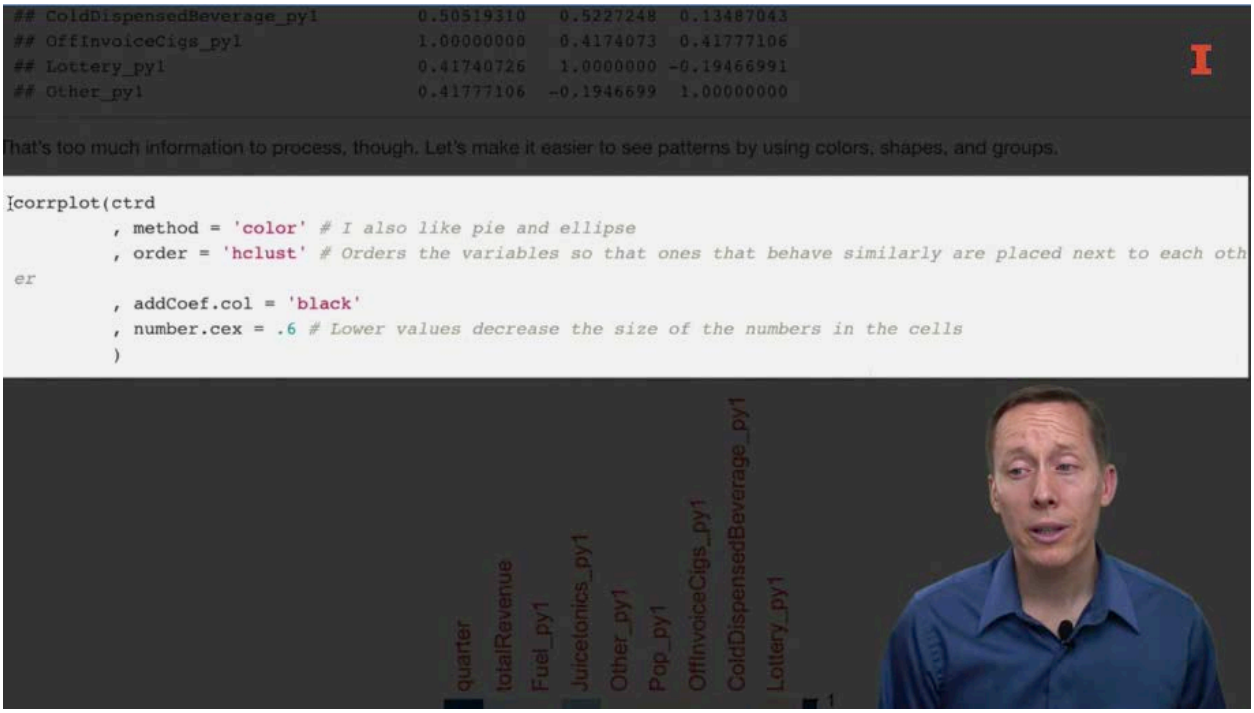
Now it's also worth asking, well, won't this create a lot of correlations when we look at all the two way correlations between all of the variables, and yes, it will. In fact, the number of correlation coefficients increases exponentially.



So you can easily create a correlation matrix with all of the two way relationships in R. And you can see here that I've done that by taking the trd data frame, piping it into the select function that selects different columns of data. And then I basically said we're only going to identify or include those columns for which they have a numeric data type.

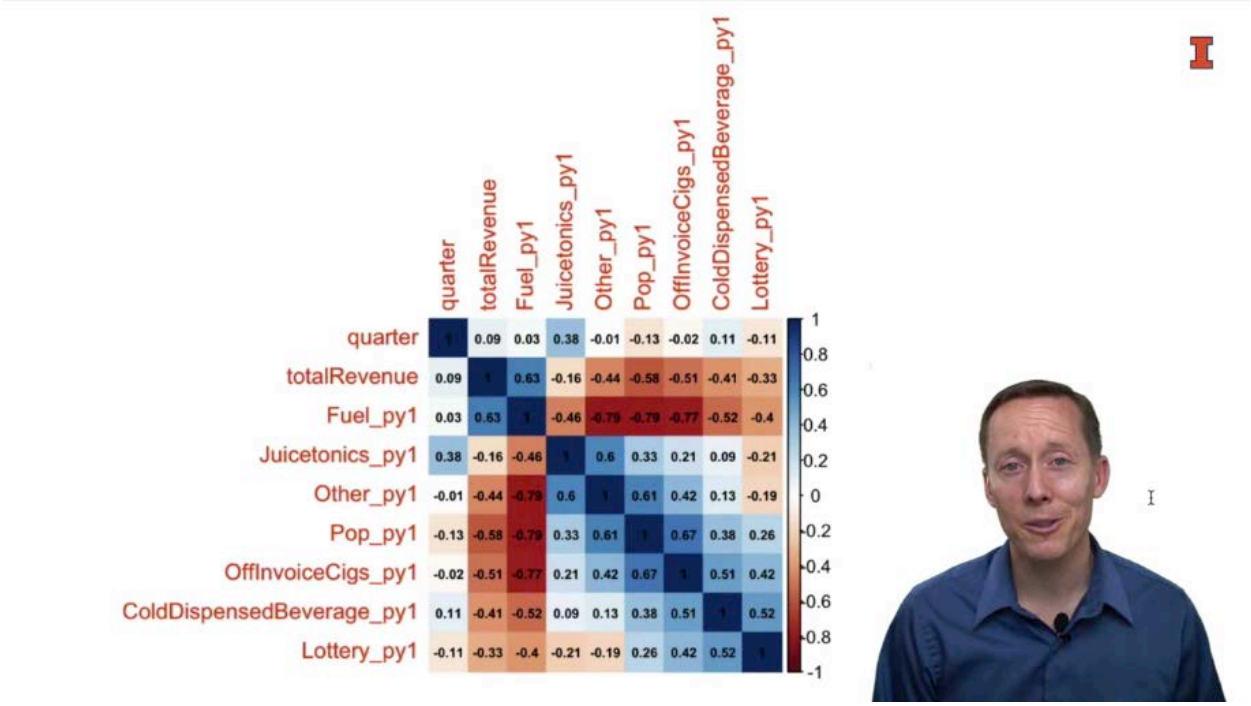


And that results in the ctrd correlation matrix, which is really big and overwhelming. So we could spend a lot of time exploring all of these correlations. But what I find to be much more helpful is to visualize those correlations.



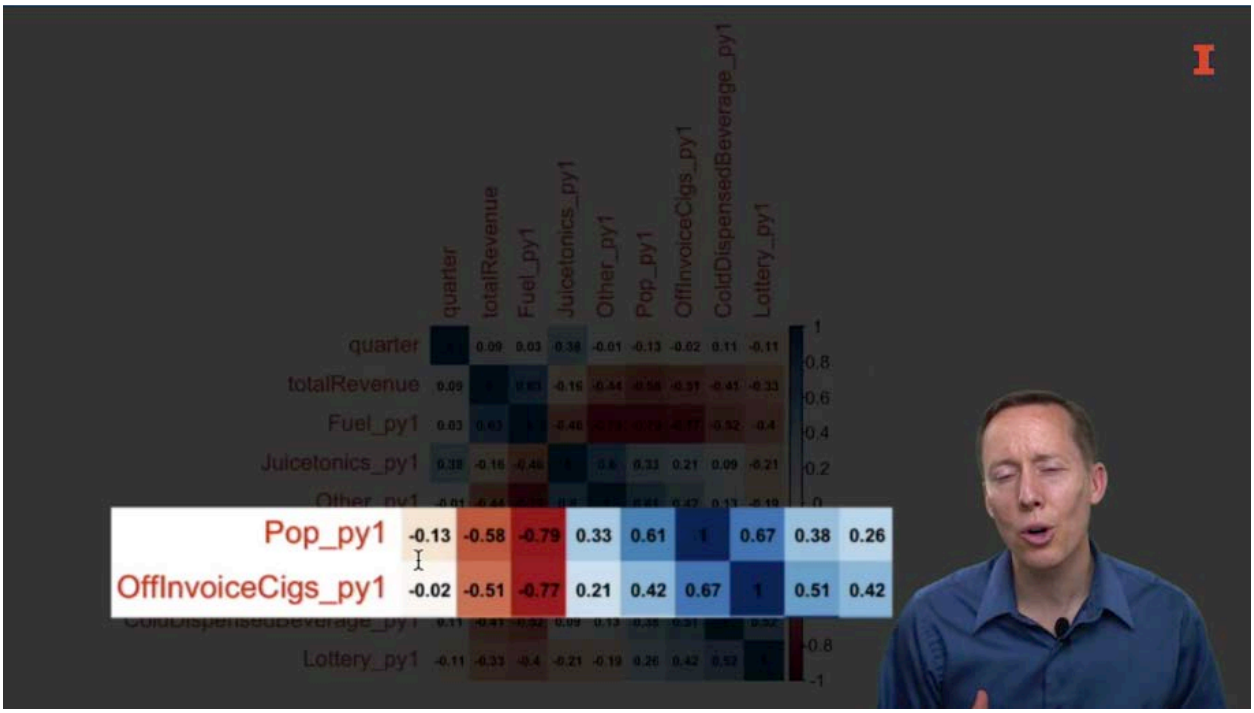
So here's where we will use the corrplot package, we'll take the corrplot function from that package, which is kind of the key function. And we will insert the ctrd correlation matrix that we just created. And then we will set the arguments to customize what this

correlation matrix visualization will look like. So we're going to color the cells of this matrix. We will order the variables using a hierarchical clustering algorithm, which basically just means we're going to put variables that have similar correlation patterns together.

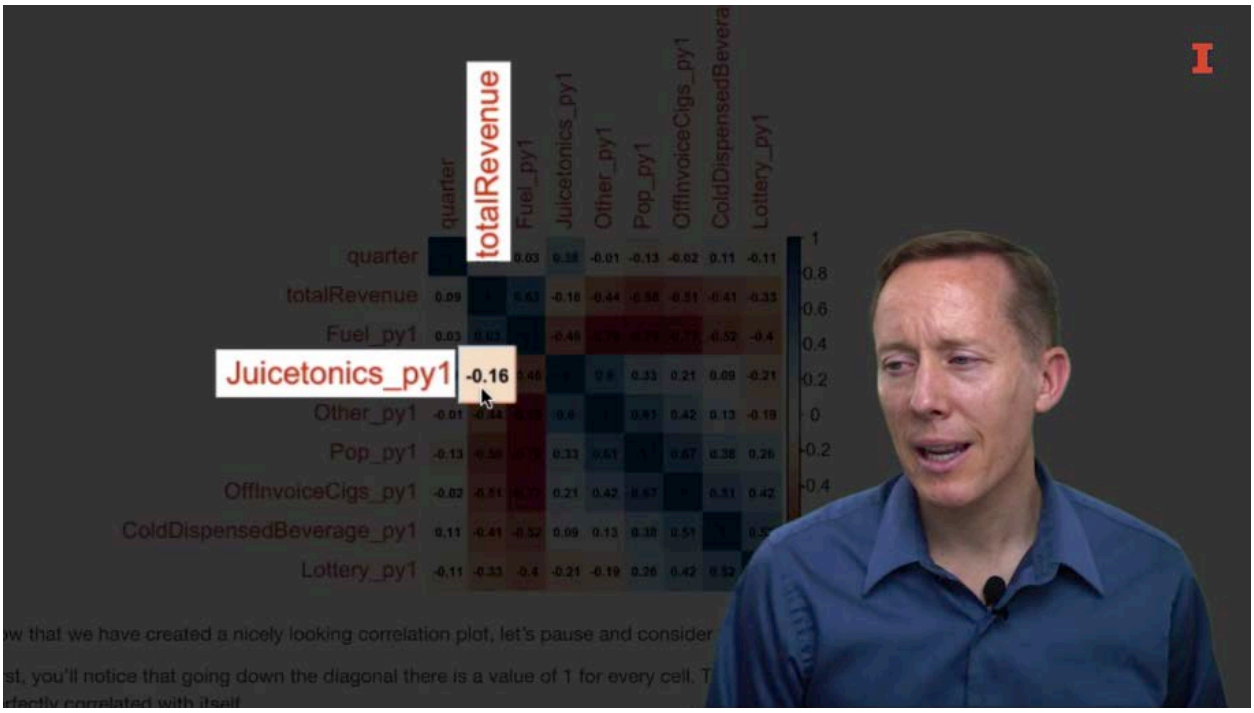


And then we'll adjust what the numbers in the correlation matrix look like and that will result in this correlation matrix right here. Now I think this is really pretty and there are a few things that stand out to me. First of all, you'll notice that along the diagonal, there's always a 1 there. And that 1 is always blue, and that just means that each of these variables is positively correlated to each other. That's no surprise and that will always be the case. The 2nd thing is just to recognize the three different colors in this correlation matrix. You have shades of blue which means positive correlations, white which means no correlation. And then shades of red which means negative correlations. So the darker the shade, the more strong is that correlation. So then you can see that there is a pattern in the reds and blues here. Because we have used that H cluster argument to organize the variables not alphabetically, but based on the pattern of correlations. Now, your next question might be, how do I use this correlation matrix?

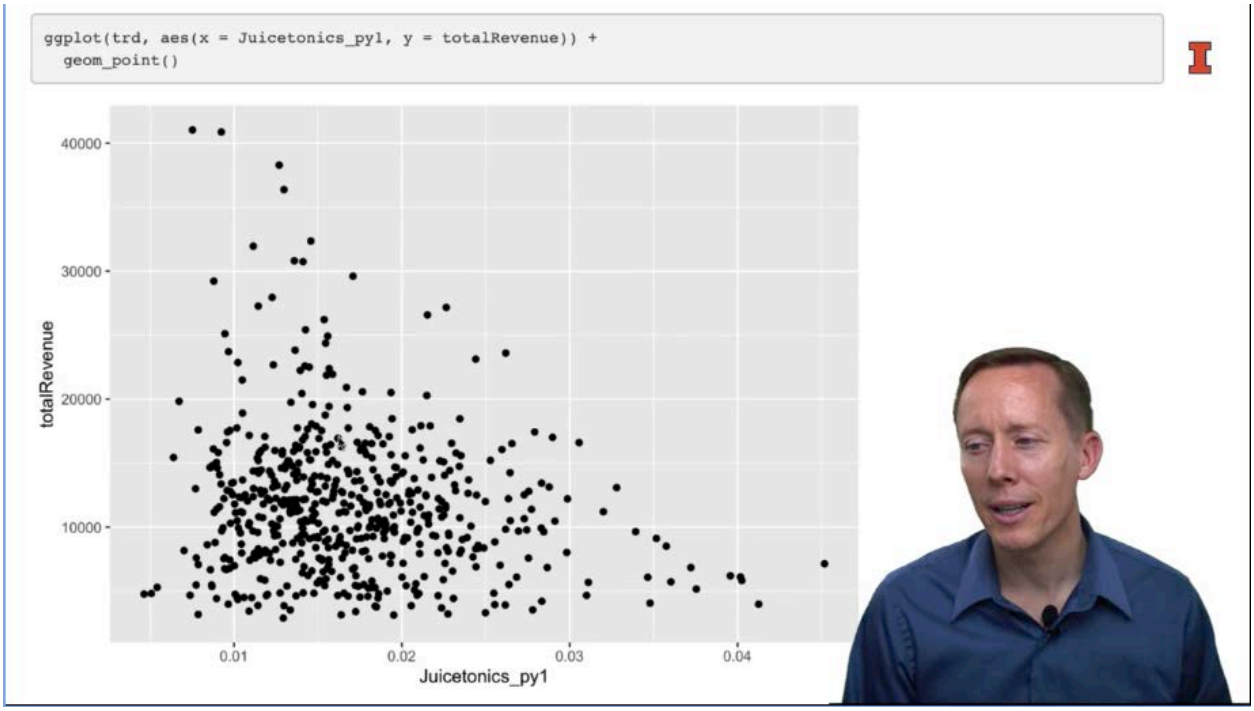
Well, this matrix can be helpful for really exploring the relationships between total revenue and the other variables. So we can see here just based on the color that Fuel py1 has the strongest positive relationship with total revenue. In fact, the only positive relationship that is worth noting. Quarter is slightly positive related but that's not a very strong correlation of 0.09. We can also see that pop has the strongest negative correlation. But there are some other variables that also have a significant negative correlation, like OffInvoiceCigs and ColdDispenseBeverages.



The other thing that you might want to look at is the relationship between some of these variables. So for instance, if you look at Pop and OffInvoiceCigs. The patterns of colors for both of these are very similar and what that means is basically that they go with each other. If you know one, you'll probably have an idea of what the other one is a really good idea. Now that we've talked about what a correlation matrix is and how to visualize it. Let's conclude our discussion of correlation by going back to scatter plots.

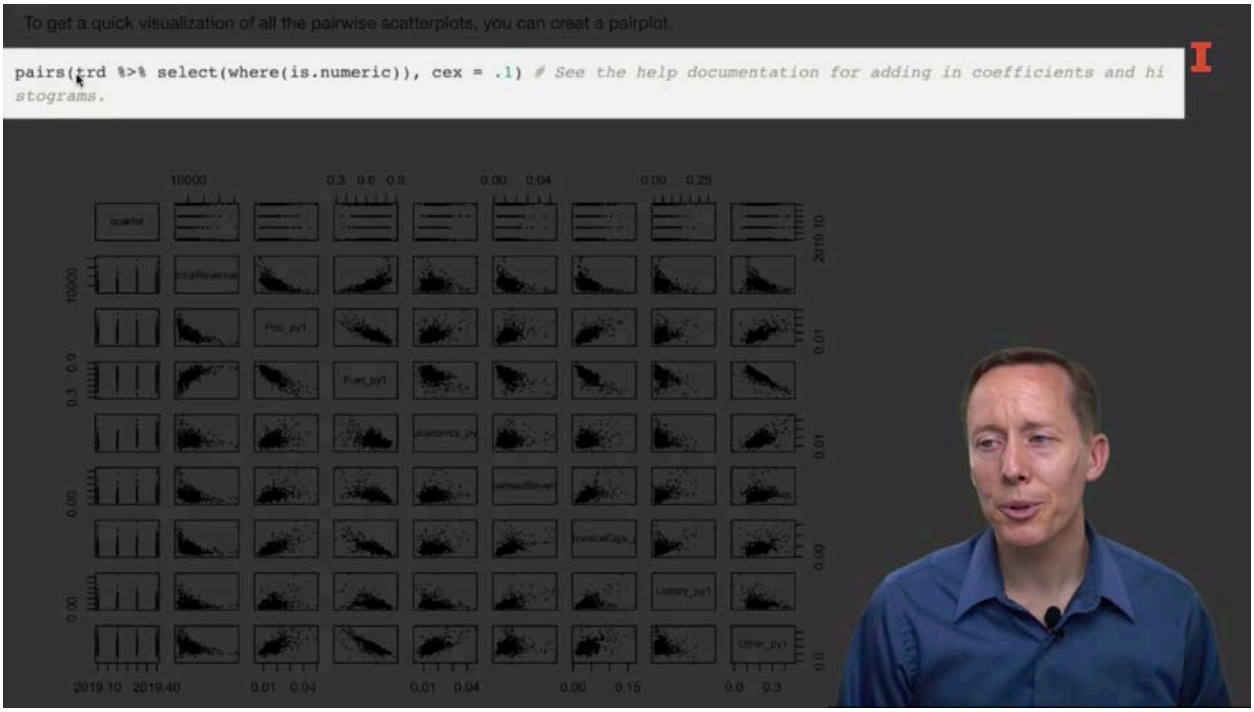


And let's look at a scatter plot between totalRevenue and juice and tonics because this has a weak correlation is close to 0 and it's negative. All right, so we should expect to see a negative slope to the scatter plot with the point spread out quite a bit.

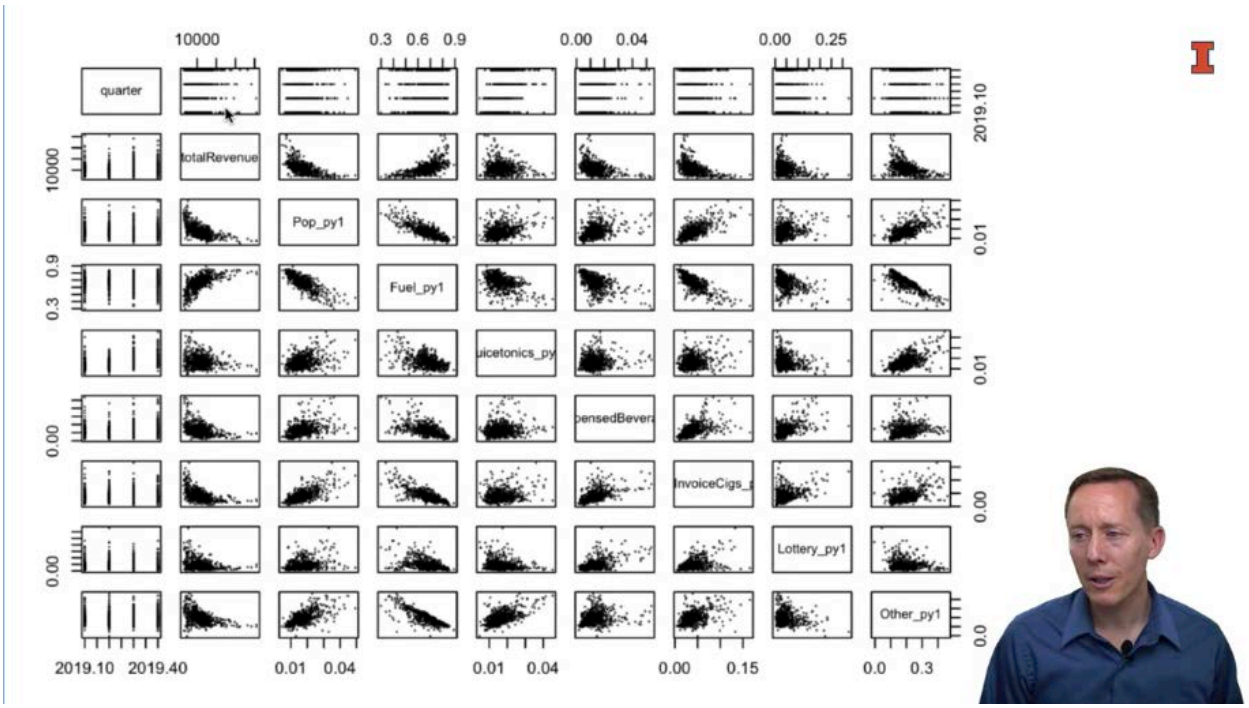


And if we create the scatter plot, that is exactly what we see. All right, so this is a weak negative correlation. Now, it may be useful to compare this scatter plot with the scatter plot between totalRevenue and Pop. Which also has a negative correlation but a much

stronger one. Well, we can actually create a matrix of all the scatter plots in this data set by using the pairs function.



And it's very simple to use, we just input the data frame and indicate which columns we want.



And then it will create this nice looking scatter plot and there's a lot in here to process. But if we look at this totalRevenue column and compare just three of these scatter plots here. We see the Pop py1 in totalRevenue relationship that we looked at before. And we can compare that to the juicetonics and definitely see that they're both have negative slopes. But the juicetonics has points that are much more spread out there, not as close to each other. And then of course there's the positive correlation between totalRevenue and Fuel py1. So hopefully these scattered plots and correlation coefficients, you can see how they are related to each other. Basically just like descriptive statistics go very well with histograms and box and whisker plots for instance. Correlation coefficients are very useful in conjunction with scatter plots. Scatter plots allow you to see whether or not an outlier maybe driving a correlation or not. However, there is so much nuance in there that they're kind of hard to communicate. And so the correlation coefficient allows you to quickly communicate the extent of the linear relationship.

Lesson 1-4.2 Linear Models

Model planes are smaller, simpler versions of real planes. They can be really helpful for converting an abstract idea into a concrete vision that can be shown and discussed with others. Some models of vehicles are even used in wind chambers for predicting how the full-sized version will respond to hurricanes when moving at high speeds.

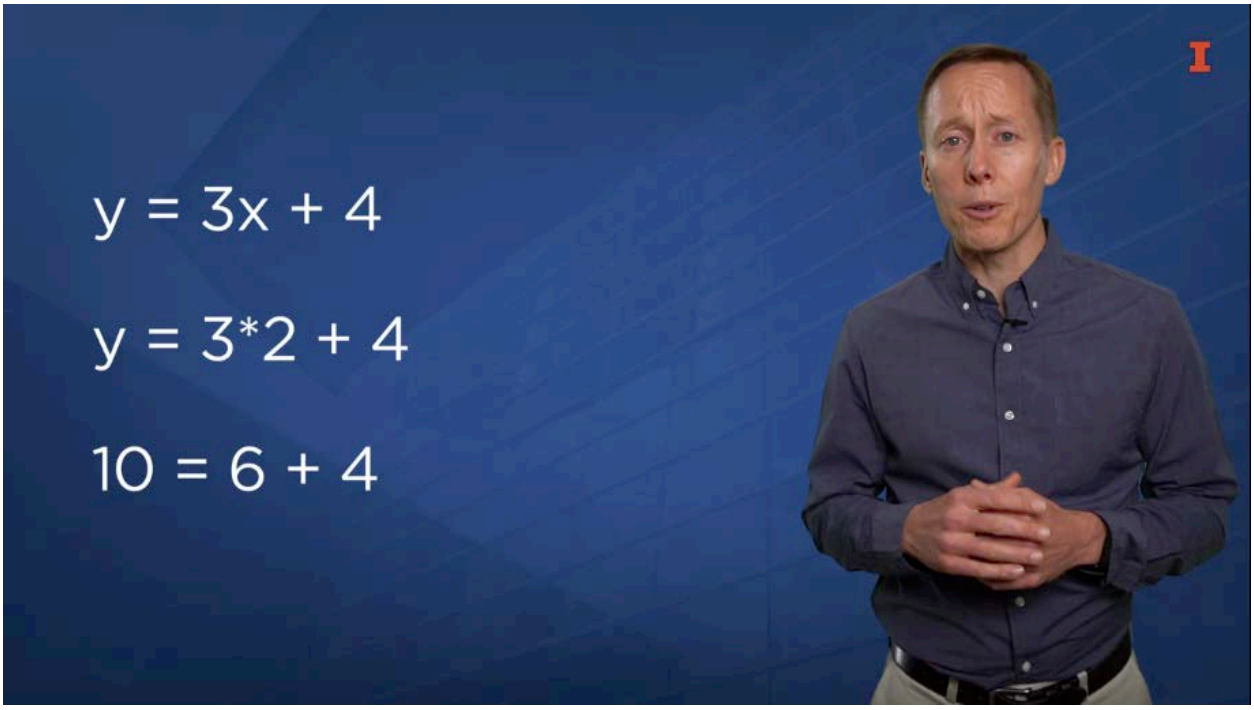
In a business context, models are often used to help convert an abstract mission into a clear vision and to evaluate how the business will respond when moving at high speeds.

Budgeting is an important example of a business model.

In a business analytic context, various models are used for predicting, how a business will respond to changes in different parameters, and for understanding how those parameters influence business decisions.

One of the most common and fundamental models is a linear model. A linear model is something with which you're probably very familiar. It's essentially a linear function with parameters that are created based on relationships in historical data.

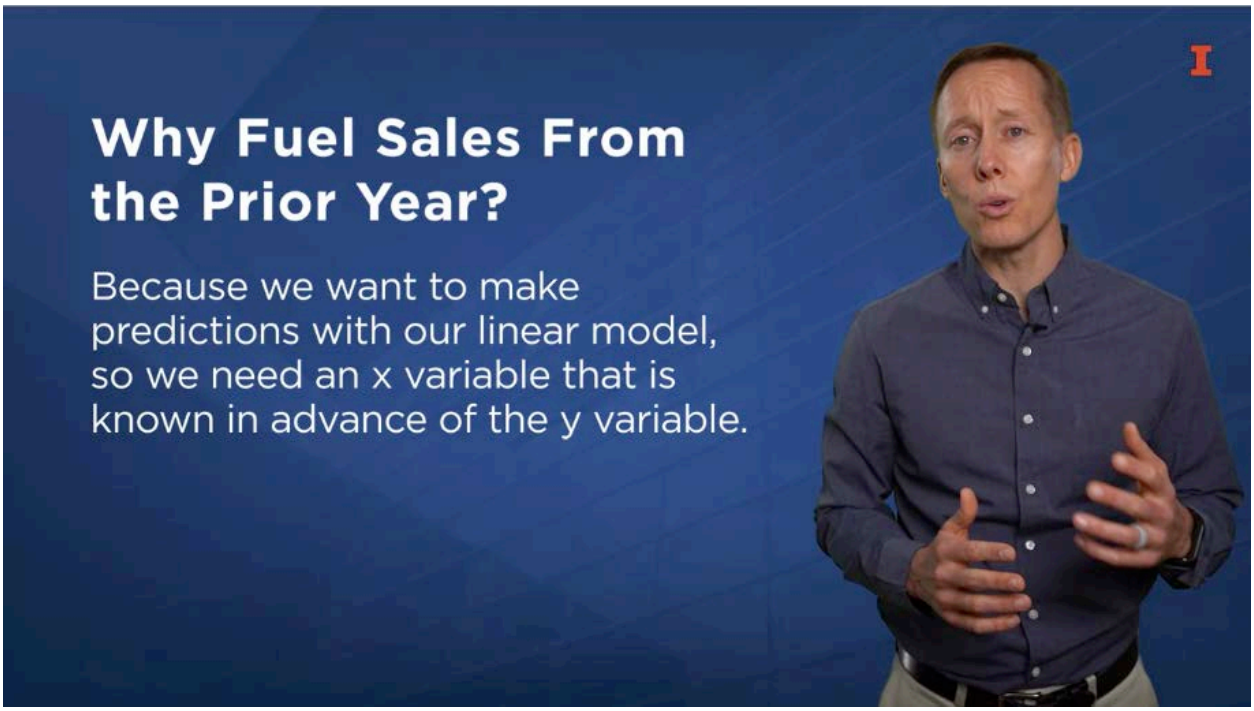
Let's briefly review a linear function. In many primary, middle, and high schools, students learn a general form of the linear function, y equals mx plus b. This is the slope intercept way to represent a line on a two-dimensional graph. The beauty of this linear function is that it expresses what y will be given a value of x in very simple terms.



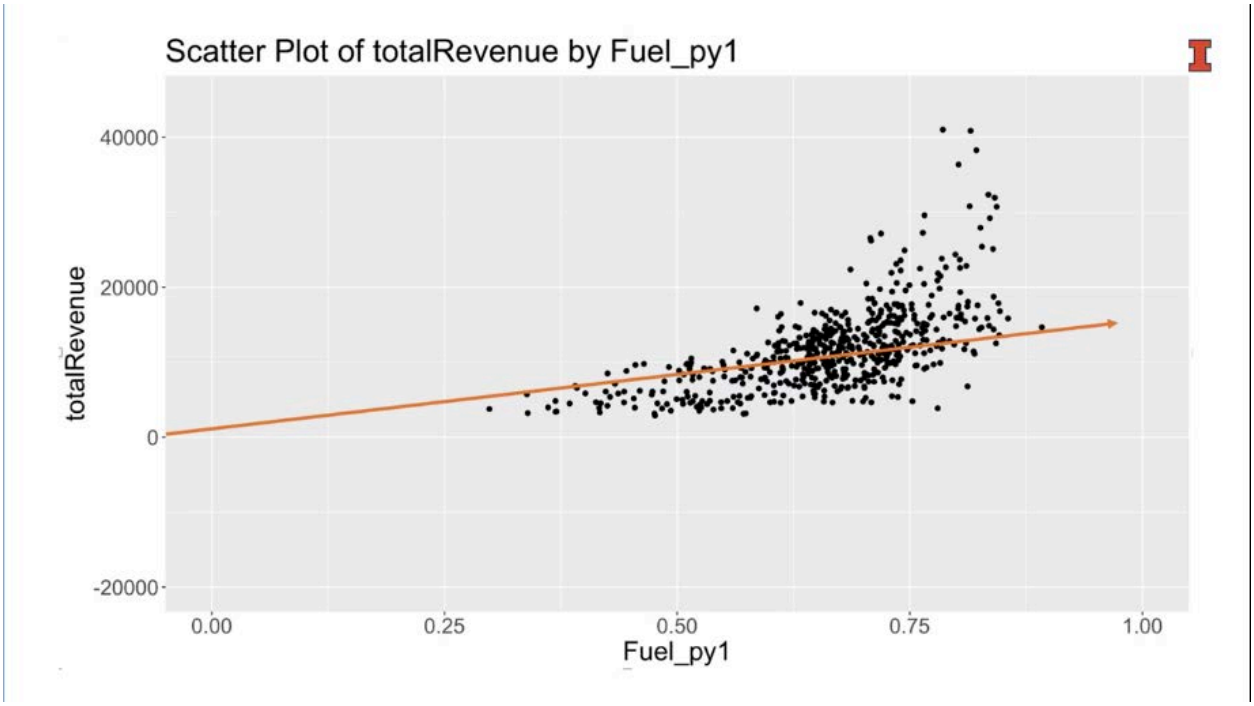
For the linear function, y equals 3x plus 4, 3 is the intercept and 4 is the slope. I know that if x is equal to 2, then y will be equal to 10. When you create a linear model for

business analytics, the hope is that you can capture much of the variation in the y-variable or the dependent variable by setting the slope and intercept parameters such that if we know the value of x, which is the independent variable, then we can estimate the value of y. Now, that's abstract.

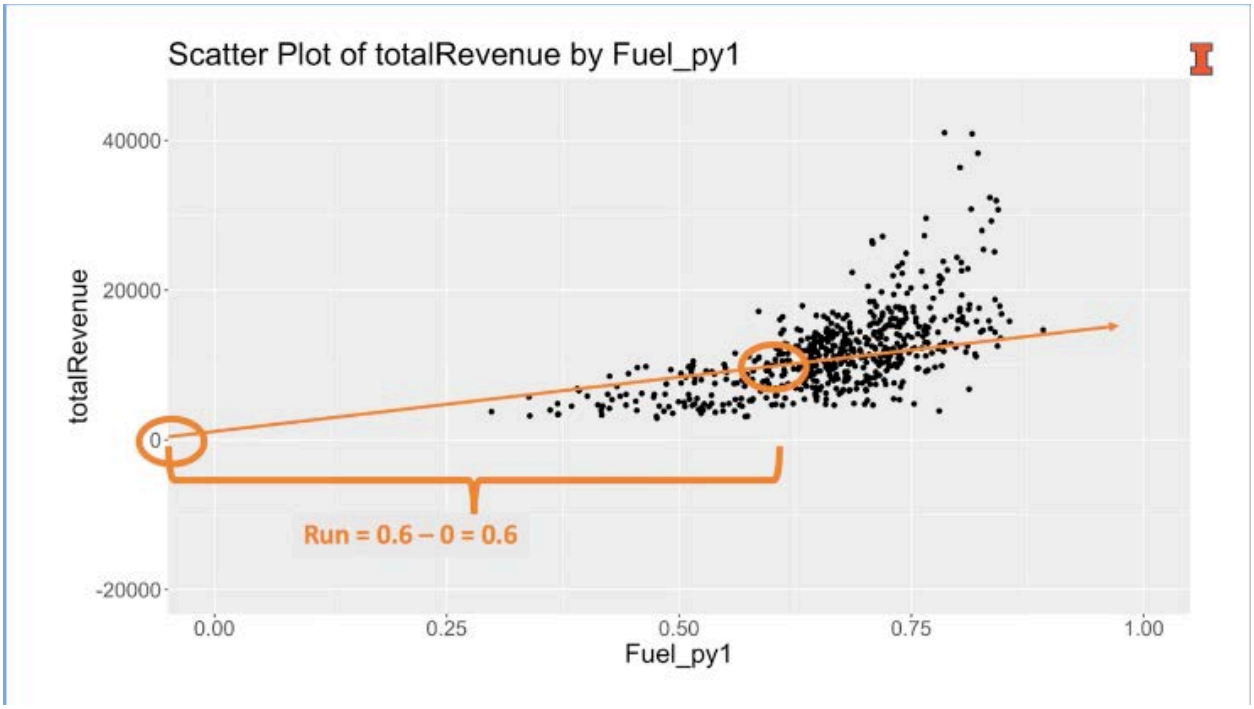
Let's talk about it in the context of our business problem, which is how our quarterly sales affected by quarter of the year, region, and by product category or parent name. The y-variable would be quarterly sales. The x-variable could be the percentage of fuel sales from the same quarter during the prior year.



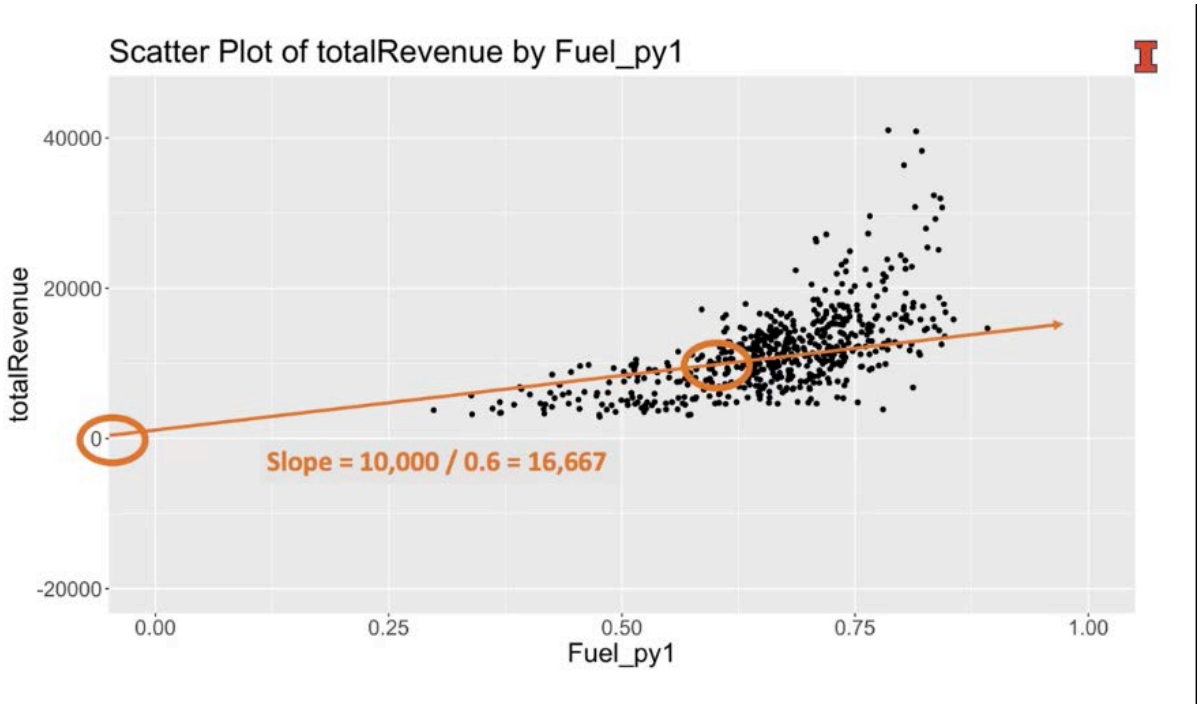
Why are we interested in the percentage of fuel cells from the prior year? Because if we want to make a prediction of quarterly revenue using a model, then we need to have access to the fuel sales sometimes before the quarterly revenue is known. Fuel sales from the prior year would allow us to make a prediction of quarterly revenue a whole year in advance. Let's start by visually creating a model.



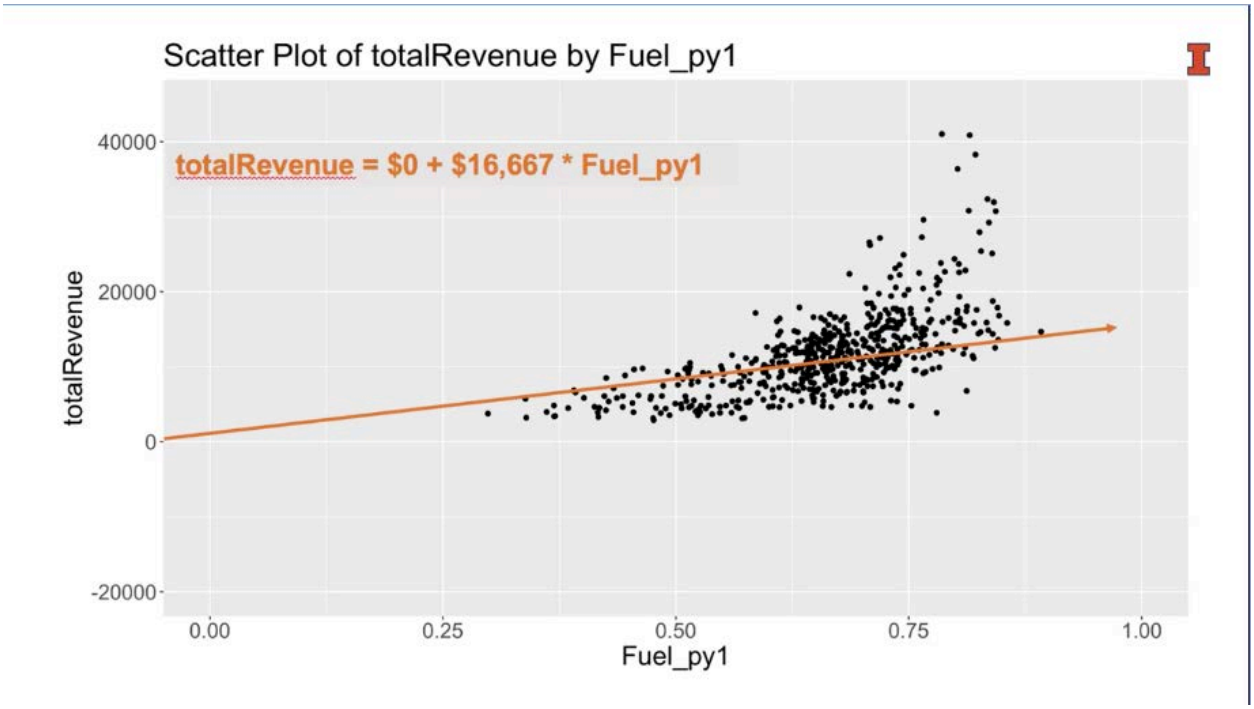
Here's the scatter plot of the quarterly sales variable, total revenue in our dataset, and the percentage of total sales from fuel during the same quarter of the previous year or Fuel_py1 in our dataset. Total revenue is on the y-axis because that's what we're trying to predict. Fuel_py1 is on the x-axis because that's what we're using to make predictions. Let's try to place a line on this chart that minimizes the sum of the distance between the data points and the line. This looks good to me. But I realized that you may have chosen a slightly different line. The intercept of this line is at about zero. This means that if we did not sell any fuel during the same quarter last year, we would still expect quarterly sales to be $0. Now let's calculate the slope by choosing any two points on the line. Let's use the intercept as one point. For the other point, let's choose one that's easy to identify. How about the one where the line crosses $10,000 in total revenue when Fuel_py1 is 0.6.



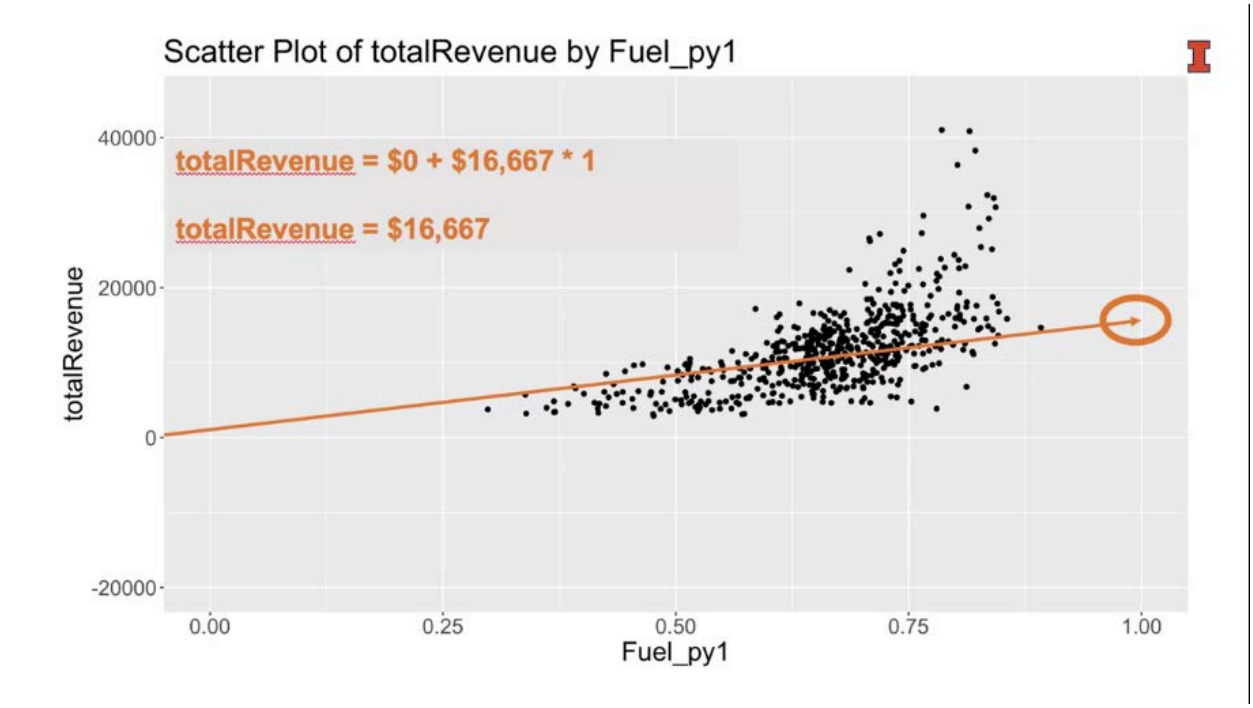
At this point, we can calculate the rise over the run. The rise is $10,000 minus zero or $10,000. The run is 0.6 minus zero, so just 0.6.



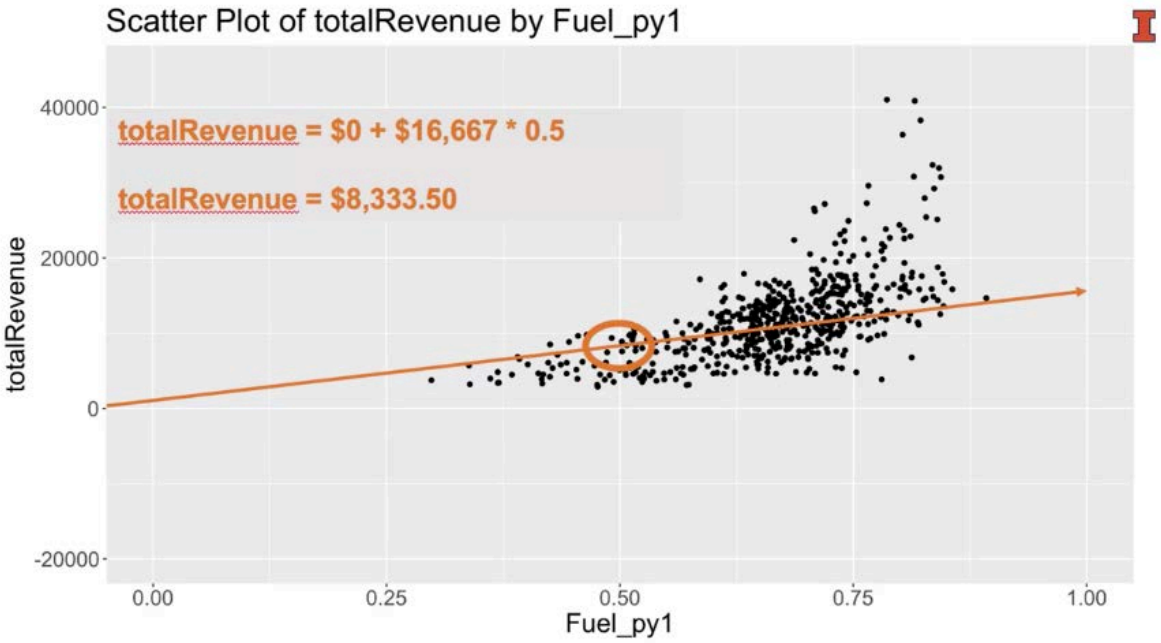
Ten thousand dollars divided by 0.6 equals $16,667, which is our slope. With these two parameters, we can create this linear model.



Total revenue equals $0 plus $16,667 times the Fuel_py1 value.

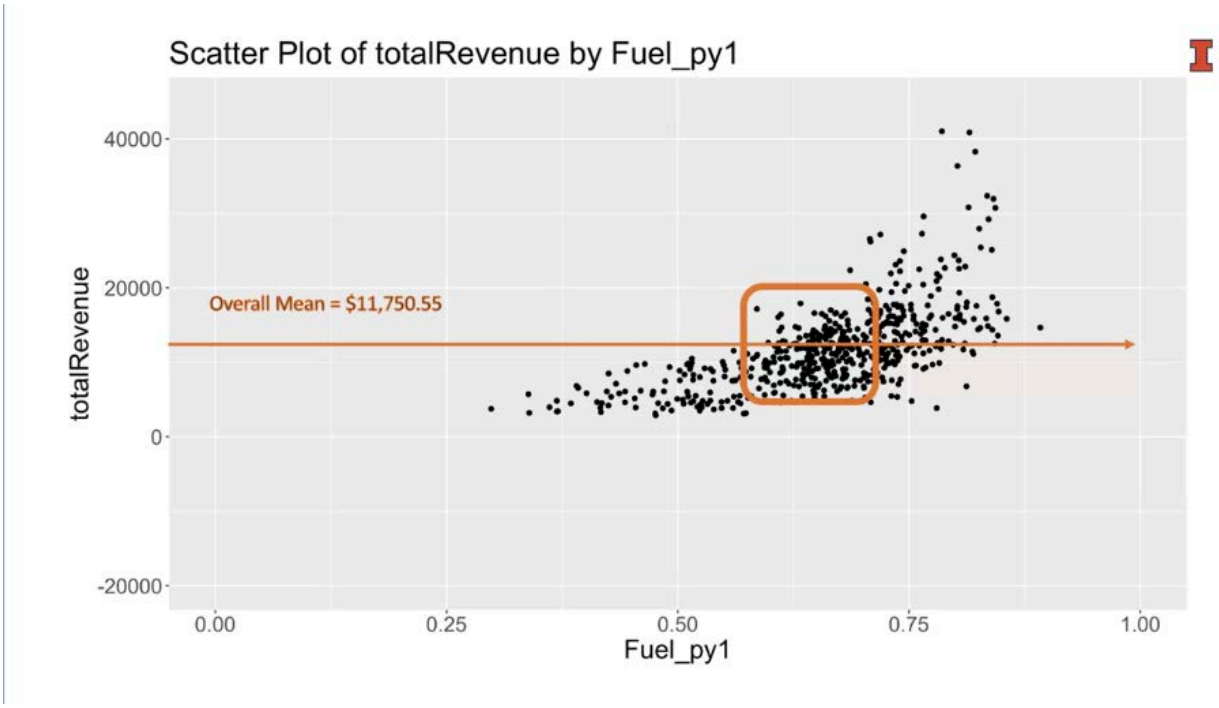


In other words, if fuel sales from the same quarter during the previous year was 100 percent from fuel, then we would expect our sales in the next period to be equal to $16,667, which is just the coefficient of 16, 667 times 1.

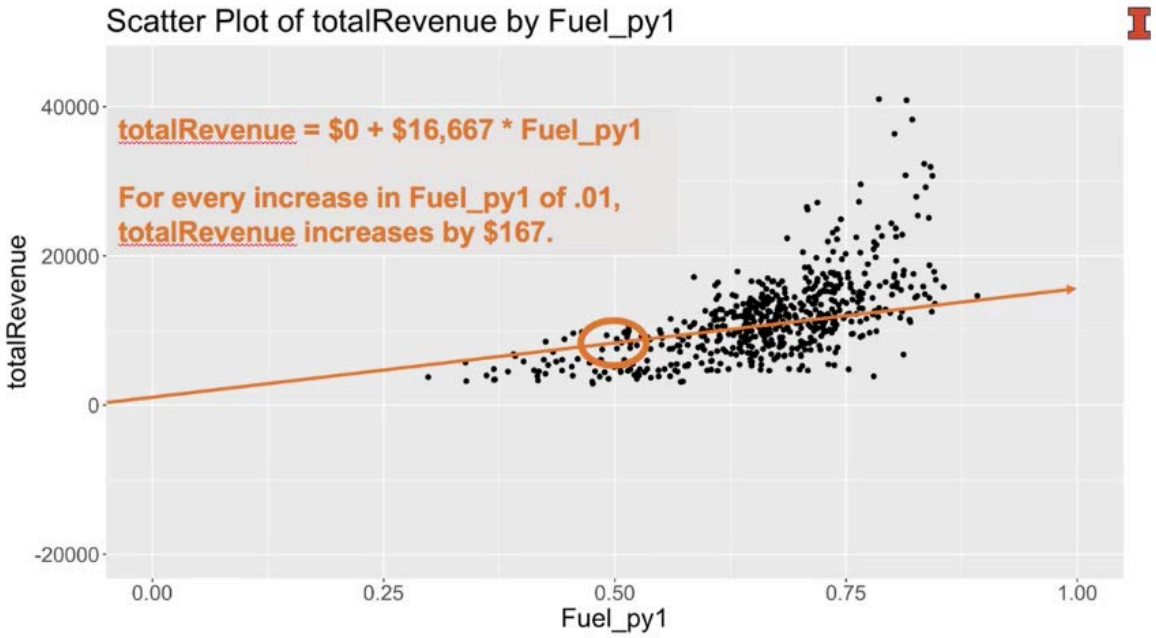


As another example, if the percentage of sales from the same quarter during the previous year was from 50 percent fuel, then we would expect our sales to be $8,333.5.

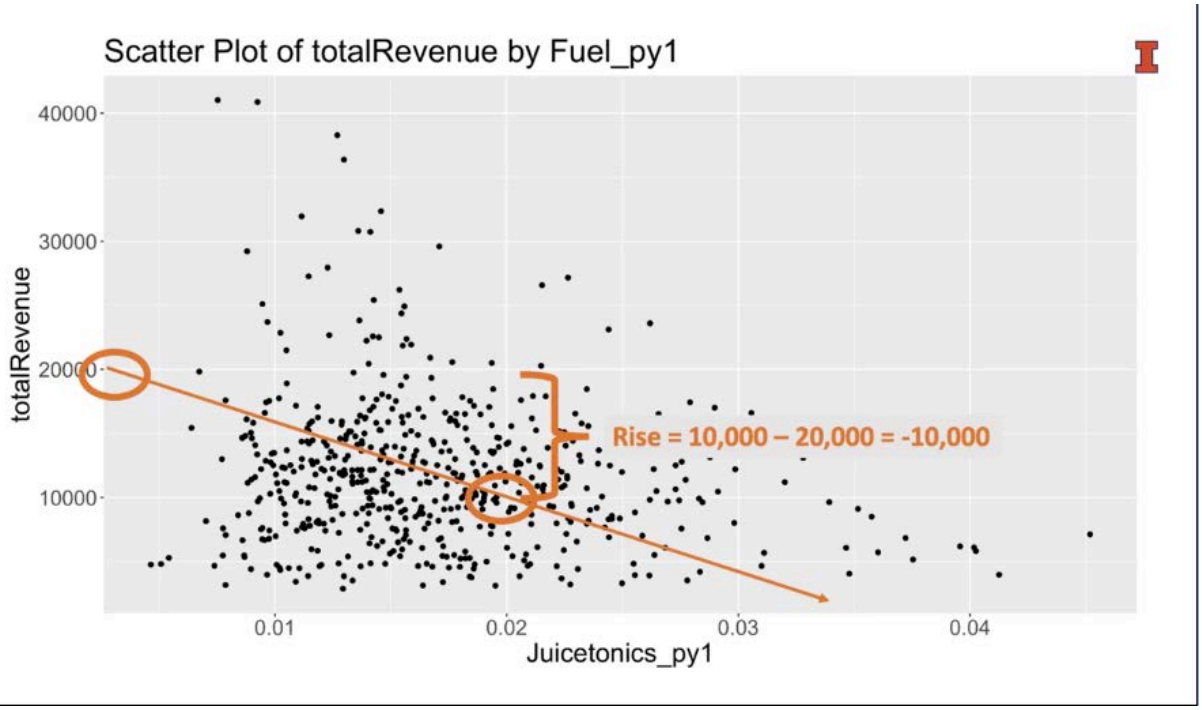
Now, you may well recognize that this linear model is not 100 percent accurate. That's okay. But hopefully, it's an improvement over not having a model. What would you use as a prediction if you didn't have a model?



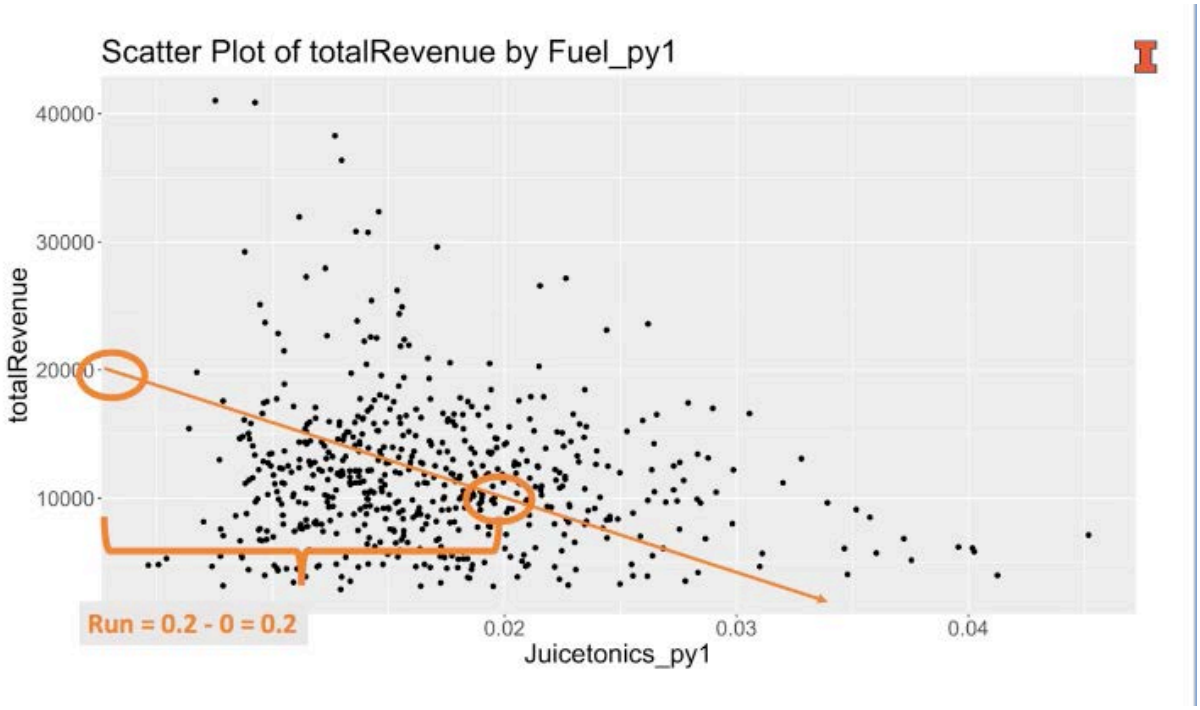
I would have probably gone with the overall average, which happens to be $11,750.55. I think it's easy to see that while that average may be a good starting point, it is really only a good estimate when the percentage of fuel sales from the previous year is between 60 percent and 70 percent. One way to know if the model is an improvement is if we're able to make predictions with more accuracy than if we were just using the overall average. This linear model can also be helpful in understanding relationships and developing strategies for increasing total quarterly sales.



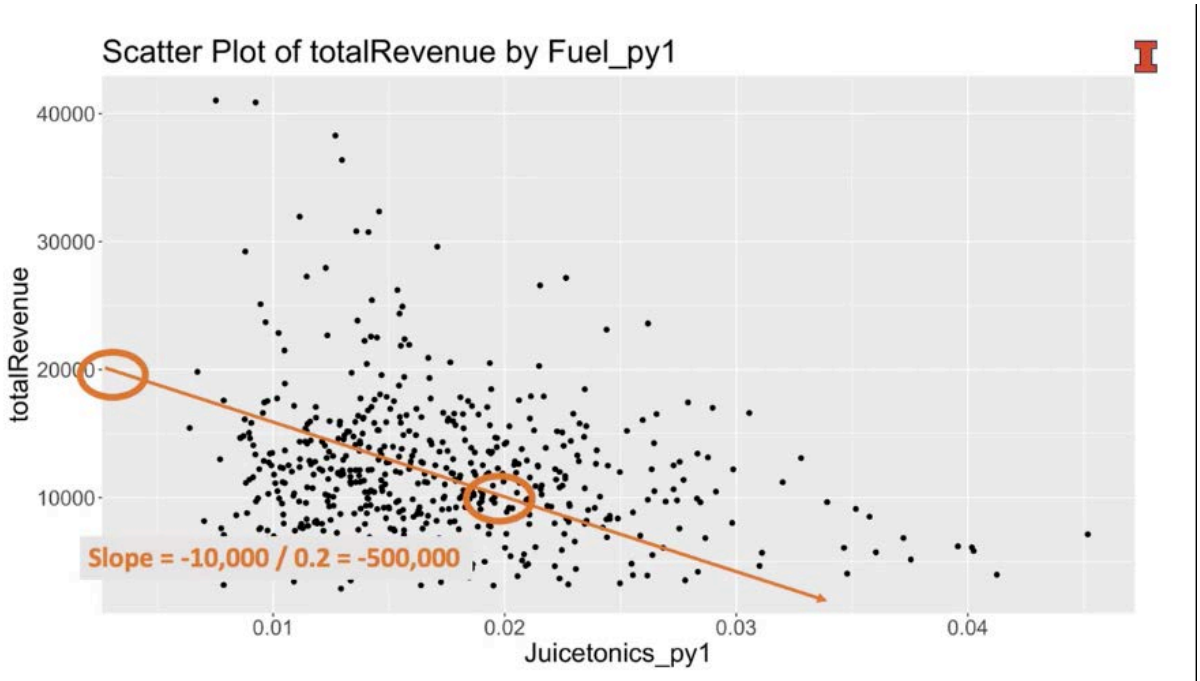
Specifically, we could also say that for every one percent increase in fuel sales during the previous period, we expect our quarterly sales next period to increase by $167. Now let's compare this model to a model that we will create in which we use the percentage of quarterly sales from juice and tonics from the same quarter of the previous year, which is the Juicetonics_py1 variable.



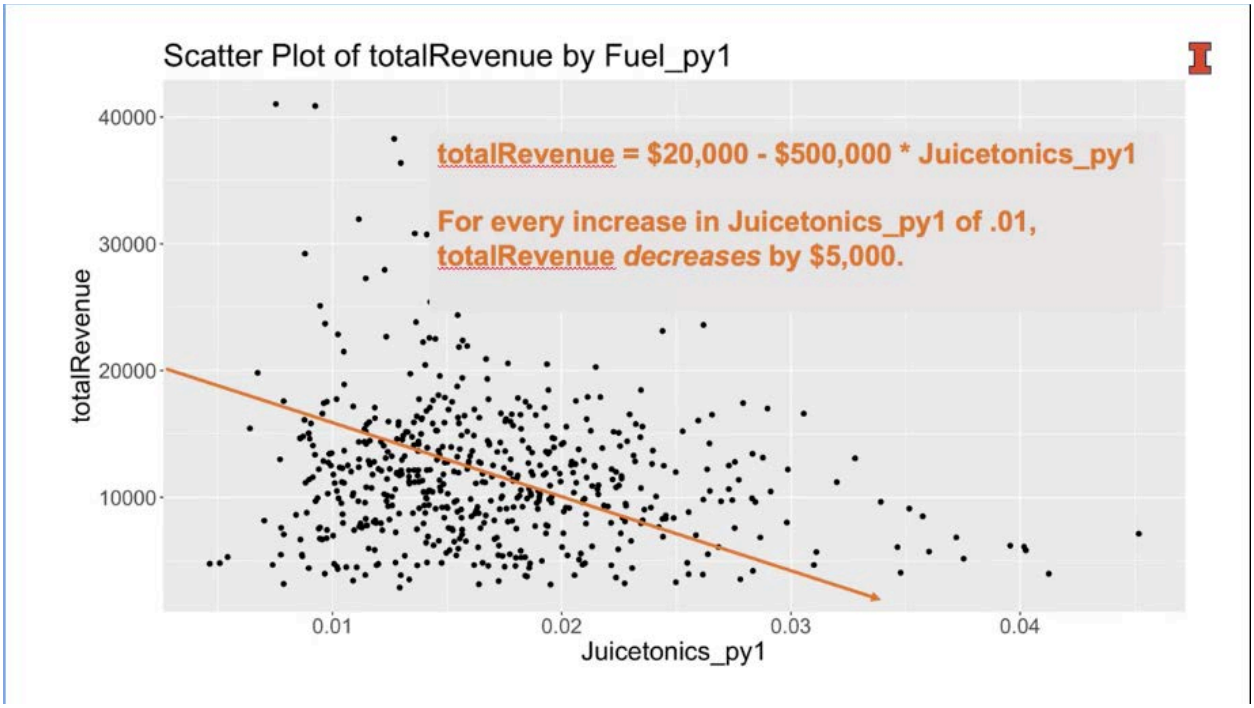
Here's the scatter plot of total revenue in our dataset and Juicetonics_py1. Again, total revenue's on the y-axis because that's what we're trying to predict, and Juicetonics_py1 is on the x-axis because that's what we're using to make predictions. I will visually fit this line in which the intercept is $20,000, and it has a downward slope. We can calculate the slope by using this point on the line at which total revenue equals $10,000, and Juicetonics_py1 equals 0.2. The rise is $10,000 minus $20,000 or negative $10,000.



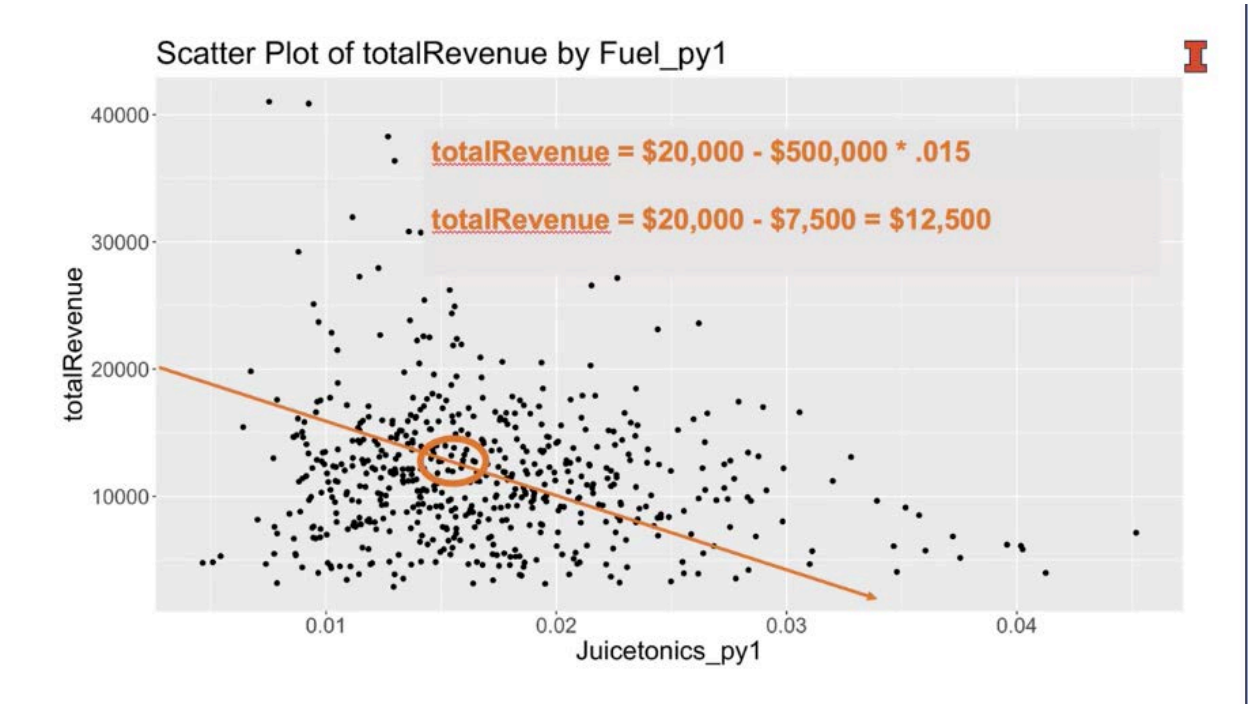
The run is 0.2 minus 0, which is just 0.2.



So the slope is negative $500,000.

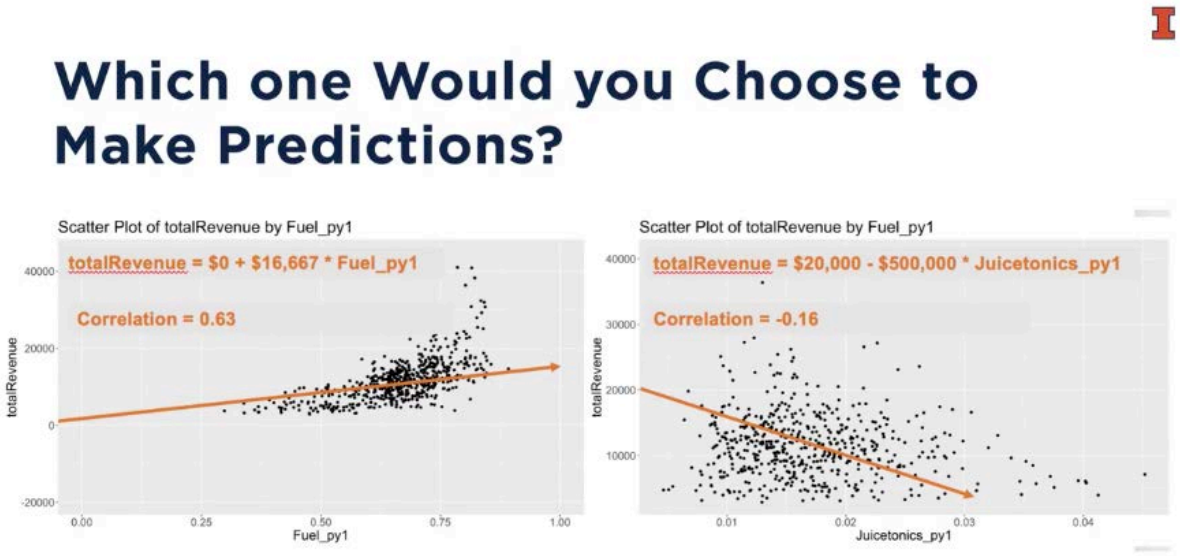


Our model to predict quarterly revenue using juice and tonics is total revenue equals $20,000 minus $500,000 times the Juicetonics_py1 value. As with the model based on fuel, this model is helpful not only for prediction but for explanation. It's nice to be able to quantify the impact on revenue from the amount of sales from juice and tonics.



In other words, for every one percent point increase in quarterly sales from juice and tonics, we would expect our total quarterly revenue in the same quarter of next year to

decrease by $5,000 with respect to prediction. If we knew that the percentage of sales from juice and tonics during the same quarter last year was 1.5 percent, then using this model, we would forecast sales to be $12,500.



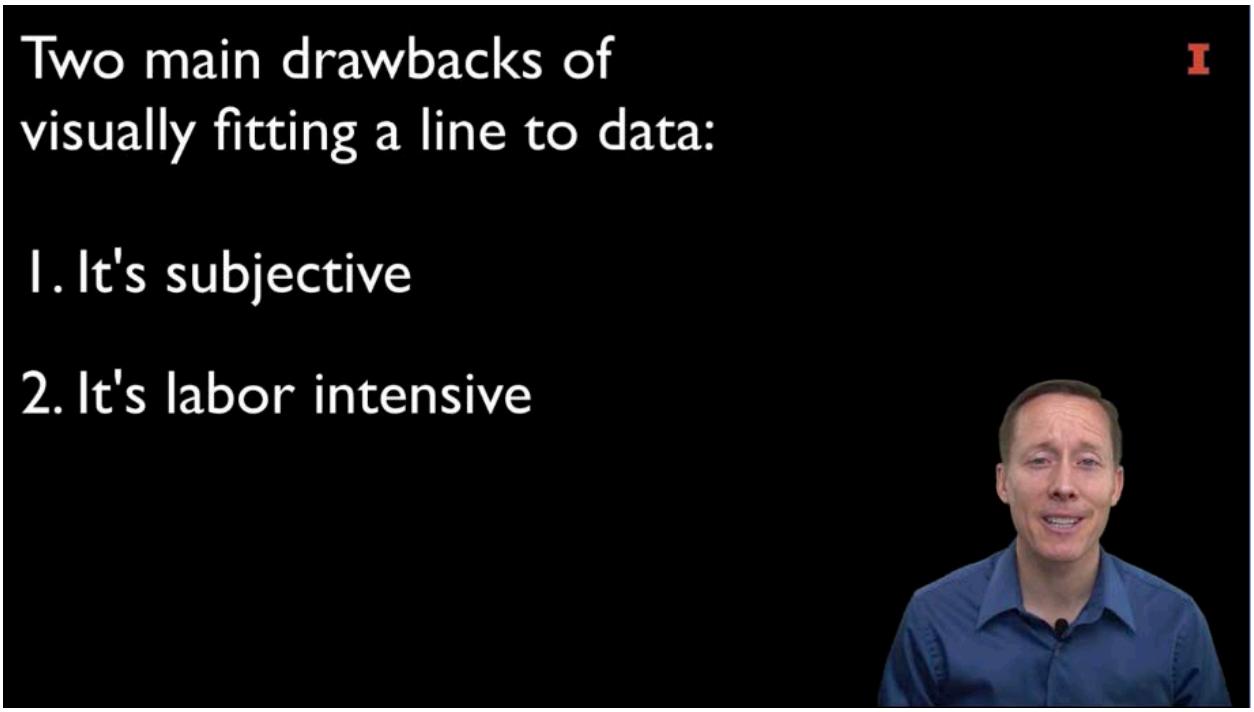
What do you think about the prediction quality of this linear model? Would you trust it more than the average? I probably would. What if you had to choose between using either the previous year's percentage of sales from fuel or juice and tonics to predict quarterly revenue? I would probably choose the model based on fuel because it's clear from looking at the distribution of points around the line when using juice and tonics, that they're even more spread out compared to our model that is based on fuel. One way to quantify my confidence is to refer to the correlation coefficients. Specifically, the correlation coefficient between total revenue and Fuel_py1 is 0.63, and it is only negative 0.16 for total revenue and Juicetonics_py1. Because the absolute value of the correlation coefficient for Fuel_py1 of 0.63 is higher than the absolute value of the correlation coefficient for Juicetonics_py1 of 0.16, I have more confidence in the accuracy of my prediction using Fuel_py1 compared to Juicetonics_py1.

In conclusion, linear models can be really useful to make predictions about the future and to explain relationships. The amount of confidence that we place in the predictions from these models is proportional to the correlation coefficient. Wouldn't it be great if we could use both fuel and juice and tonics to create a prediction? Wouldn't it also be great if we could objectively fit a line to the data points rather than manually do so in a subjective way? Well, we can. But we will save that for another lesson.

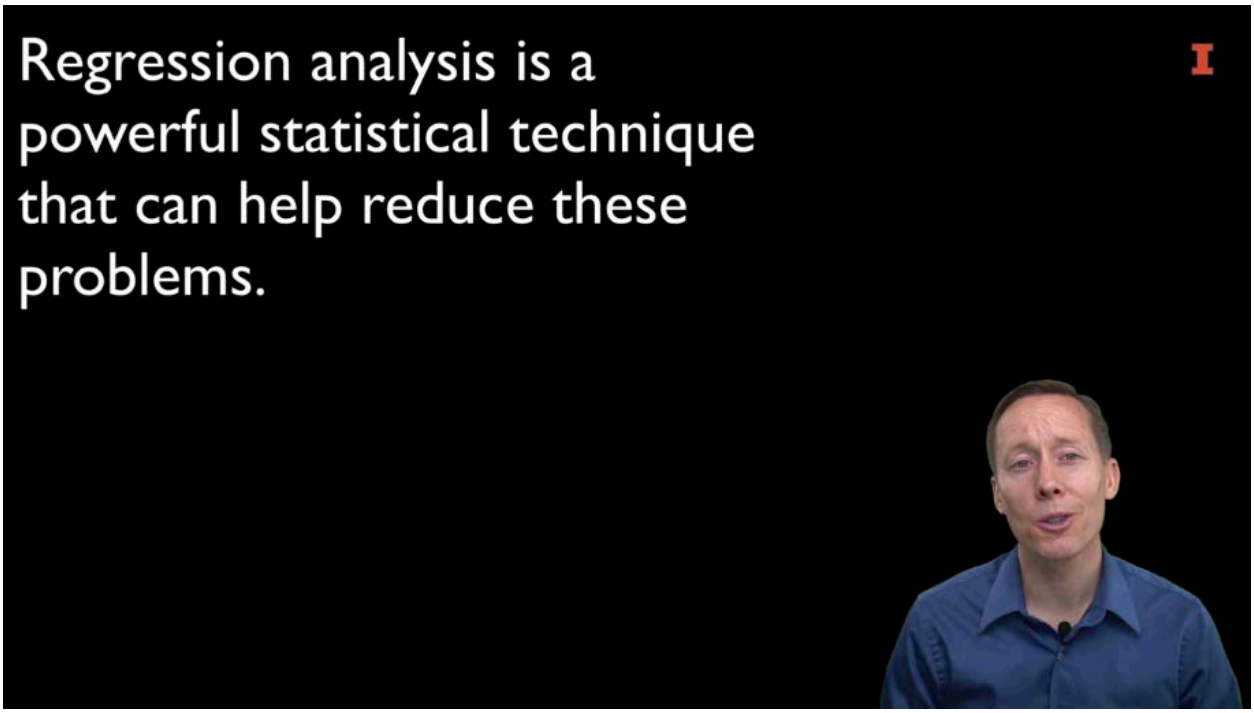

References:
Rattakarn. (2019, September 29). Model Airplane Flight Pattern [Photograph].
https://pixabay.com/photos/model-airplane-flight-pattern-4519750/V

Lesson 1-4.3 Simple Regression

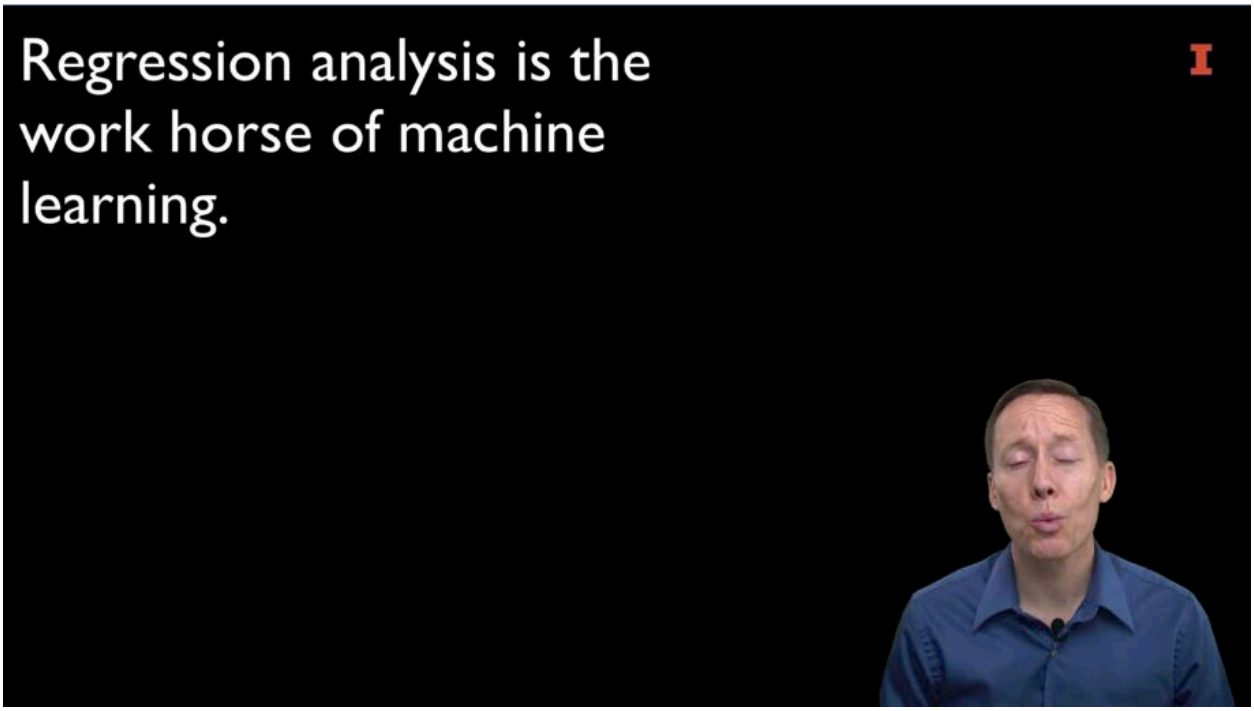
Linear models can be very effective tools for forecasting a business's performance. Visually fitting a line to a scatter plot is effective, but it has two main drawbacks.



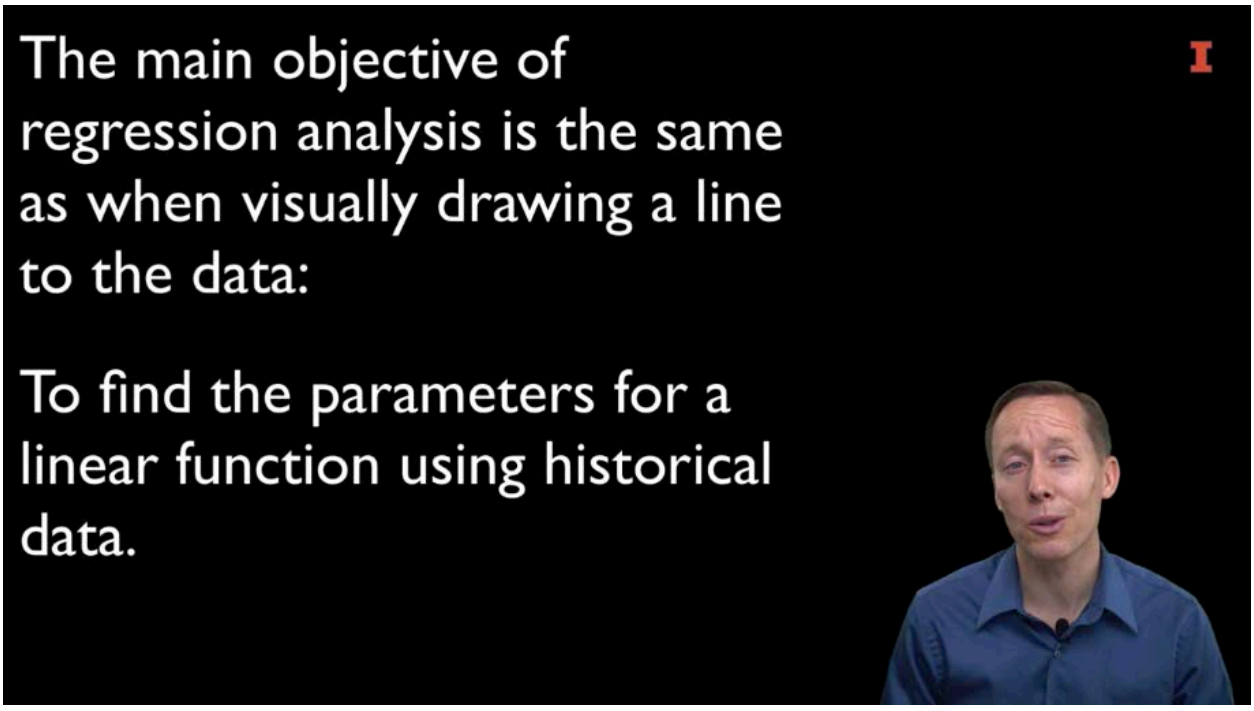
First, it's subjective. The line I draw can be different from the line you draw, and there's not a great way to determine which line is best. The second problem with visually fitting a line is that it's fairly labor-intensive.



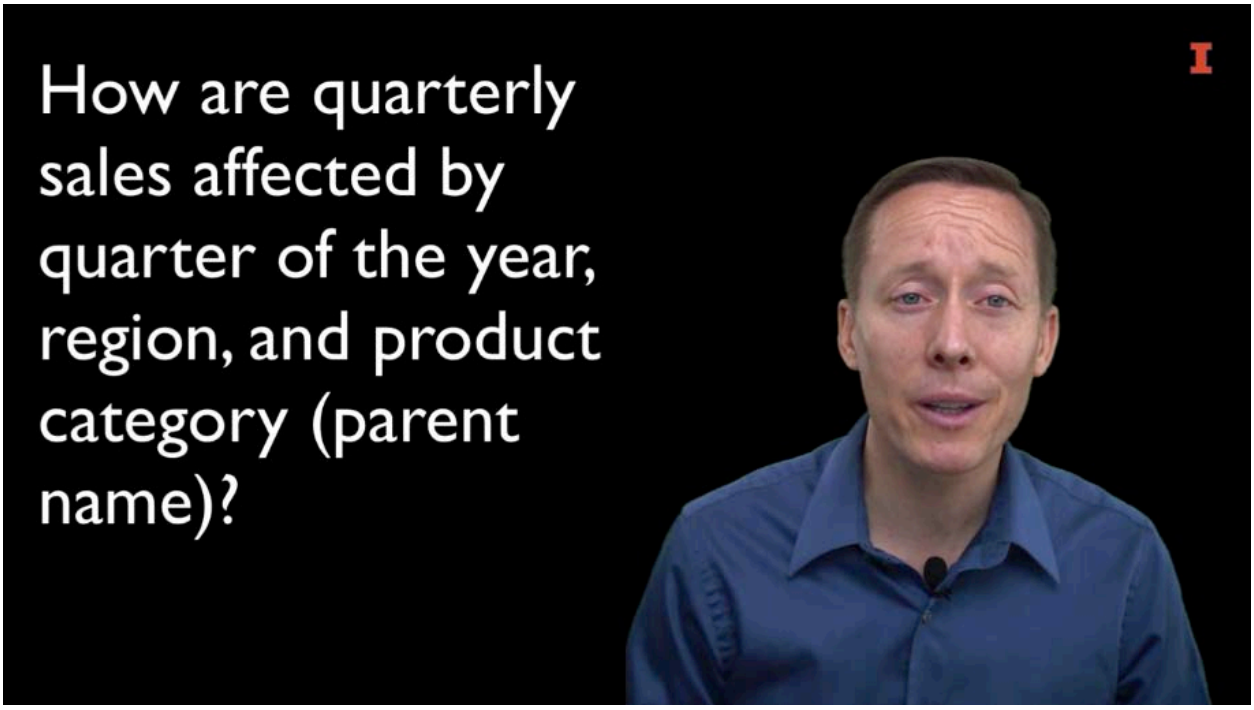
Regression analysis is a powerful statistical technique that can help reduce these problems.



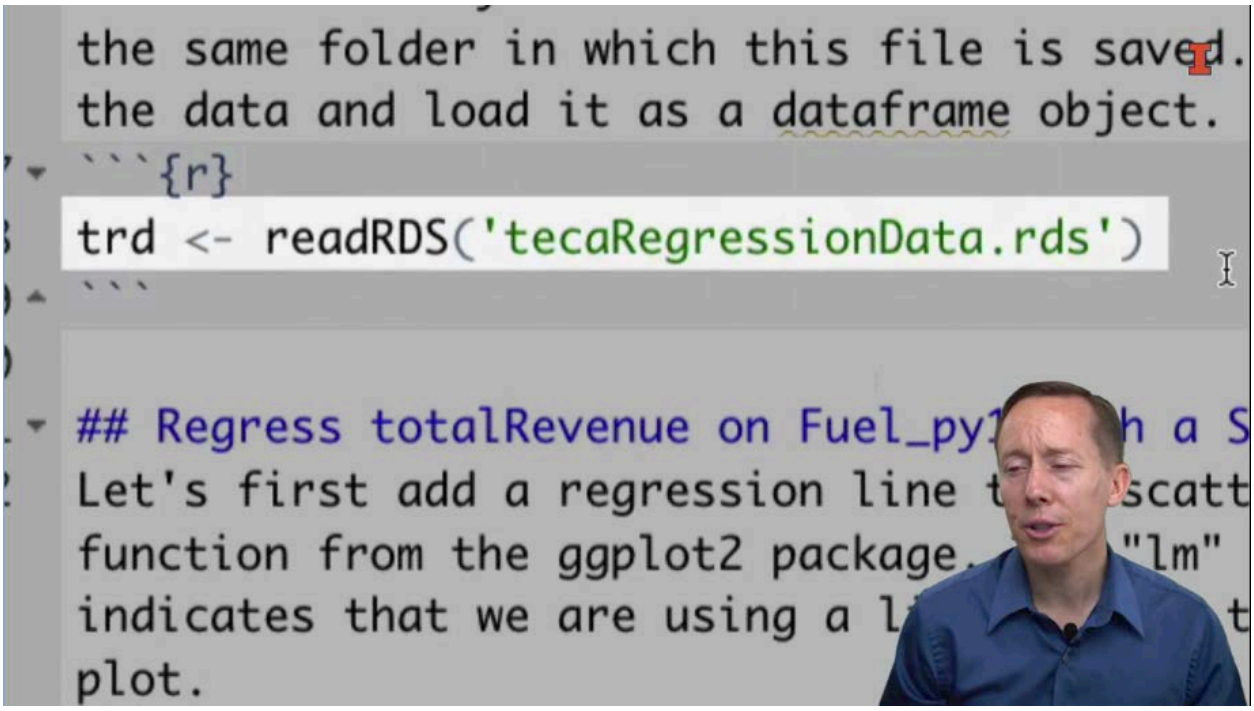
Regression analysis is the workhorse of machine learning.



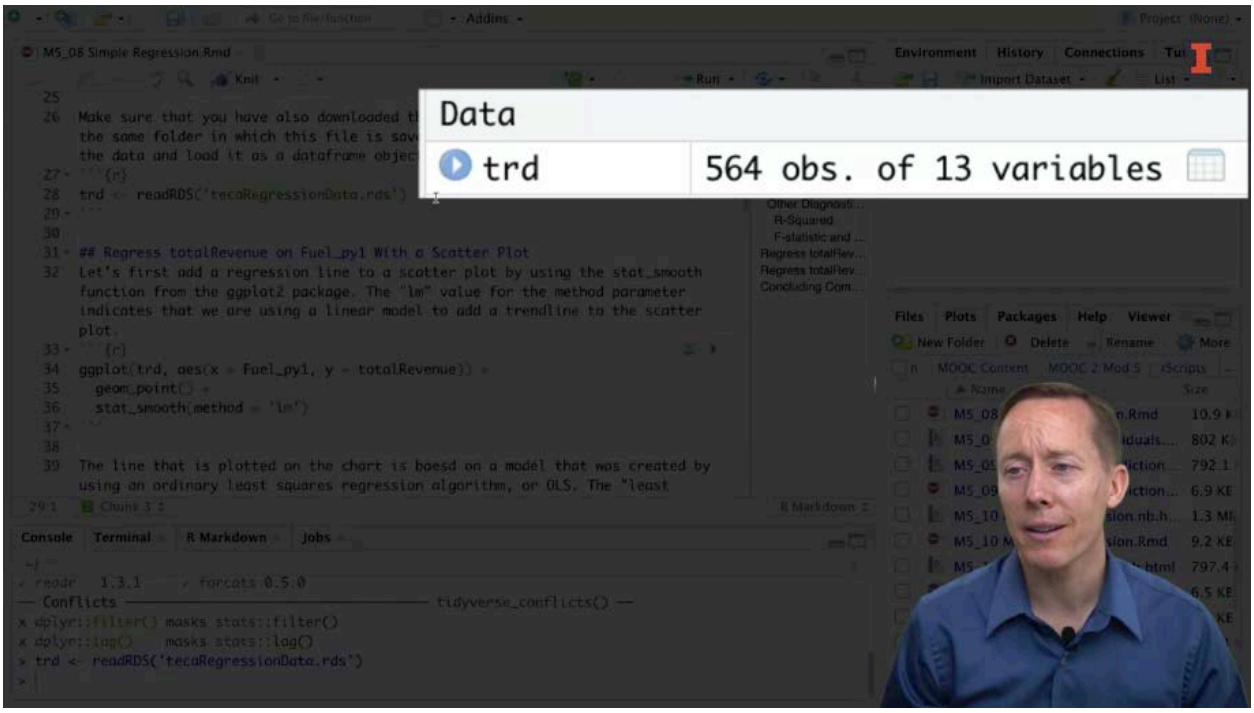
The main objective of regression analysis is the same as our objective when visually drawing a line to the data to find the parameters for a linear function using historical data.



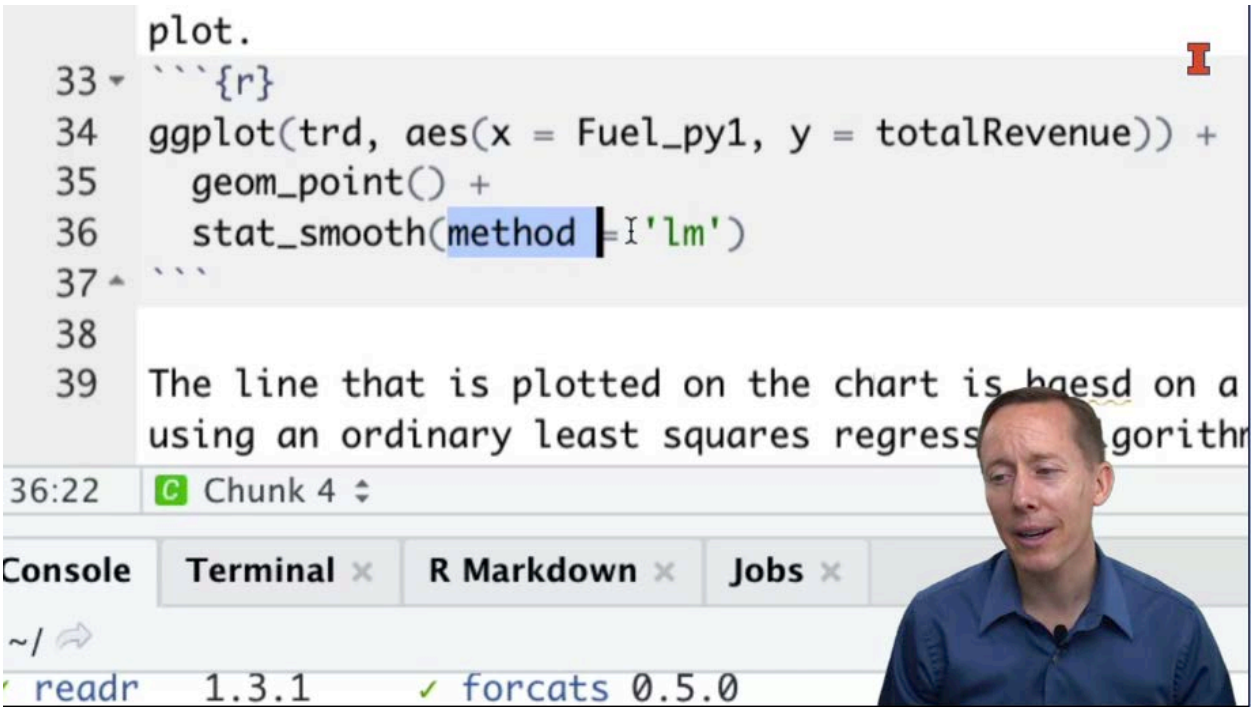
This can help us answer the main question of interests in our setting, which is how are quarterly sales affected by quarter of the year region and by product category?



Let's jump right in and do some regression analyses in R and then we'll have a concrete scenario for discussing the terms and principles. As we explore regression together, I'm going to use the RStudio environment and a Markdown file. If you want to follow along, go ahead and download the associated Markdown file and the TECA Regression Data. Also, make sure that you've installed the Tidyverse collection packages, and if you've done that, then go ahead and load those packages by using the library function. Next, also makes sure that you read in the tecaRegressionData.rds file and save that as a DataFrame.

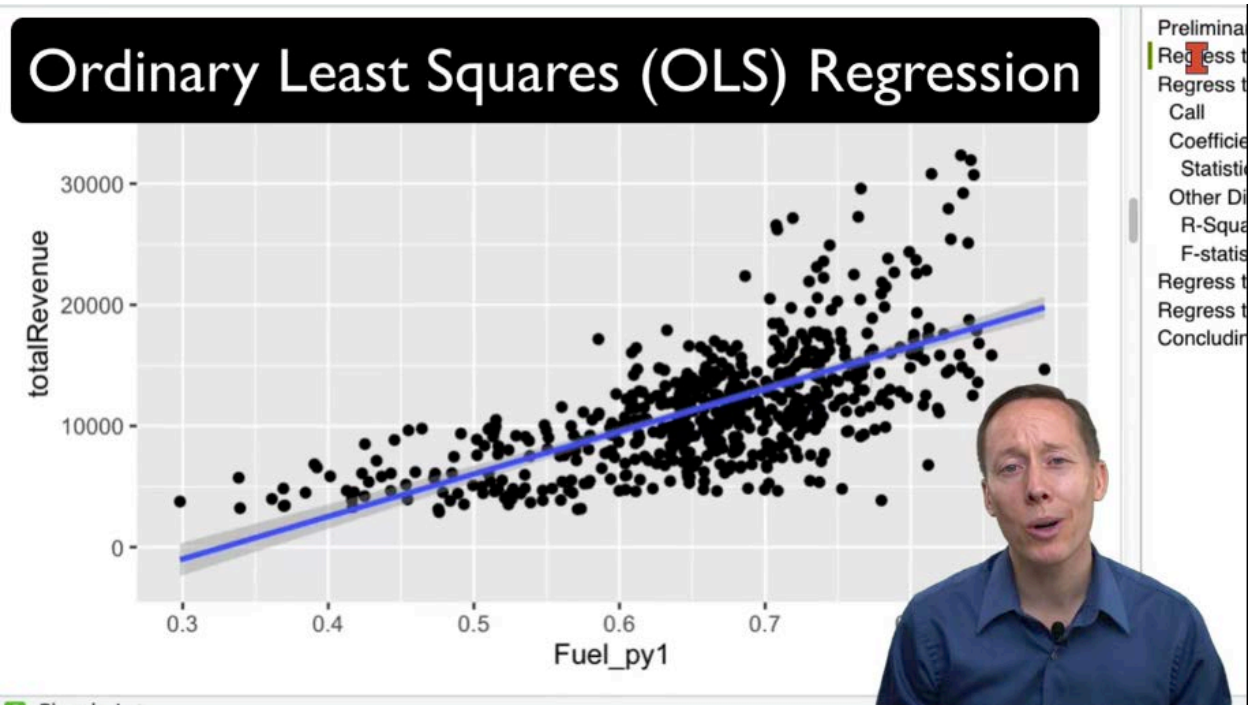


I've saved that as trd, which you can see in the environment pane. Let's first add a regression line to a scatter plot by using the stat_smooth function from the ggplot2 package.

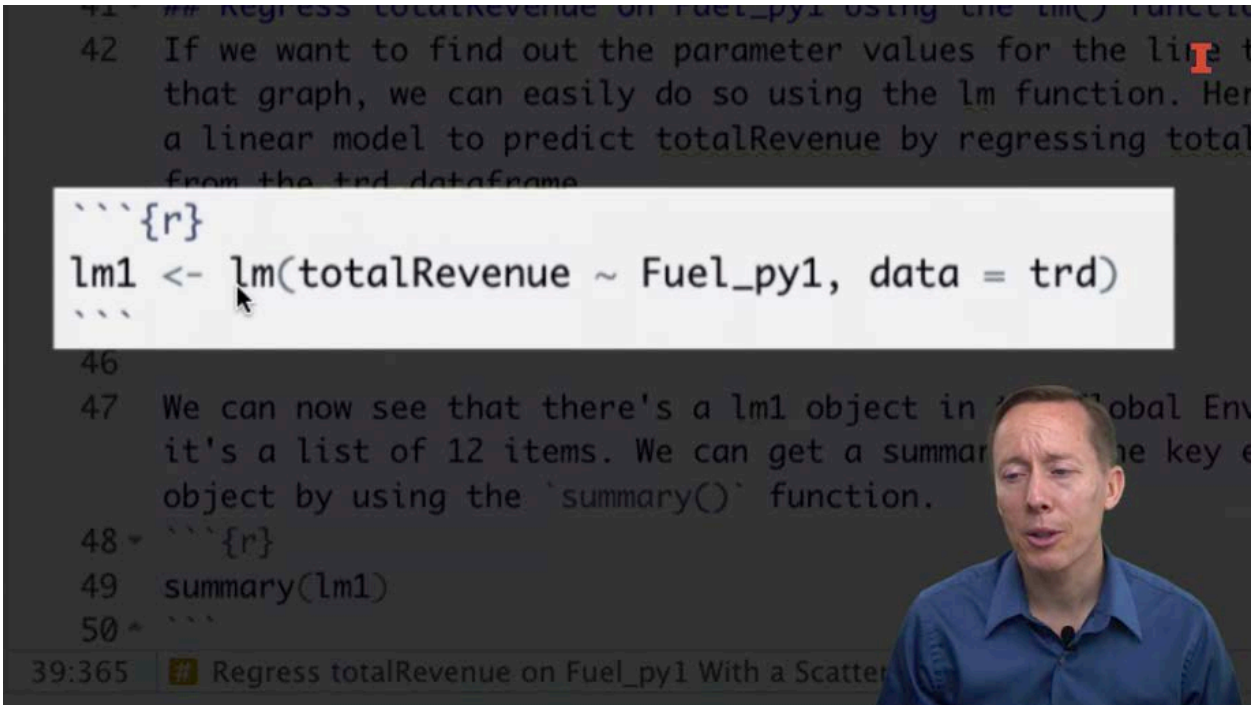


Here's how I'm going to do that. I'm going to use this ggplot function and insert the trd DataFrame. I'm going to set fuel_py1 on the x-axis and total revenue on the y-axis. Then I'll call geom_point to create a scatter plot. Then I'm going to use the stat_smooth

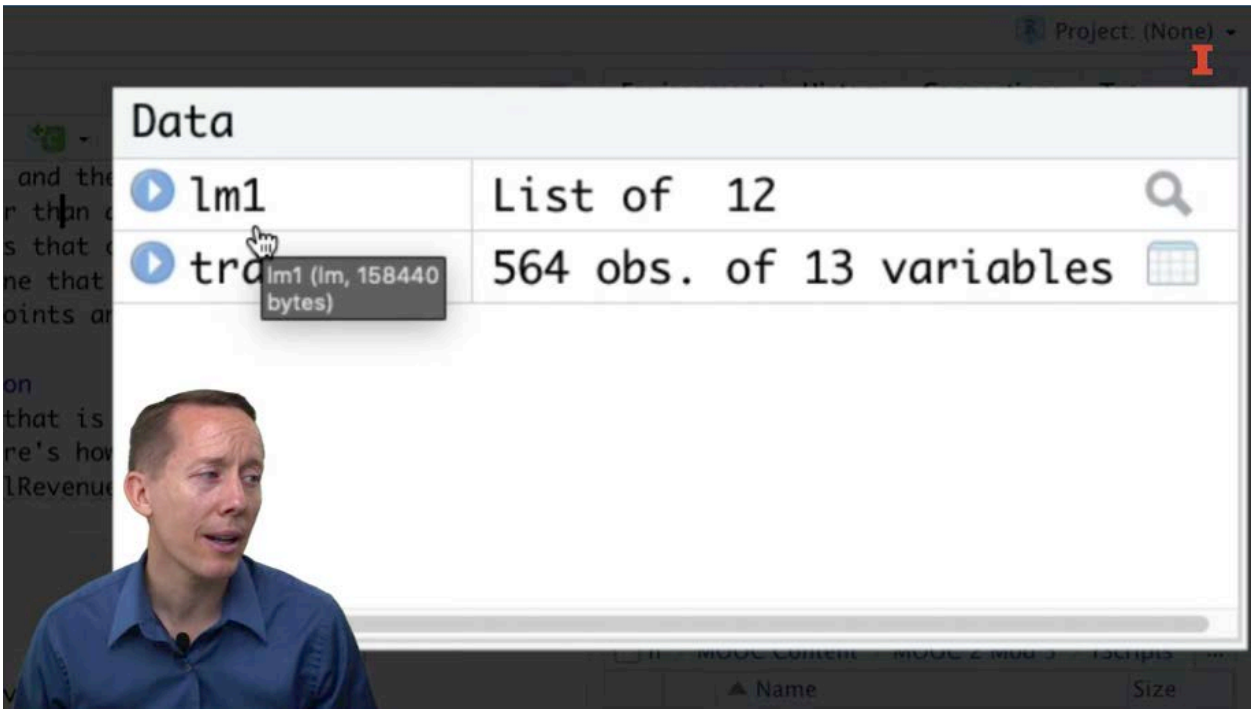
function. This will put a trend line on the plot. The method that I've selected is 'lm' which stands for linear model.



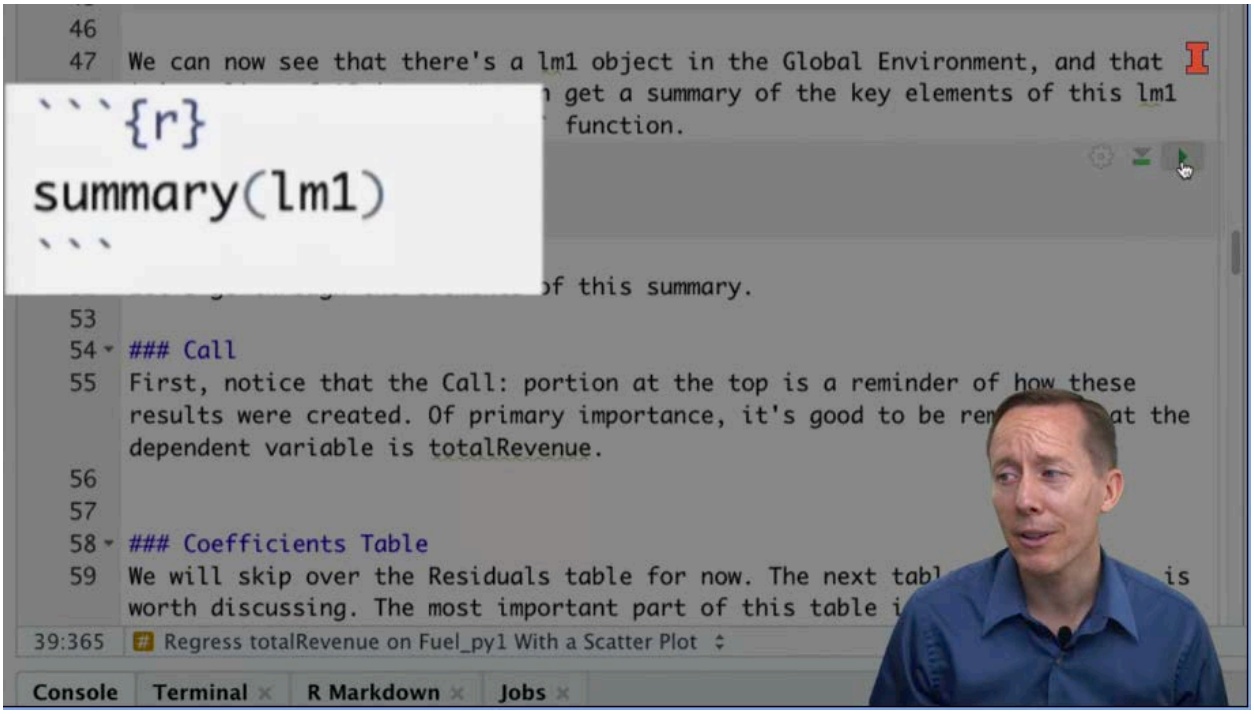
I'll go ahead and run this code chunk. The line that is plotted on this chart is based on a model that was created using an ordinary least squares regression algorithm or OLS. The least-squares part of OLS is just a way to explain that it's creating a line that reduces the distance from each data point and the line. We don't need to go into y squared distances used rather than absolute distance. The important thing to remember at this point is that conceptually, this algorithm calculates the slope and intercept of a line that mathematically reduces the sum of the distance between all of the data points and the line. If we want to find out the parameter values for the line that is plotted on that graph, we can easily do so using the lm function. Here's how we can create a linear model to predict totalRevenue by regressing totalRevenue on fuel_py1 from the trd DataFrame.



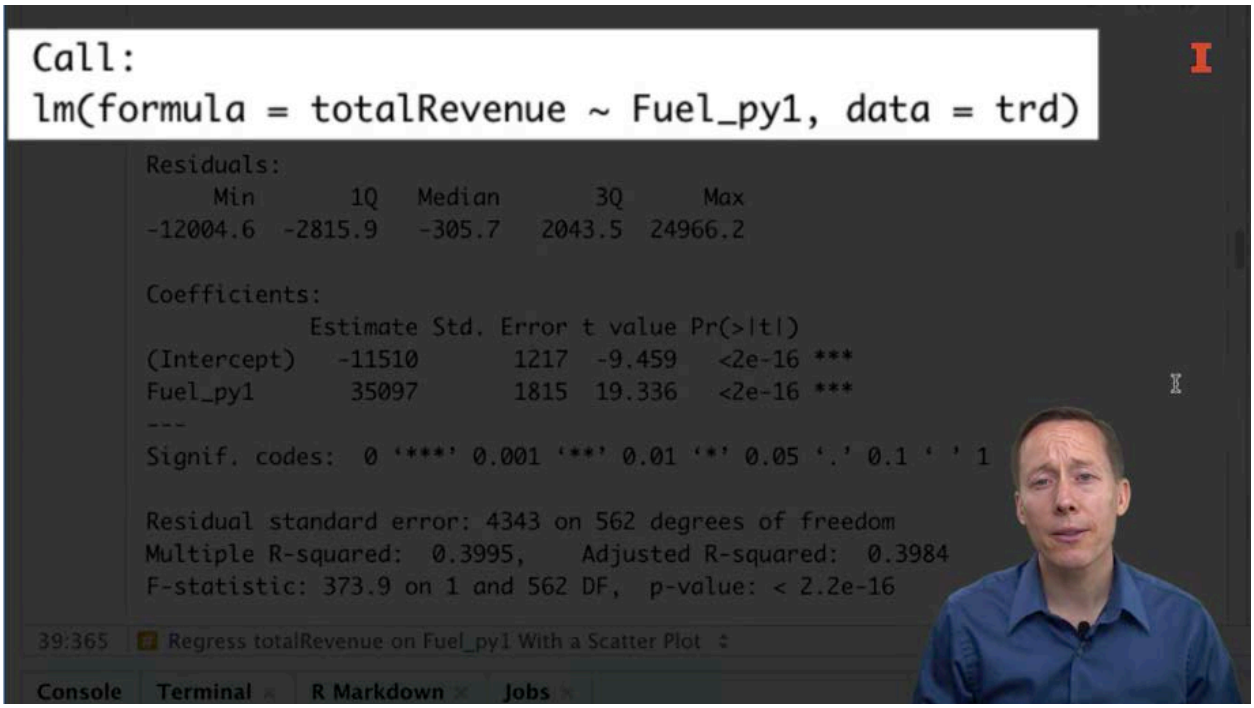
I've got this code chunk here where I'm using the lm function and I've got totalRevenue as the dependent variable. Then instead of saying equals, we use the tilde, which is often above the Tab key. Then we will say fuel_py1 as the variable that we are using to predict totalRevenue, and the data is a trd DataFrame.



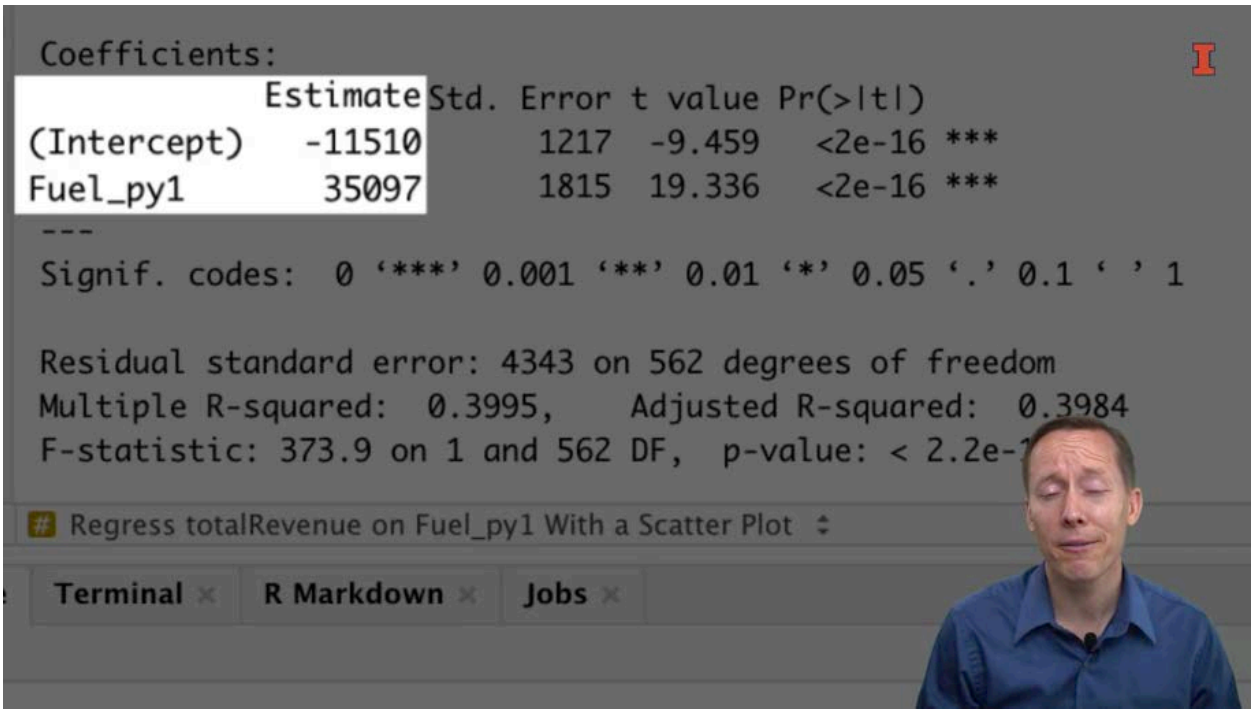
Let's go ahead and run this code chunk. That ran very quickly. You can see we've got this lm1 object in the environment, and if we click on the arrow, we can see that it's a list of 12 different items in there.



Now, we can get a summary of the elements in this lm1 object by using the summary function.

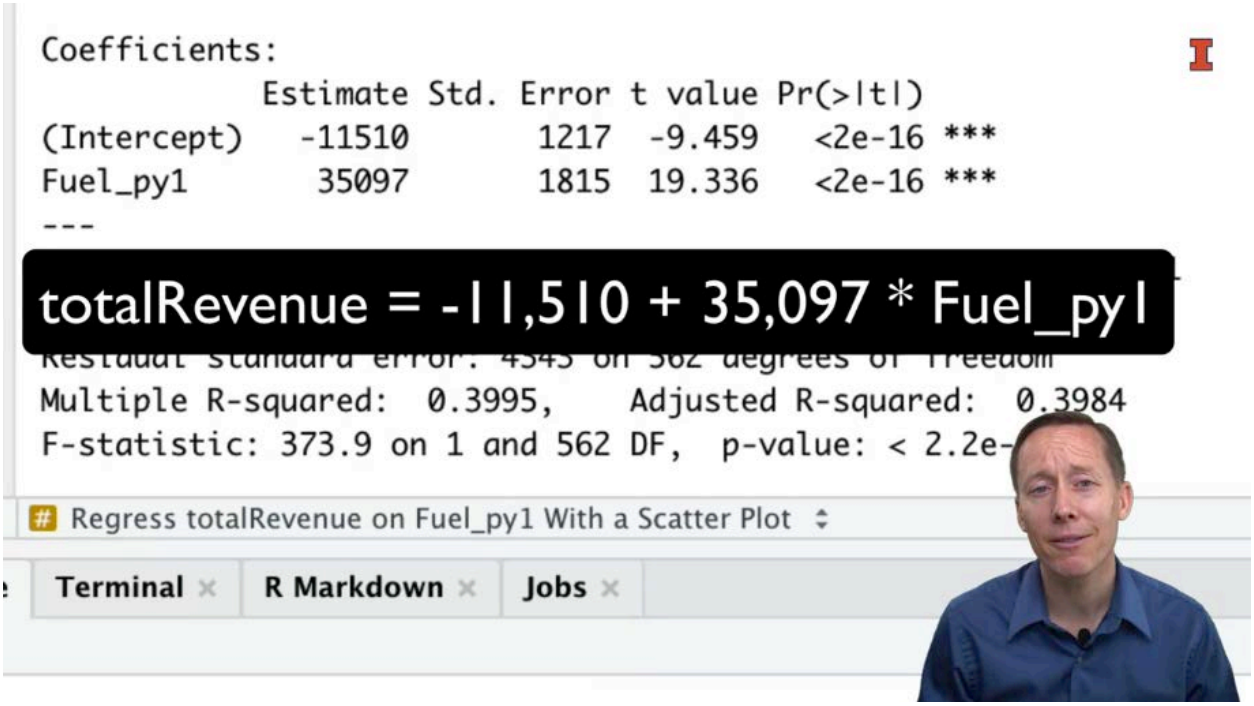


When we do that, you can see that there's quite a few pieces of information in here. Let's go through the elements of the summary. First, notice that the call portion at the top is a reminder of how these results were created, of primary importance, It's good to be reminded that the dependent variable is totalRevenue.

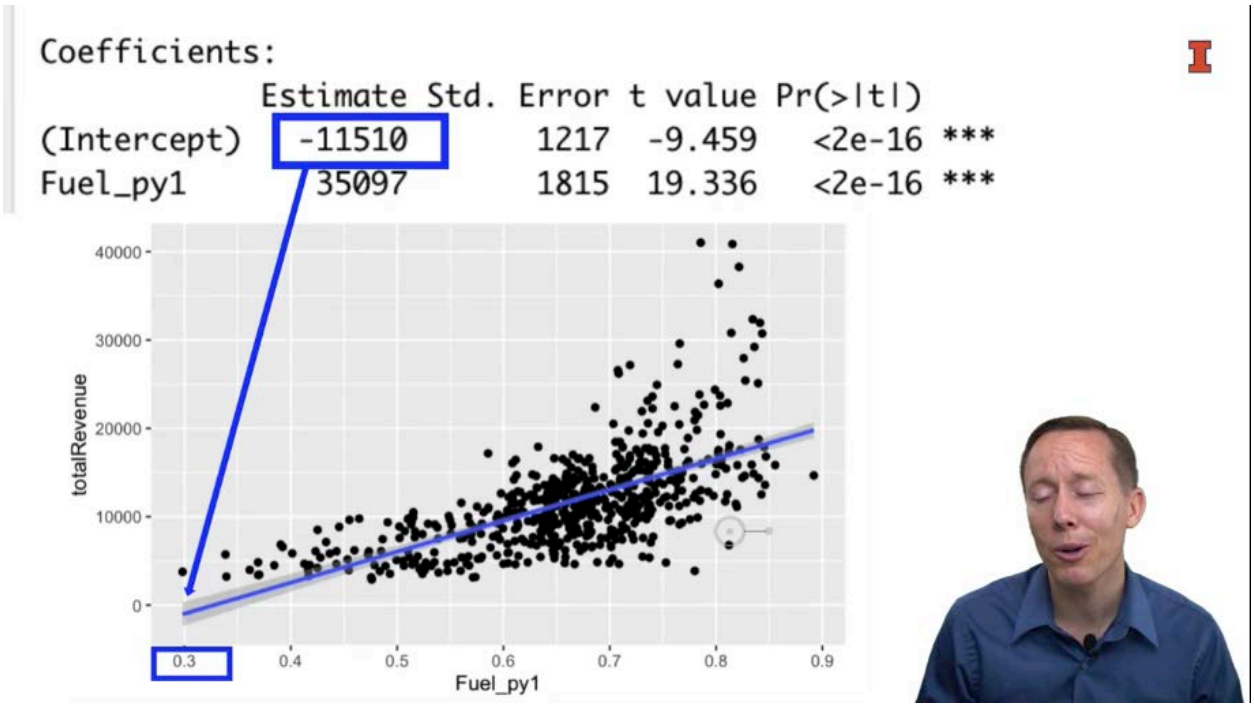


We will skip over the residuals table for now. The next table coefficients is worth discussing at this point. The most important part of this table is the coefficient estimates

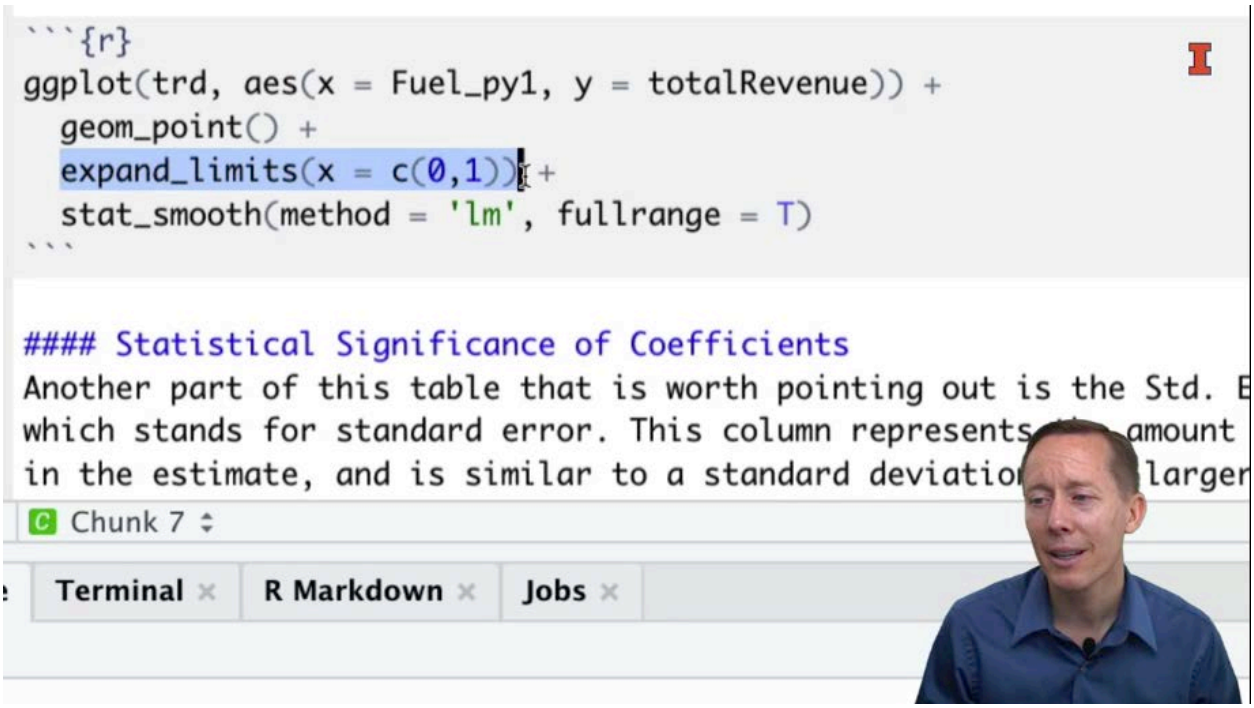
of the intercept and the fuel_py1 variable, which are negative 11,510 and 35,097 respectively, which are somewhat similar to the values we found when we manually fit a line to the scatter plot. Using these parameters from the regression algorithm, we have the following linear model.



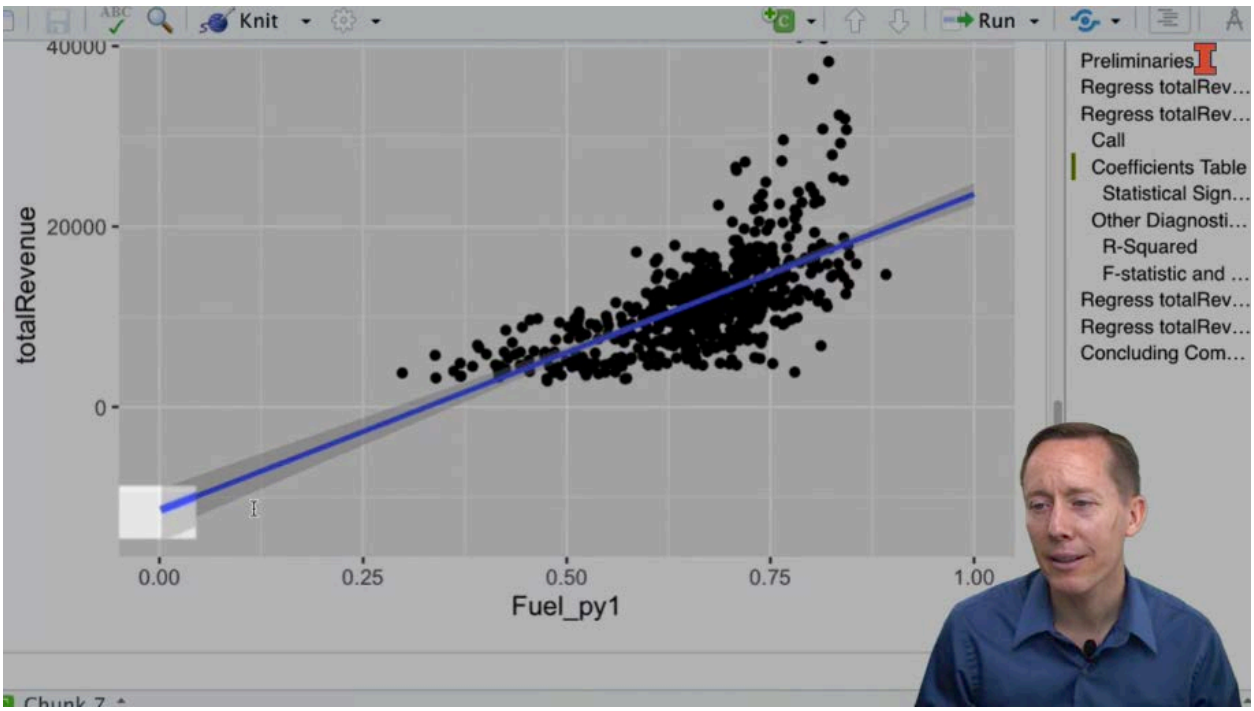
Total revenue is equal to negative 11,150 plus $35,097 times whatever the value of fuel_py1 is.



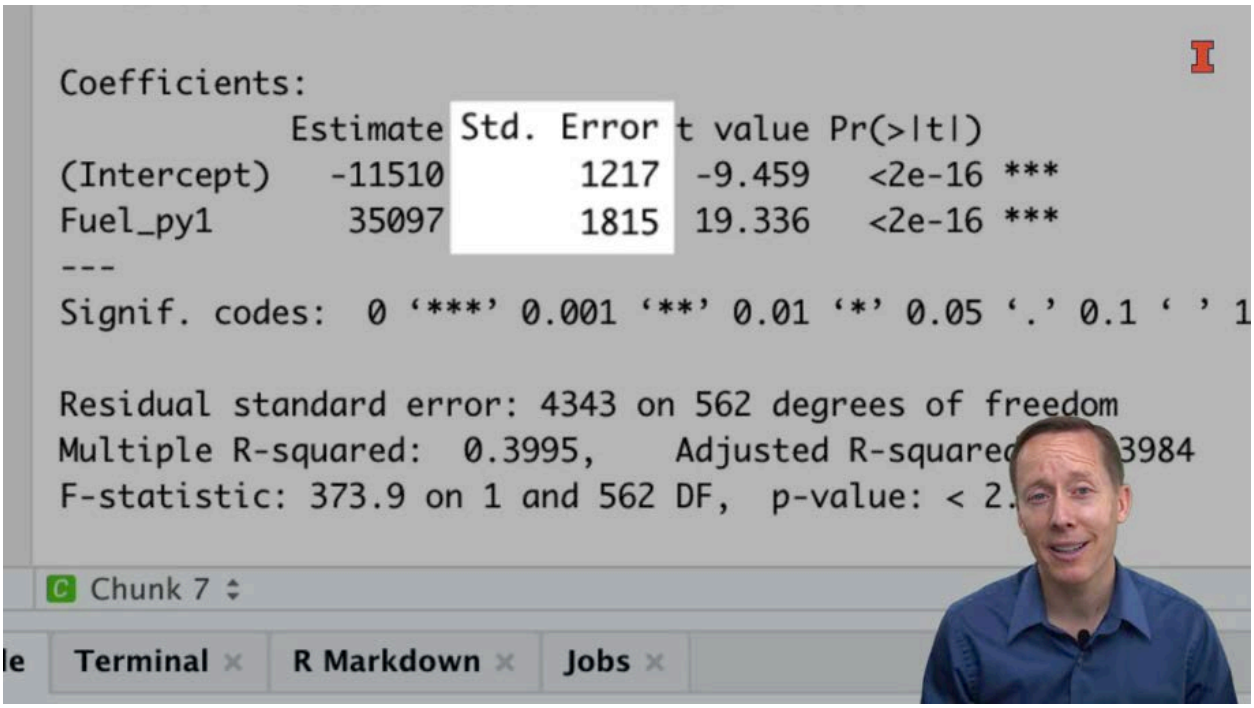
At first glance, the coefficient for the intercept appears to be inconsistent with the line on the scatter plot. But that's because the x axis starts at 0.3 rather than at 0. If we expand the limits of the x axis and the range of the linear models so that it goes to 0, then we can see that the line crosses the y axis at the point that corresponds to the intercept coefficient of negative 11,510.

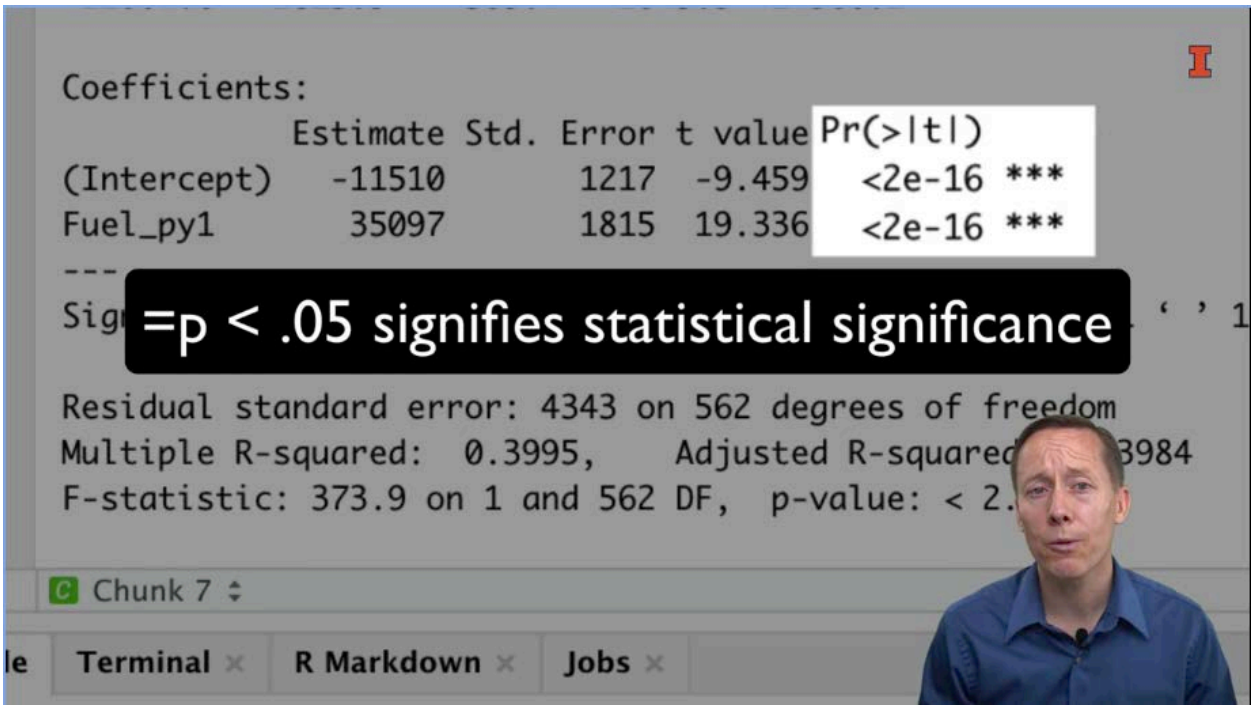


Here's how we can do that. We will use this expand limits function in this ggplot function.



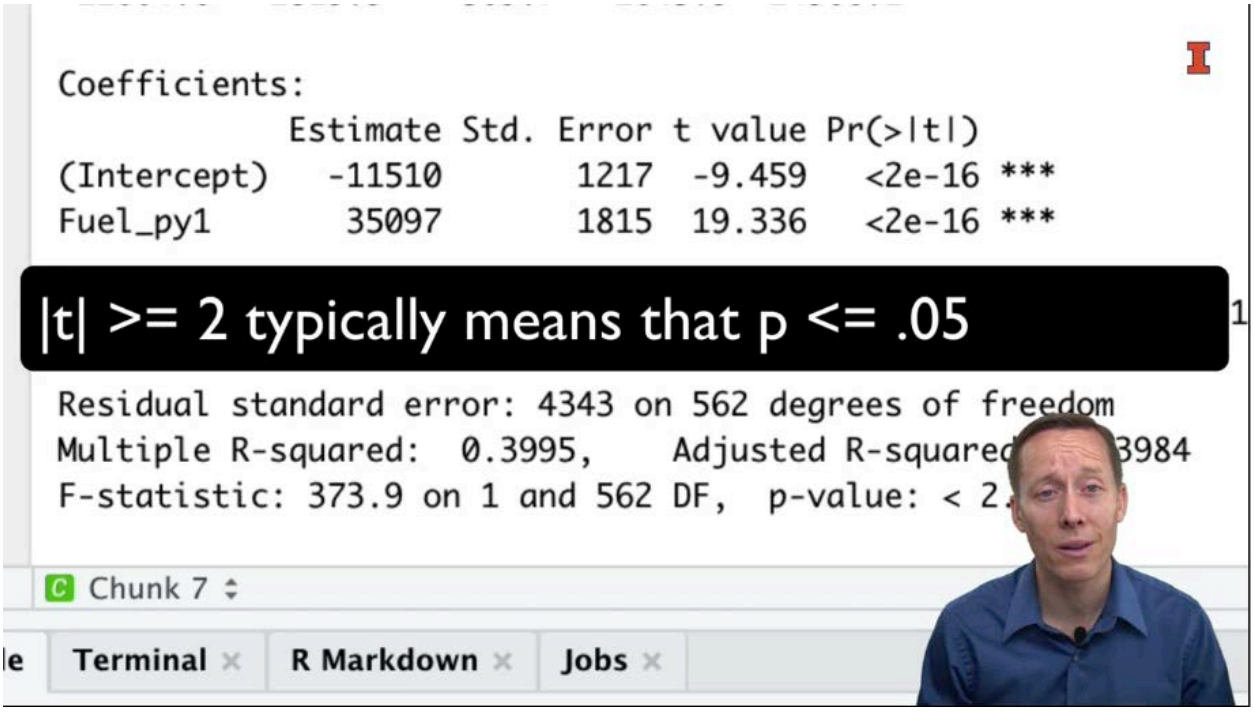
We'll also say fullrange is equal to true and run that. Now we can see that that line crosses the y-axis at approximately negative 11,000.



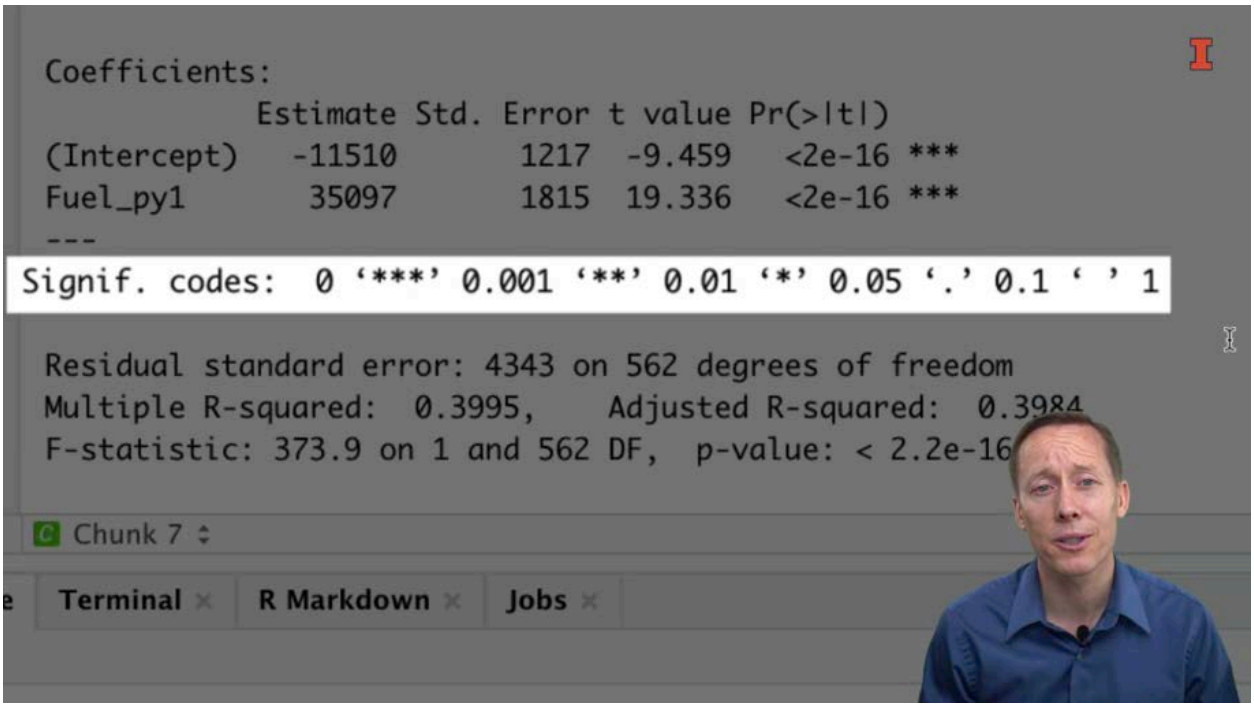


Another part of this table that is worth pointing out is the standard error column. This column represents the amount of variation in the estimate, and it's similar to a standard deviation. The larger the standard error, the less certain we are in the estimate. If you divide the coefficient estimate by the standard error, you get the t value, which is in the third column. This corresponds to a distance from the center of a t distribution. The t value is used to calculate the value in the last column, that last column, the probability that it's greater than the absolute value of t is also known simply as the p value. This

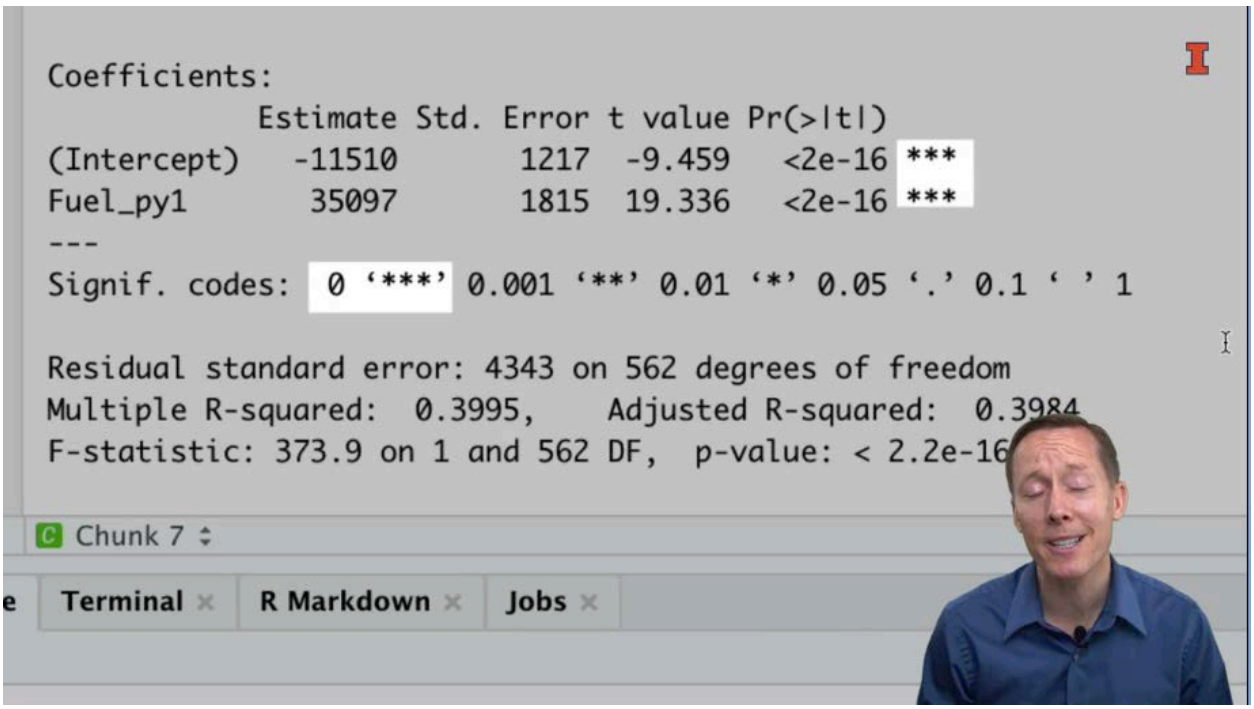
value is what tells us if we can be confident that the coefficient estimate is different from 0. If the p-value is less than 0.05, then typically we conclude that the coefficient estimates are statistically significant, meaning that we are confident that they are not due to chance.



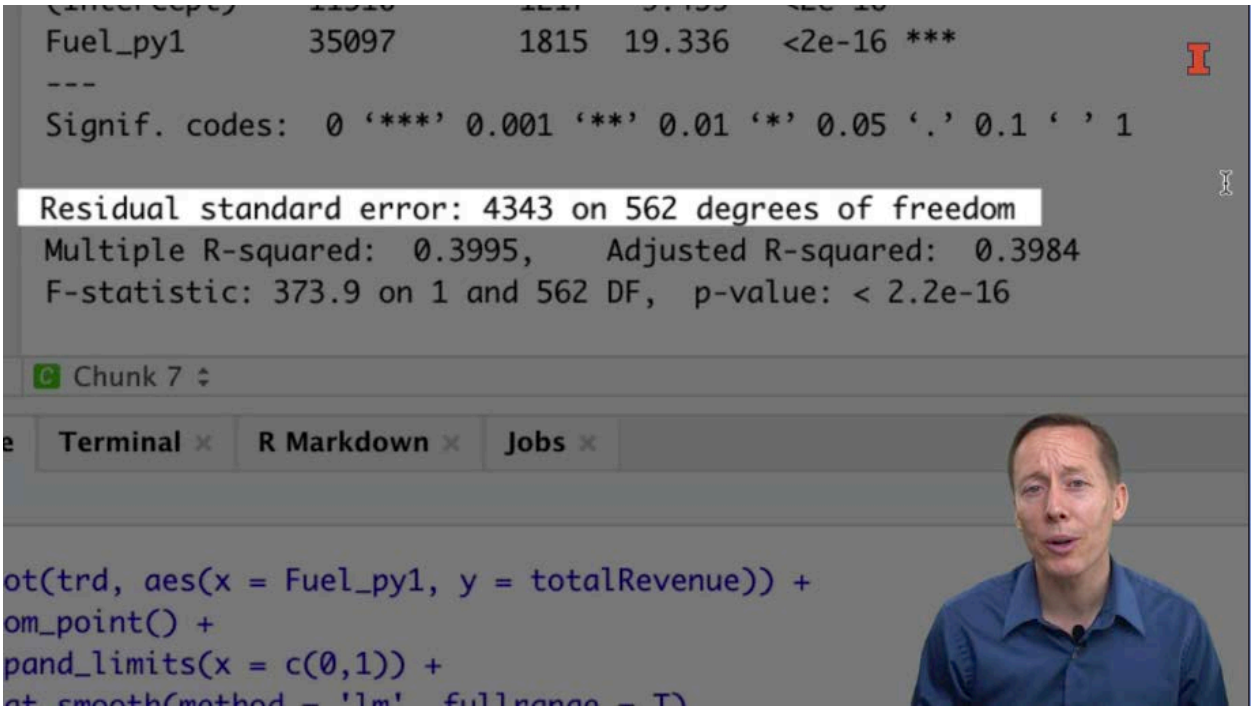
T values that have an absolute value of 2 typically results in a p-value that is 0.05 or less. The statistical significance is important when making inferences and getting understanding of relationships.



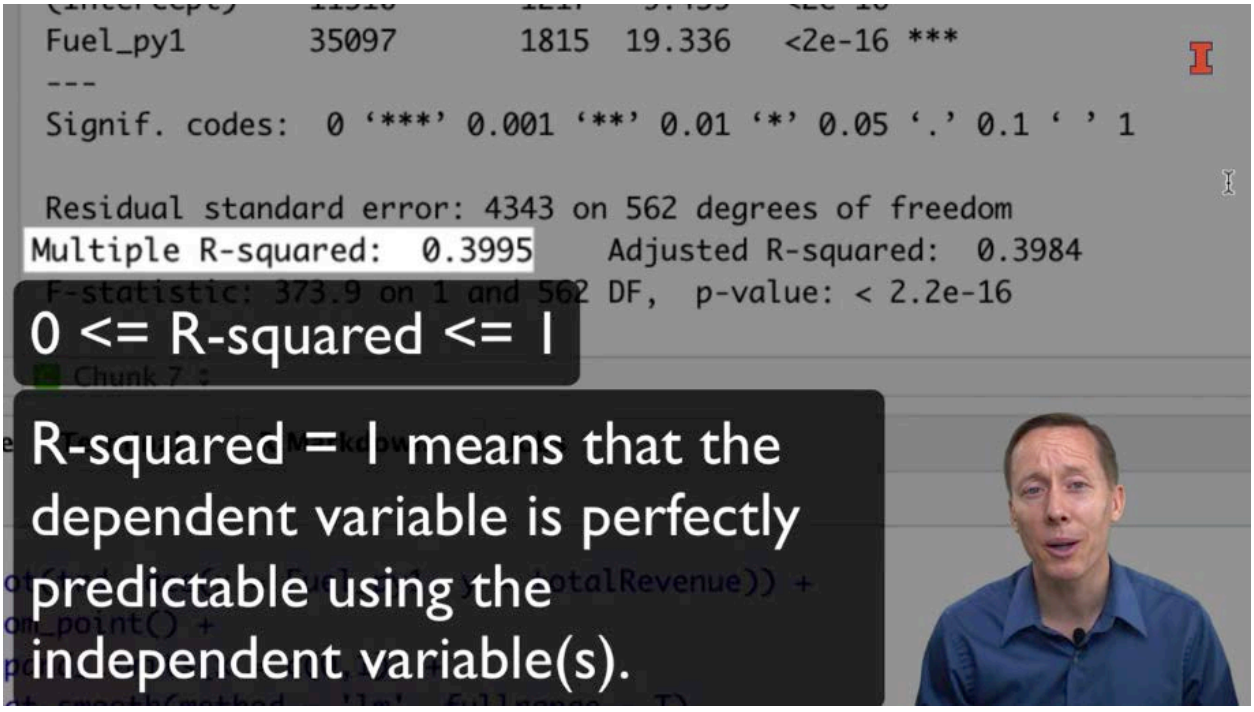
You can see that the table below uses a series of asterisks and periods to quickly denote the level of statistical significance.



In the case of this regression model, we can see that both coefficients for the intercept and the fuel_py1 variable, have p-values that are very small and are therefore statistically significant.

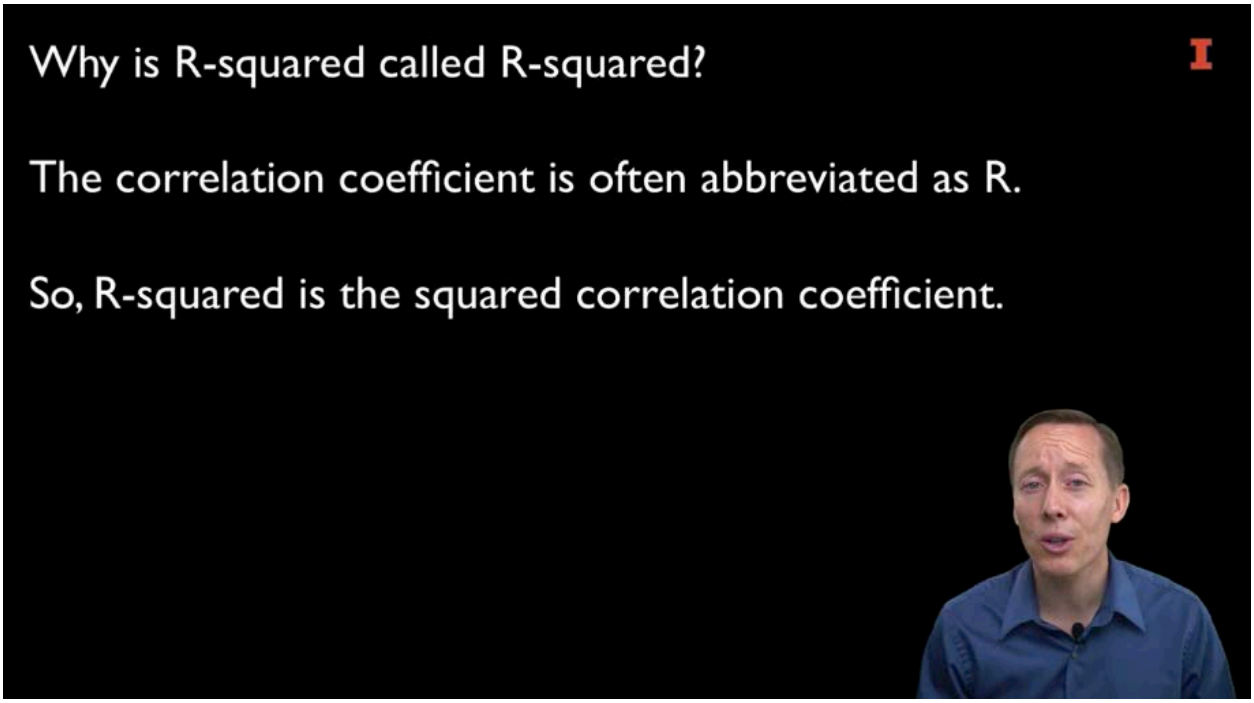


The information under the bottom of the table is useful for prediction purposes. The residual standard error is similar to the average deviation between the actual data points and the line.

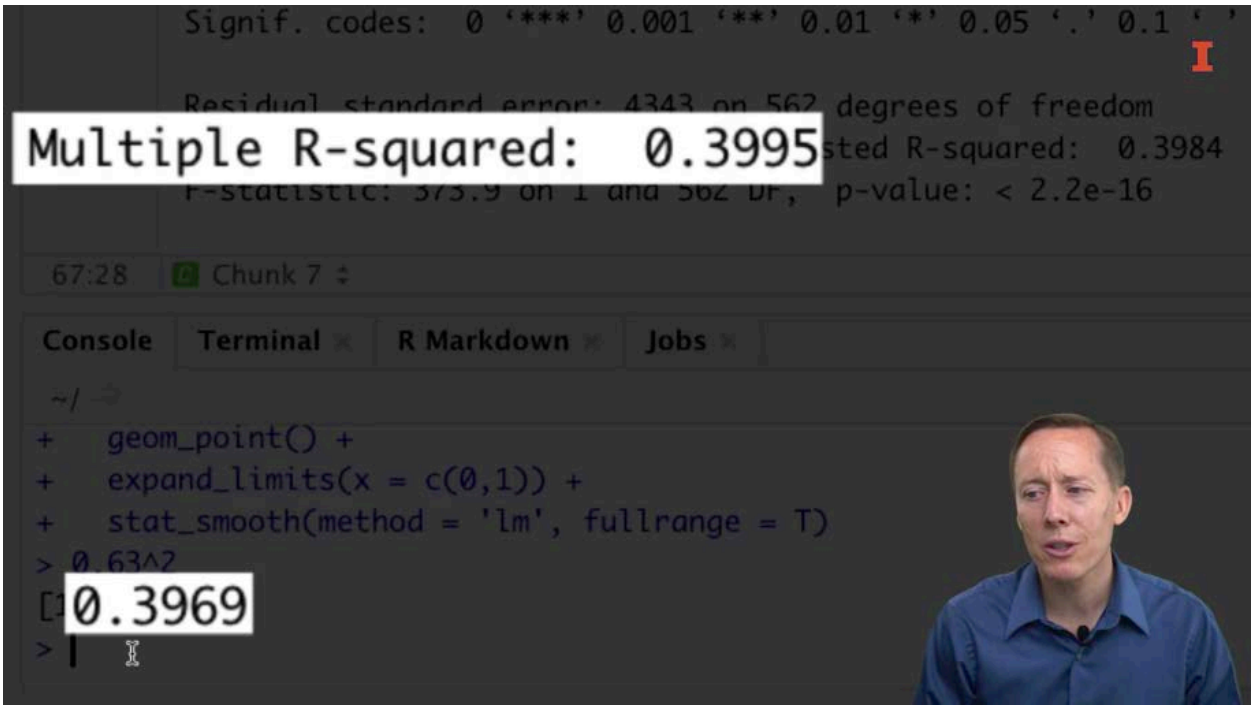


The multiple R-squared, often referred to simply as R-squared, tells us how much variation in the dependent variable is explained by the independent variable. In our case, 39.95 percent of the variation in total revenue is explained by fuel_py1.The value

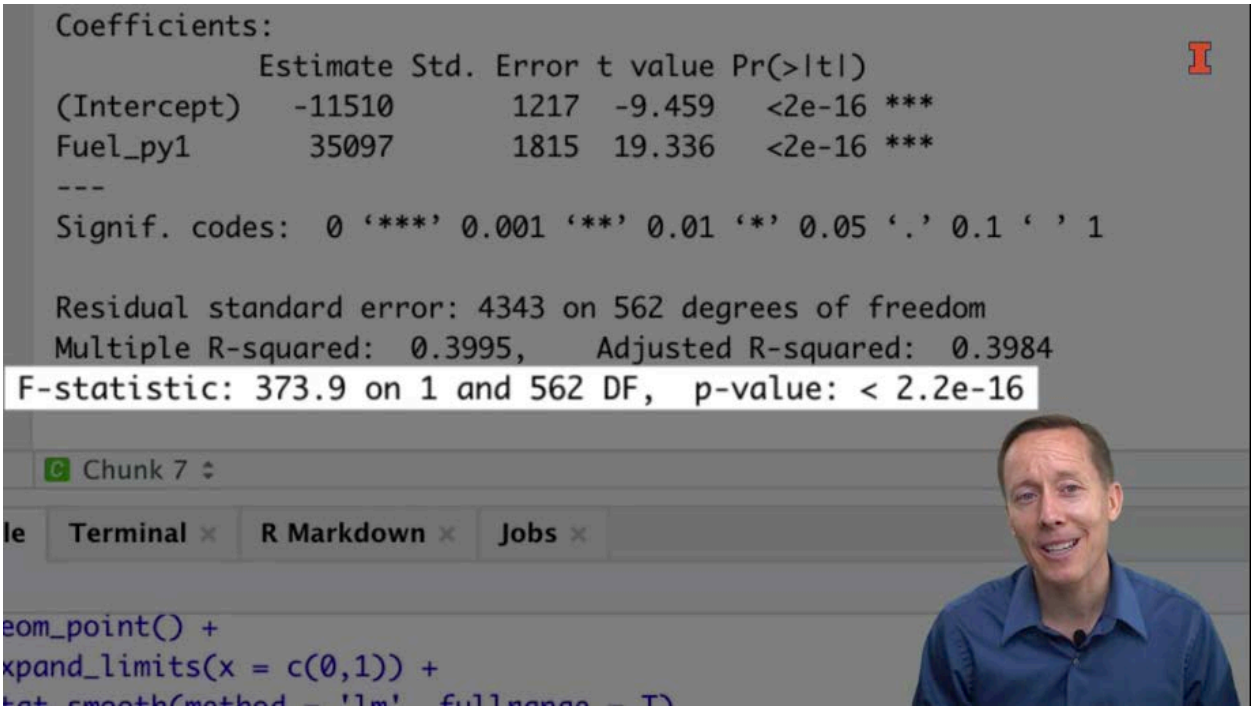
of multiple R-squared can range from 0-1. If the value is 1, then the value of total revenue is perfectly predictable by fuel_py1, and all of the dots would fall on the regression line. For prediction purposes, the confidence that we place in our predictions correspond to the R-squared.



Now, why is it called R-squared? Well, correlation is often abbreviated by the letter R. R-squared is simply the squared correlation coefficient between total revenue and fuel_py1.

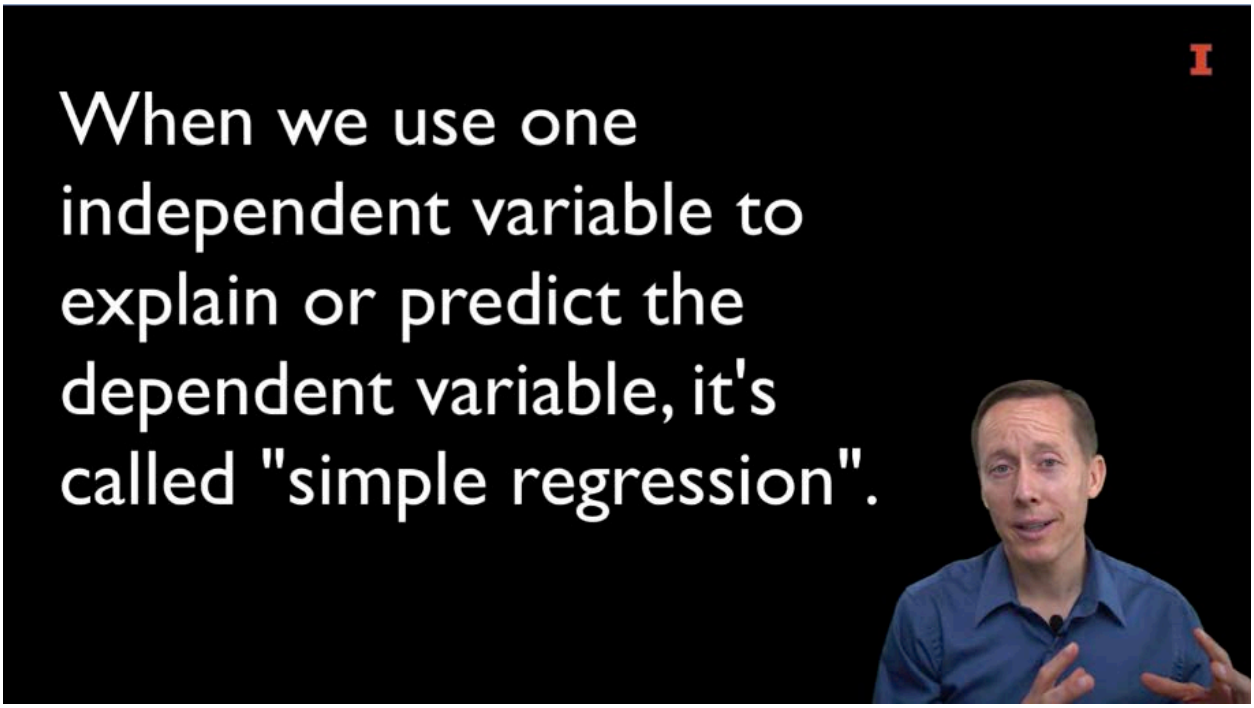


We can verify this by taking the correlation coefficient of 0.63 and raising it to the second power. We get 0.3969, which is approximately our R-squared value. The small difference is due to rounding.

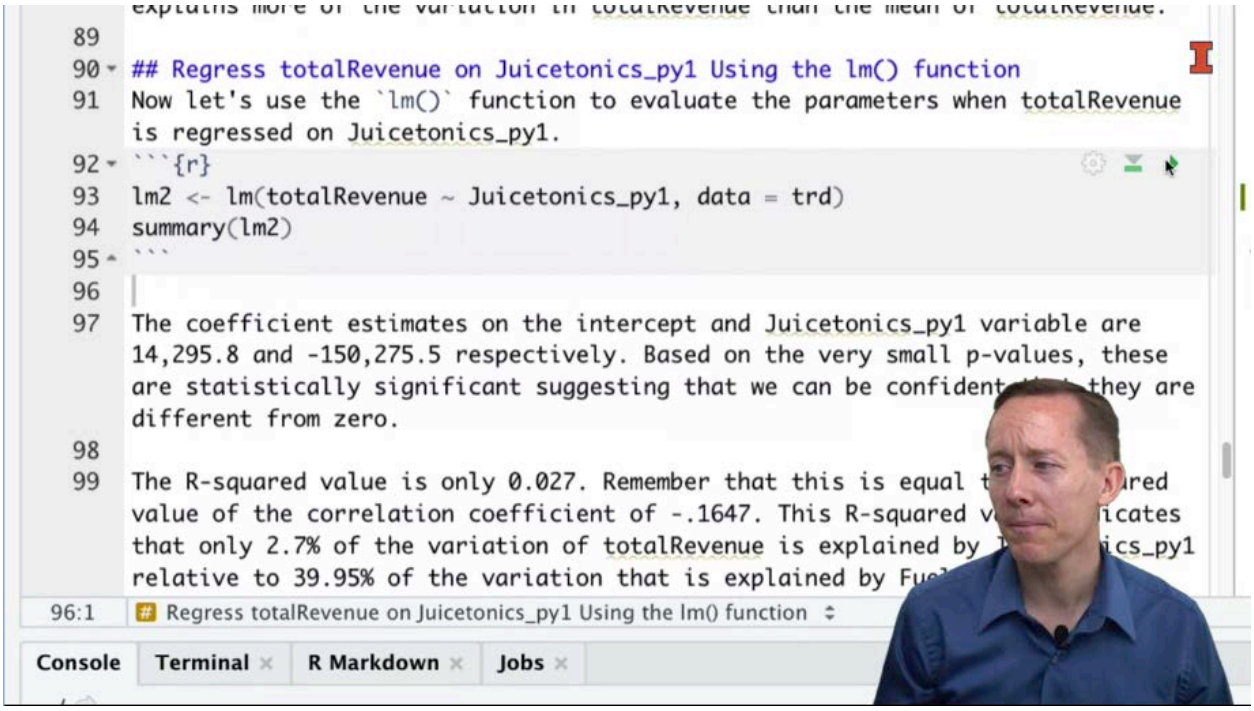


The last piece of information for now is to consider the p-value of less than 0.2^-16 that corresponds to the F statistic. This tells us whether or not the model improves predictions relative to using only the average value. A p-value of less than 0.05 is the

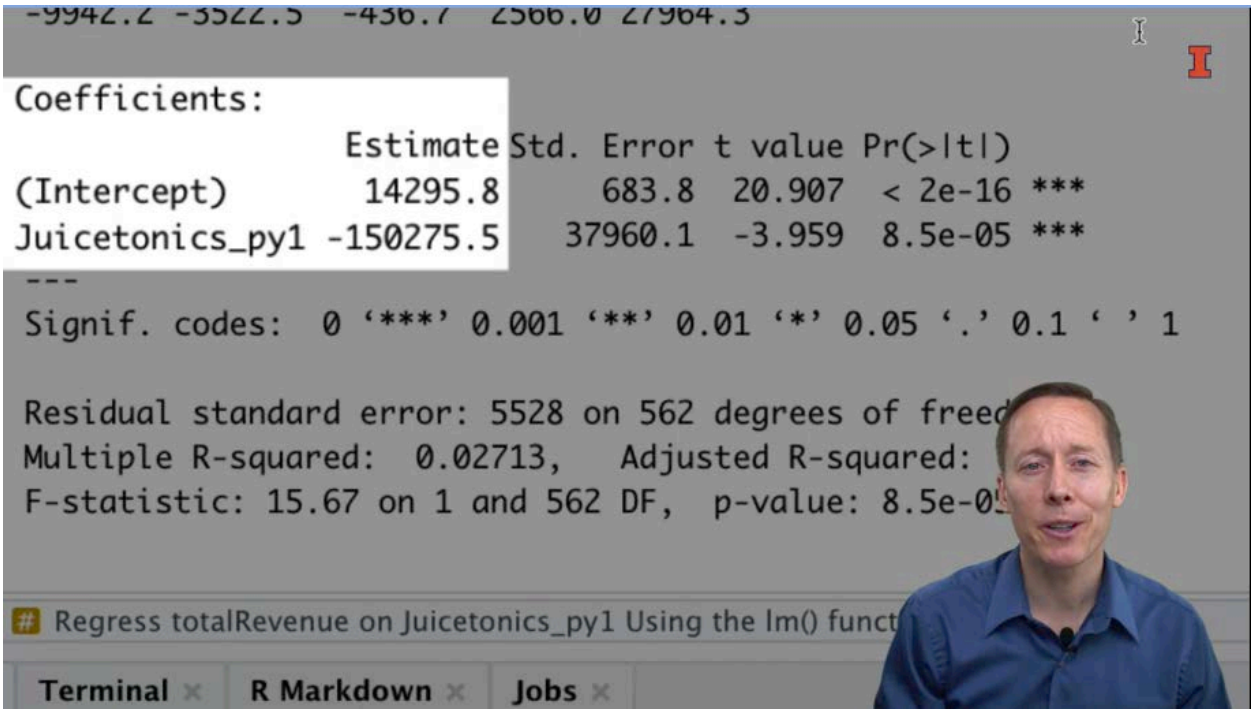
typical cut-off point for concluding that the model explains more of the variation in total revenue than just using the mean of total revenue. Now, you don't need to remember all of this information, but in case you have questions, that's a quick overview of what the information is in this table.



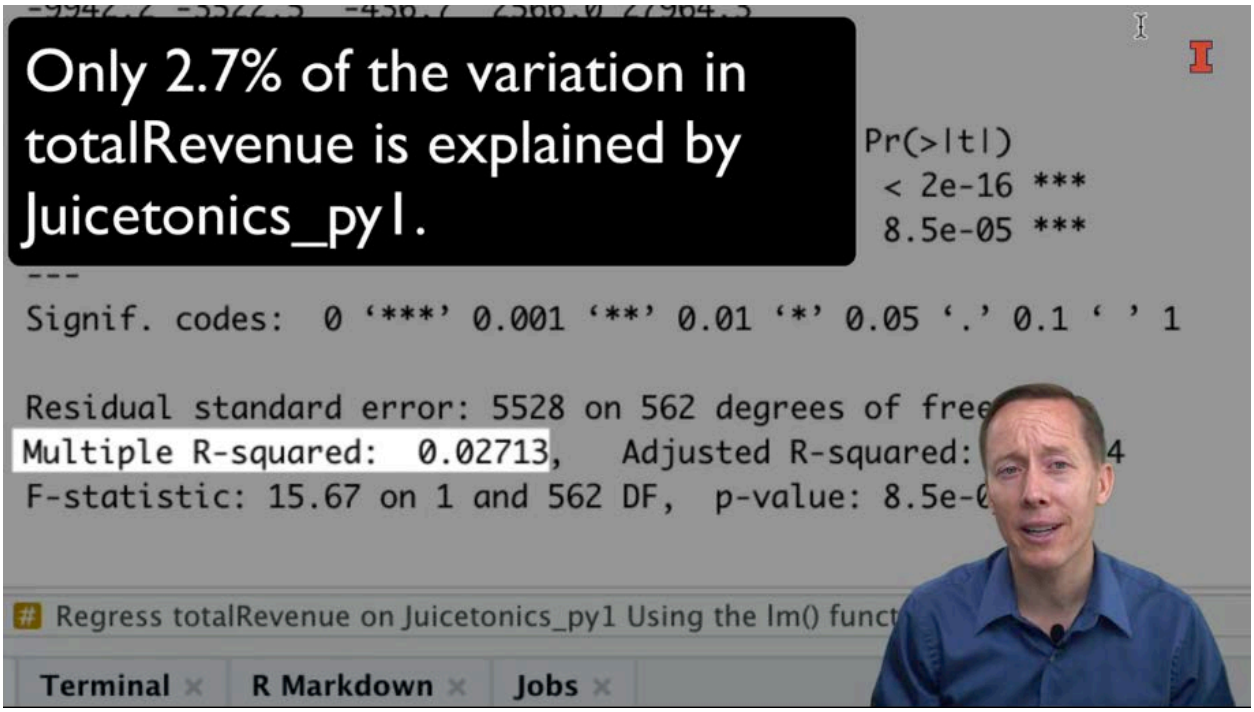
Now, when we use just one independent variable on one dependent variable, it's called simple regression.

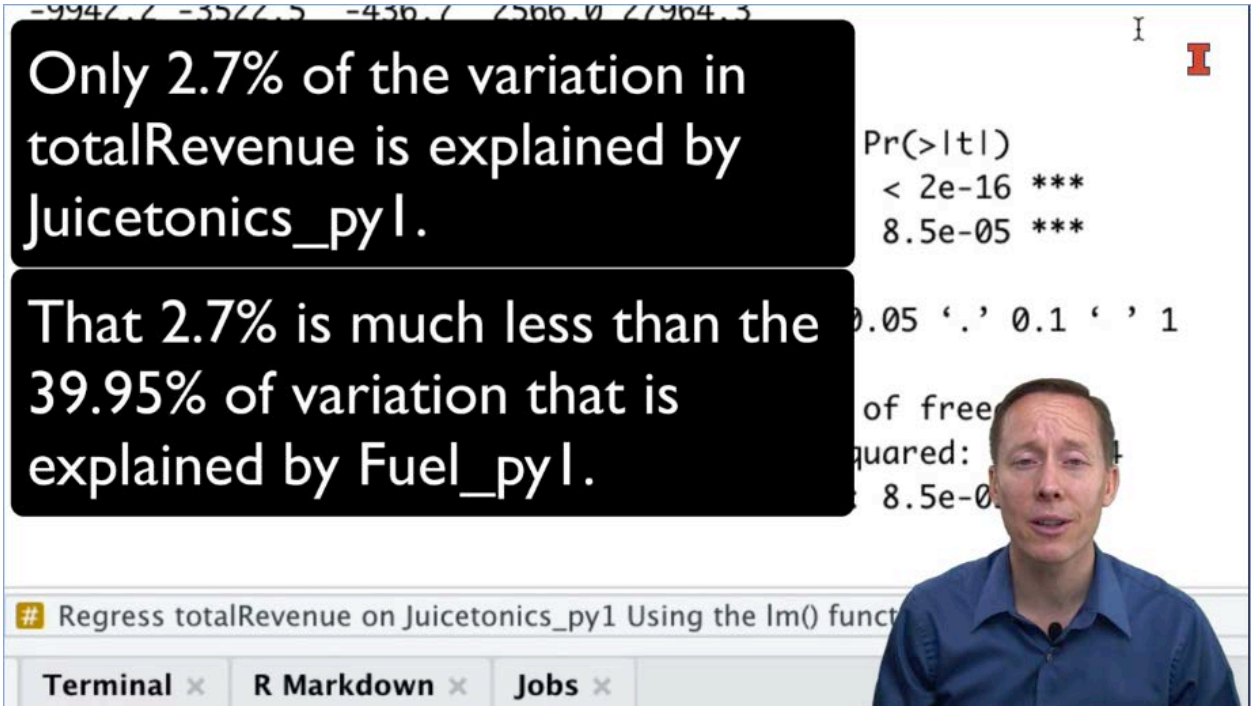


Let's try another simple regression and regress total revenue on Juicetonics_py1 and look at the results. We'll save this in the object lm2.

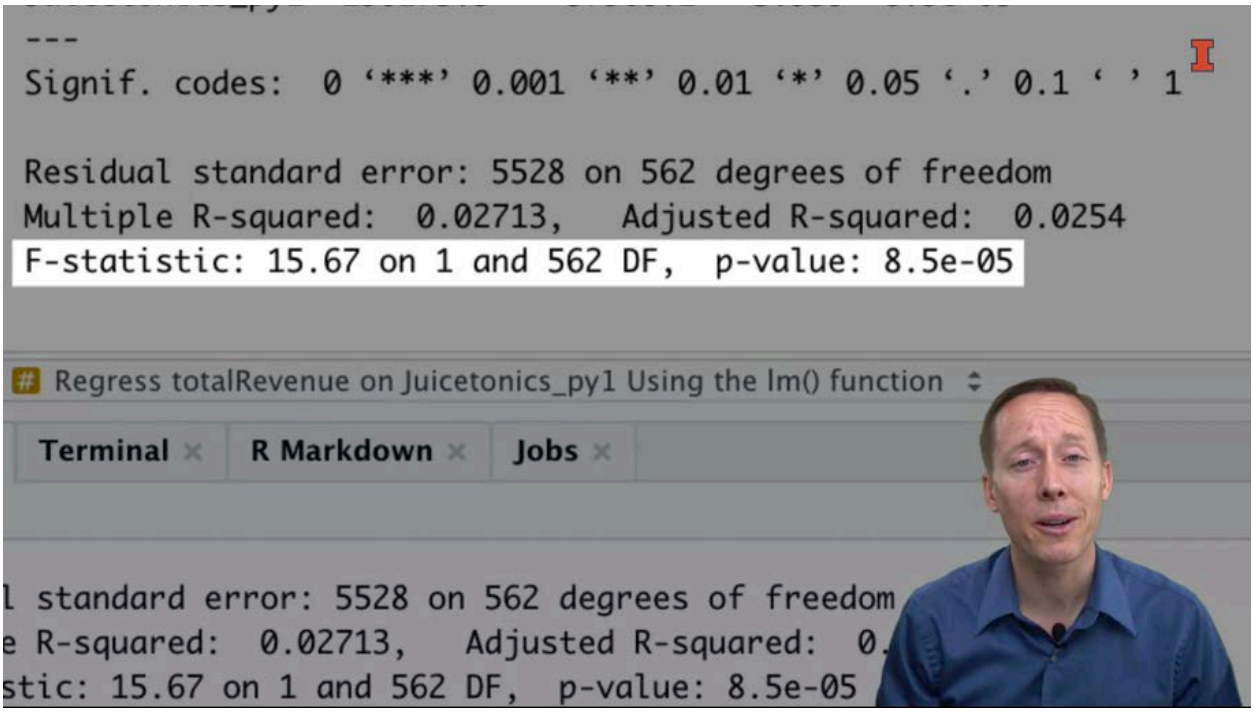


The coefficient estimates on the intercept and on Juicetonics_py1 variable are 14,295.8 and negative 150,275.5 respectively. Based on the very small p-values, these are statistically significant suggesting that we can be confident that they are different from zero.

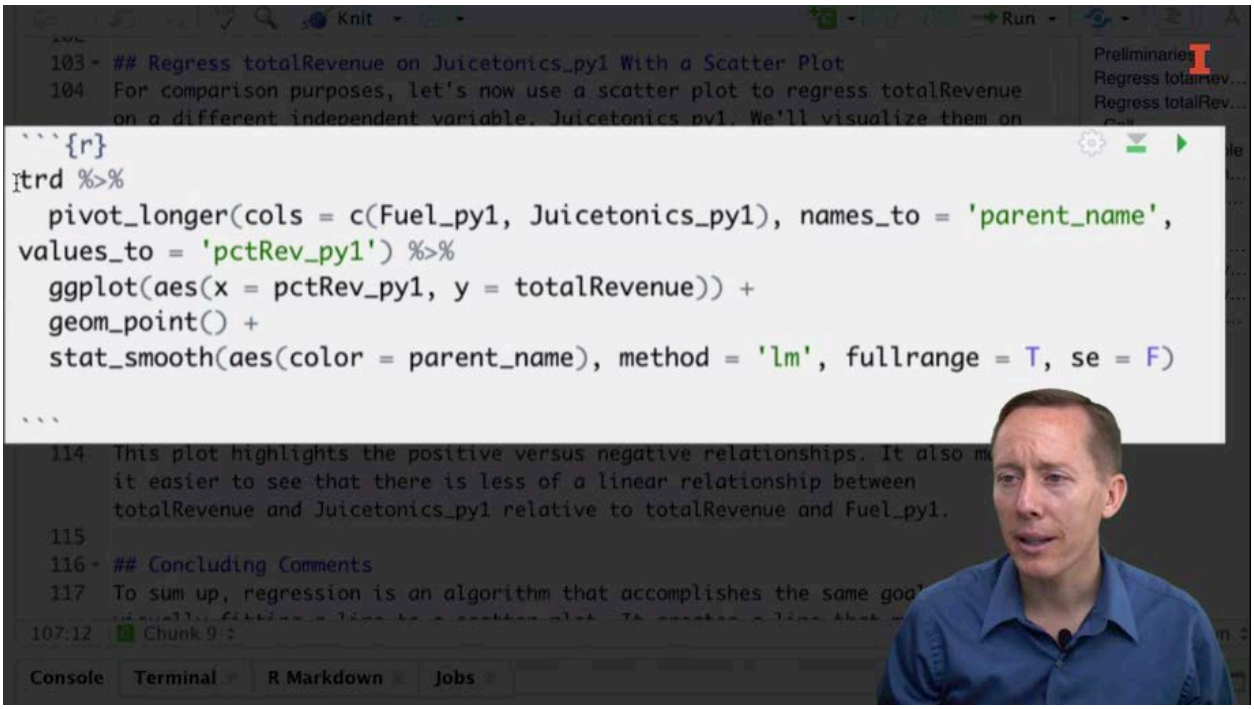




The R-squared value is only 0.027. Remember that this is equal to the squared value of the correlation coefficient of negative 0.1647. This R-squared value indicates that only 2.7 percent of the variation of totalRevenue is explained by Juicetonics_py1, relative to 39.95 percent of the variation that is explained by Fuel_py1. Now, while 2.7 percent is not very much, it's still an improvement over using only the mean.

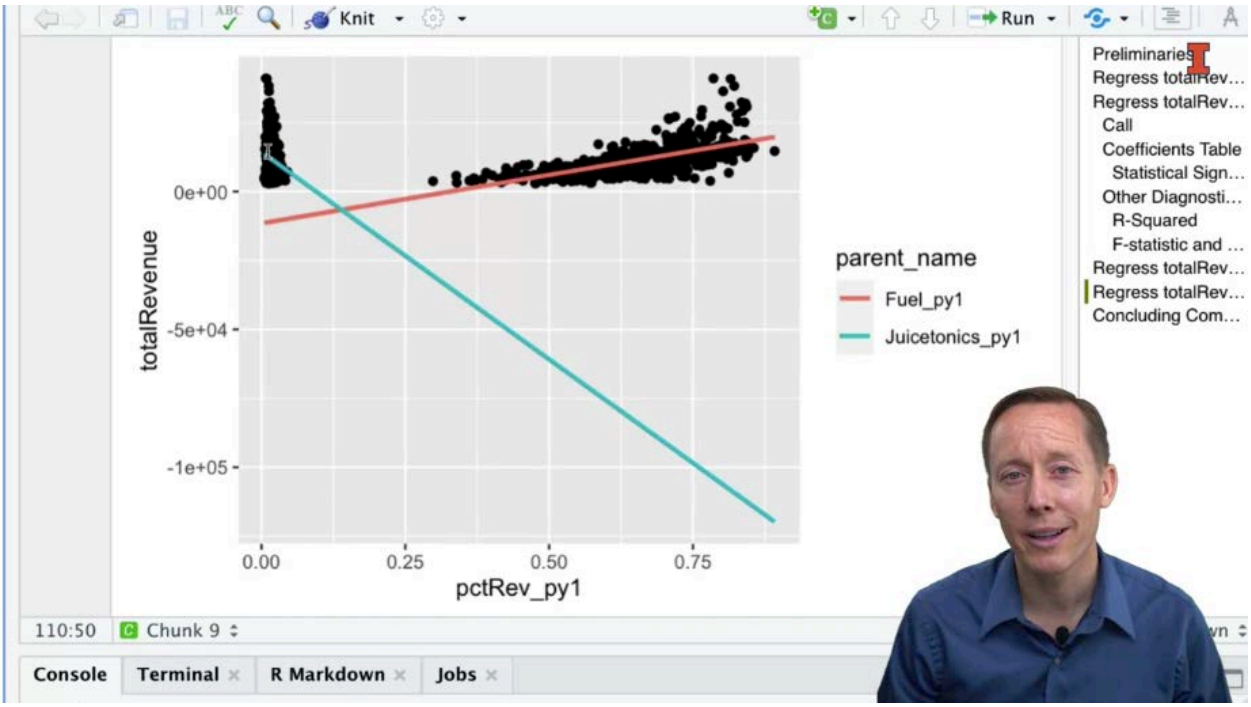


This is indicated by the p-value on the F-statistic of point 0.000085, which is much smaller than 0.05. For comparison purposes, let's now create a single scatter plot that has the regression line for both Fuel_py1 and Juicetonics_py1 on total revenue.

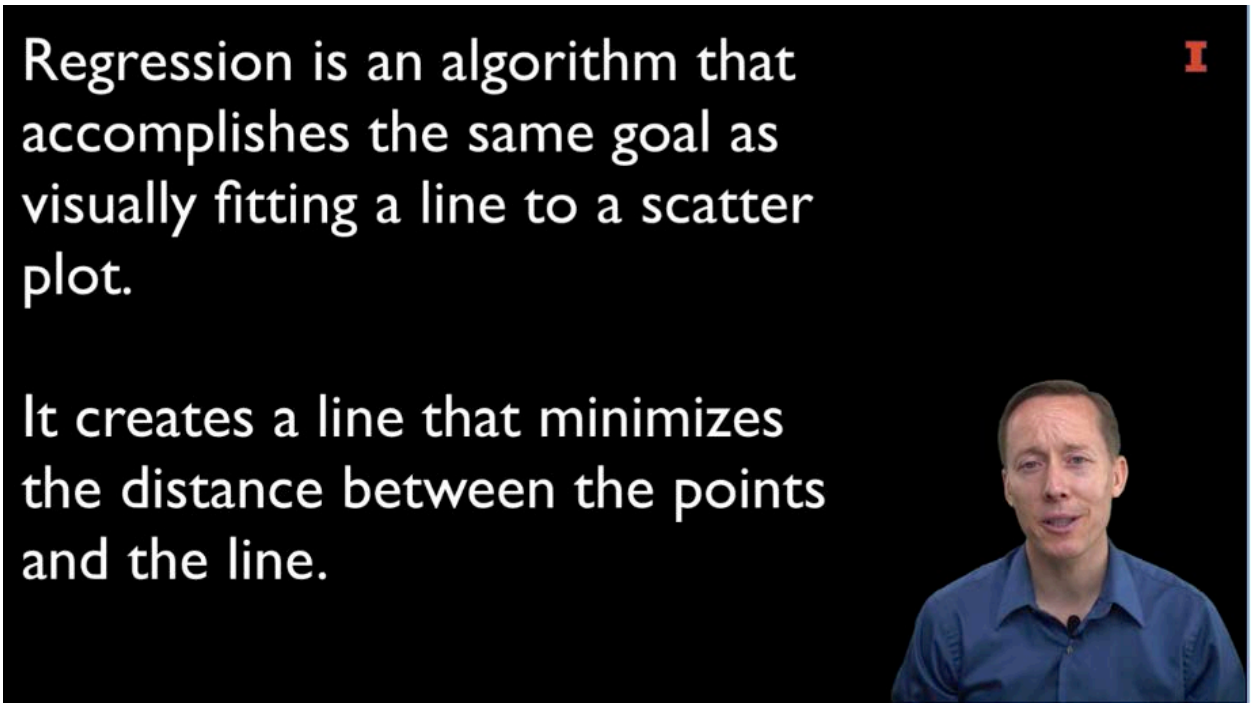


The way I'll do that is by taking that trd DataFrame, making it longer by putting Fuel_py1 and Juicetonics_py1 into a single column called parent_name and then putting their values into a new column called pctRev_py1. Then we will put pctRev_py1 on the x-axis

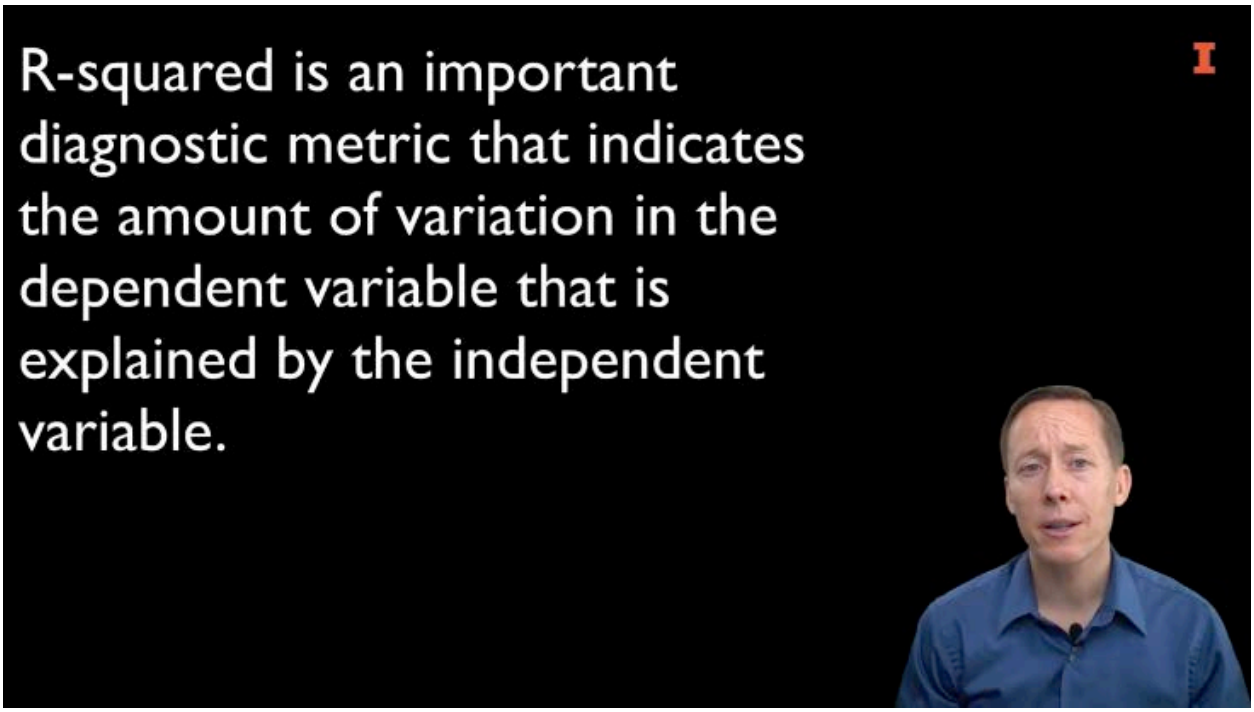
in the gg plot function, and totalRevenue on the y-axis. We'll use geom_point to create a scatter plot, and then we will create a line for each independent variable, and we will color those lines by the parent_name, and we are using the linear model method here. We'll also use the fullrange so that we can see where those lines intercept, and we will set it so that we don't see those standard error gray shadows around the lines. Let's go ahead and run that.



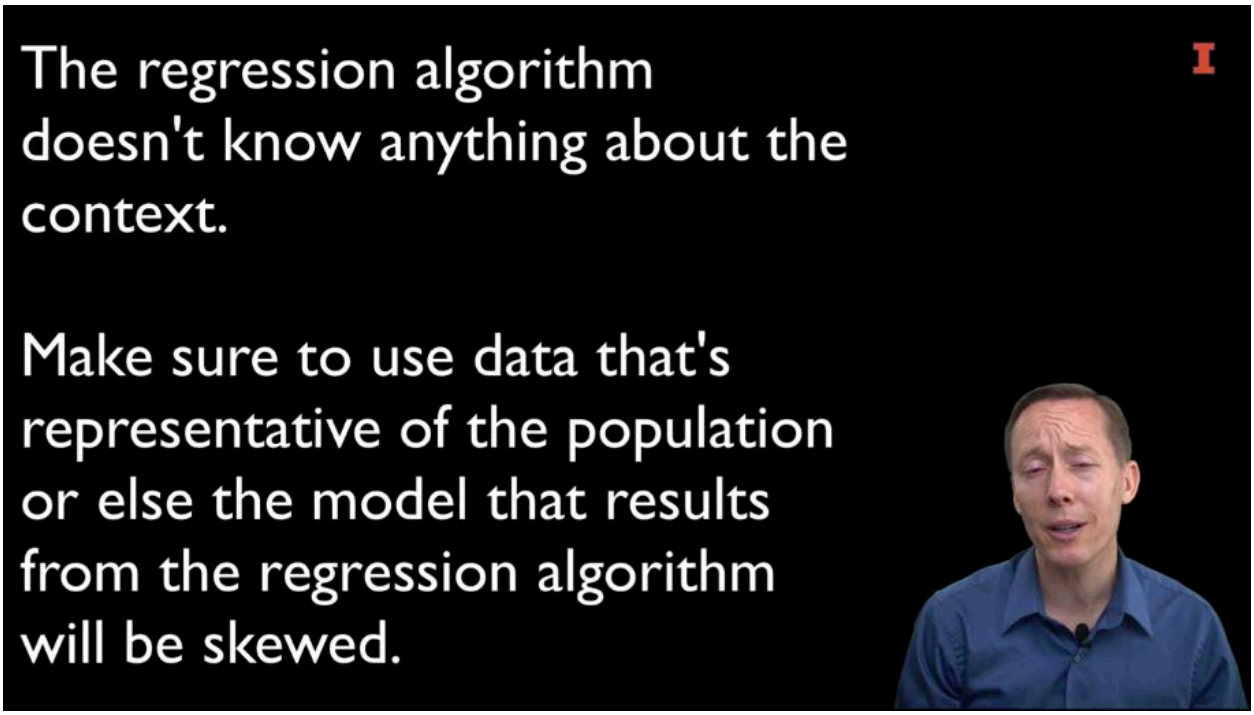
This scatter plot makes it easy to see that Fuel_py1 has a positive relationship while Juicetonics_py1 has a negative relationship with totalRevenue. It's also easier to see that there's much more variation around the line for Juicetonics relative to Fuel, which is why Juicetonics has a much lower R-squared in that simple regression.



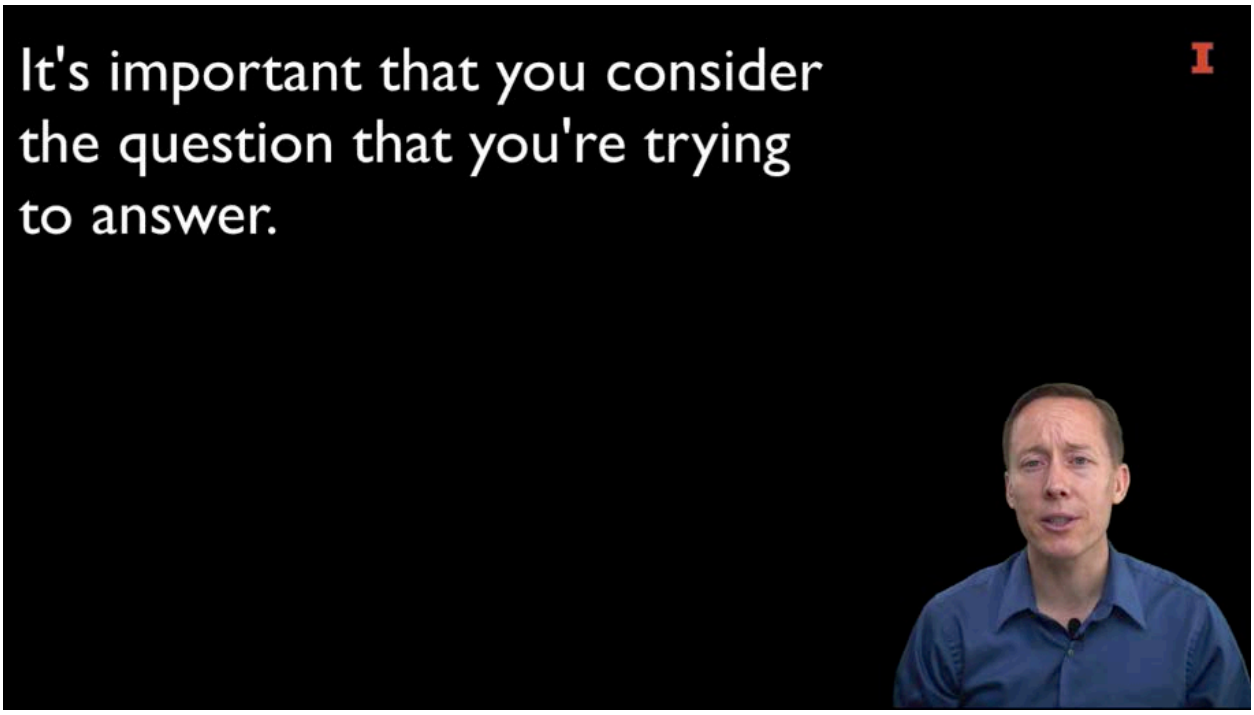
To sum up, regression is an algorithm that accomplishes the same goal as visually fitting a line to a scatter plot. It creates a line that minimizes the distance between the points and the line.



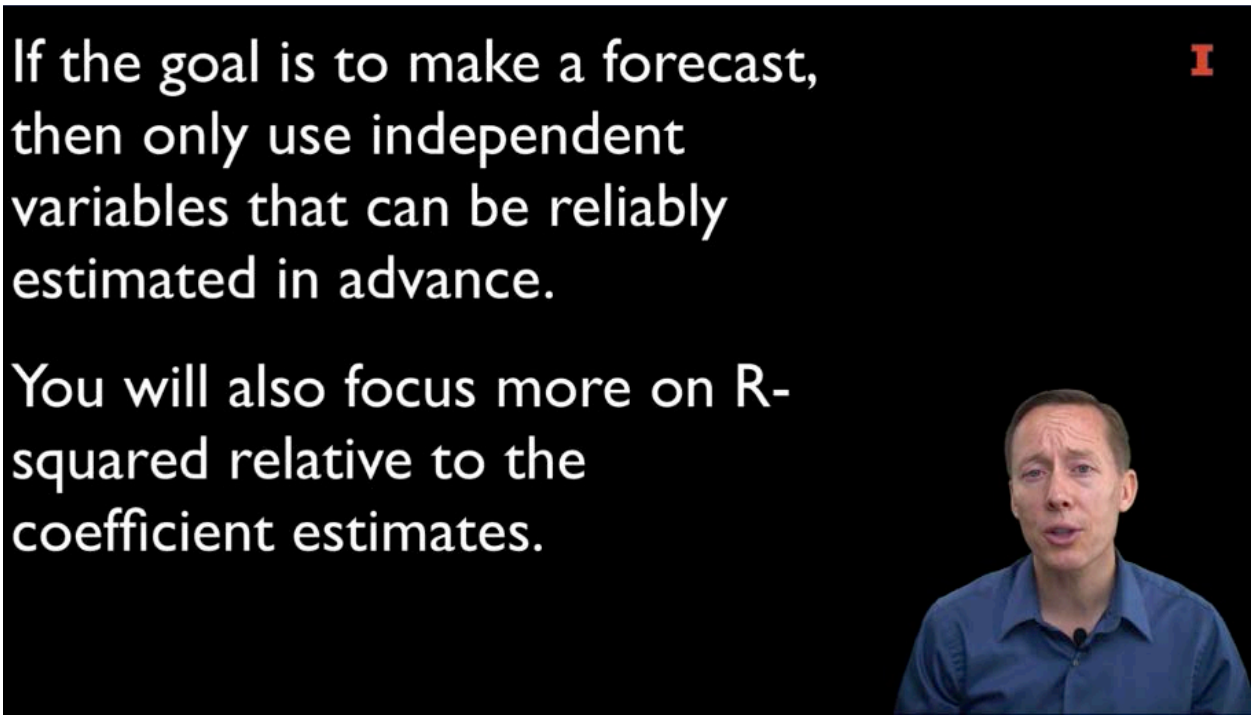
R-squared is an important diagnostic metric that indicates the amount of variation in the dependent variable that is explained by the independent variable. Let's consider a few important points in relation to using this algorithm for business analytics.



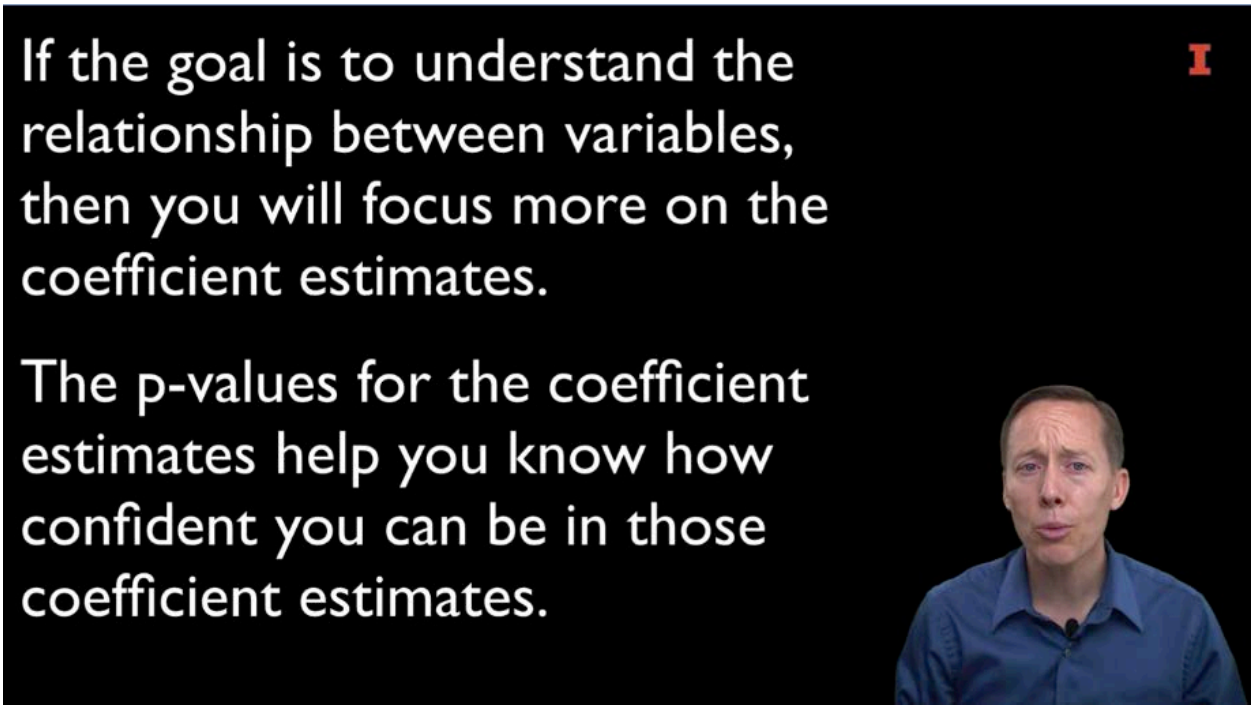
First, the regression algorithm doesn't know anything about the context, so it's important to make sure that you're using data that is representative of the population, or else the model that results from the regression algorithm will be skewed towards that unrepresented data, such as outliers, and it will not give helpful predictions.



Second, it's important that you consider the question that you're trying to answer.



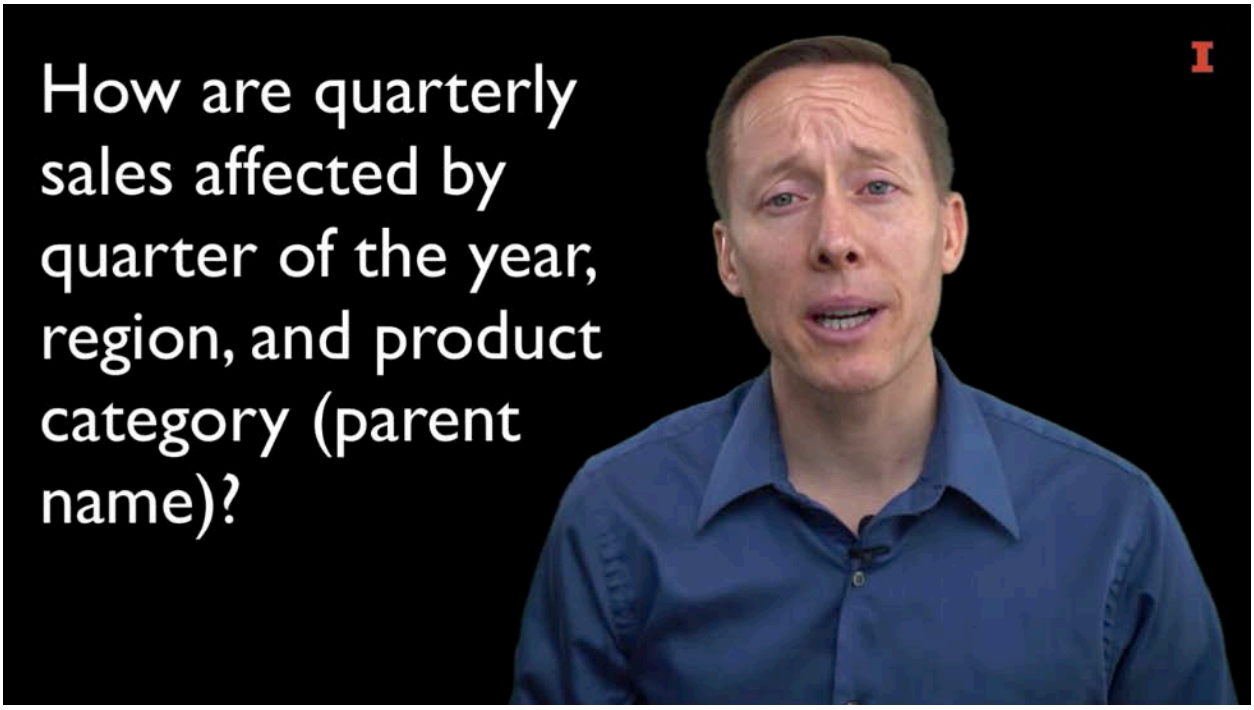
If the goal is to make a forecast as it is in our example, then you should make sure that you consider only variables that can be reliably estimated in advance, like last year sales. You'll also probably focus more on R-squared relative to the coefficient estimates.



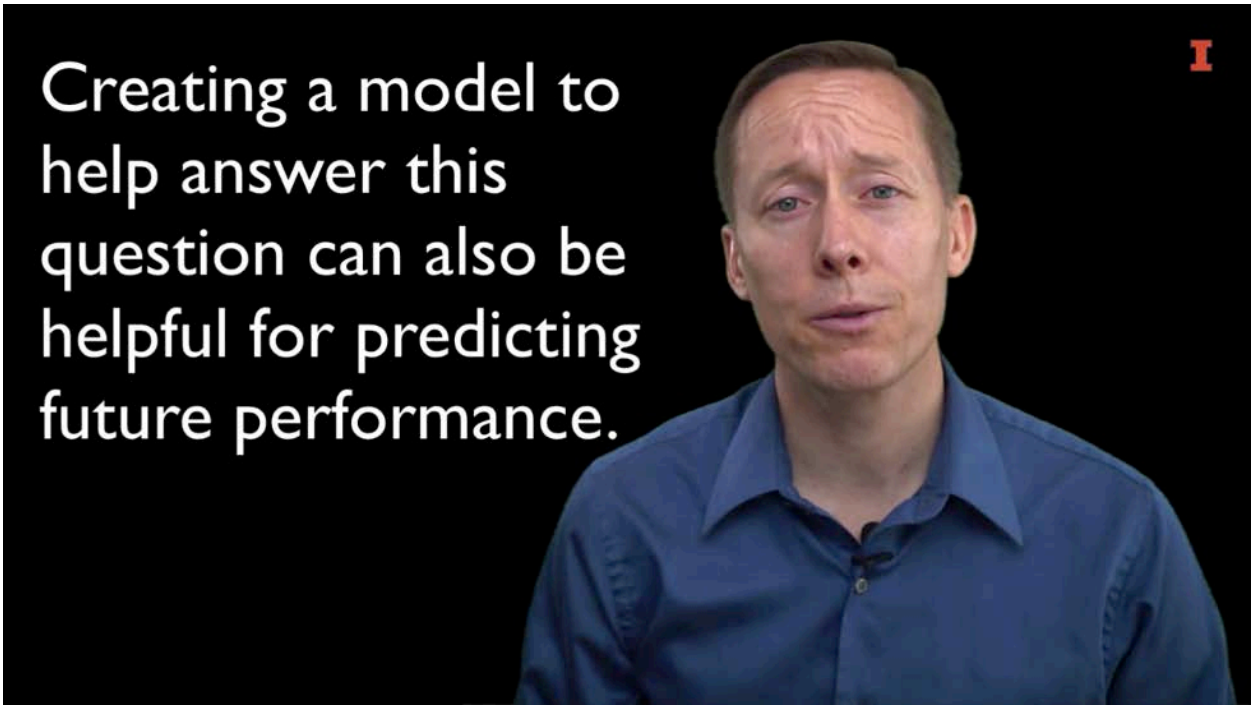
On the other hand, if the goal is to understand the relationship between variables, then you will probably focus more on the coefficient estimates because they can quantify how the dependent variable changes with a one-unit change in the independent

variable. The p-values for the coefficient estimates help you know how confident you can be in those coefficient estimates.

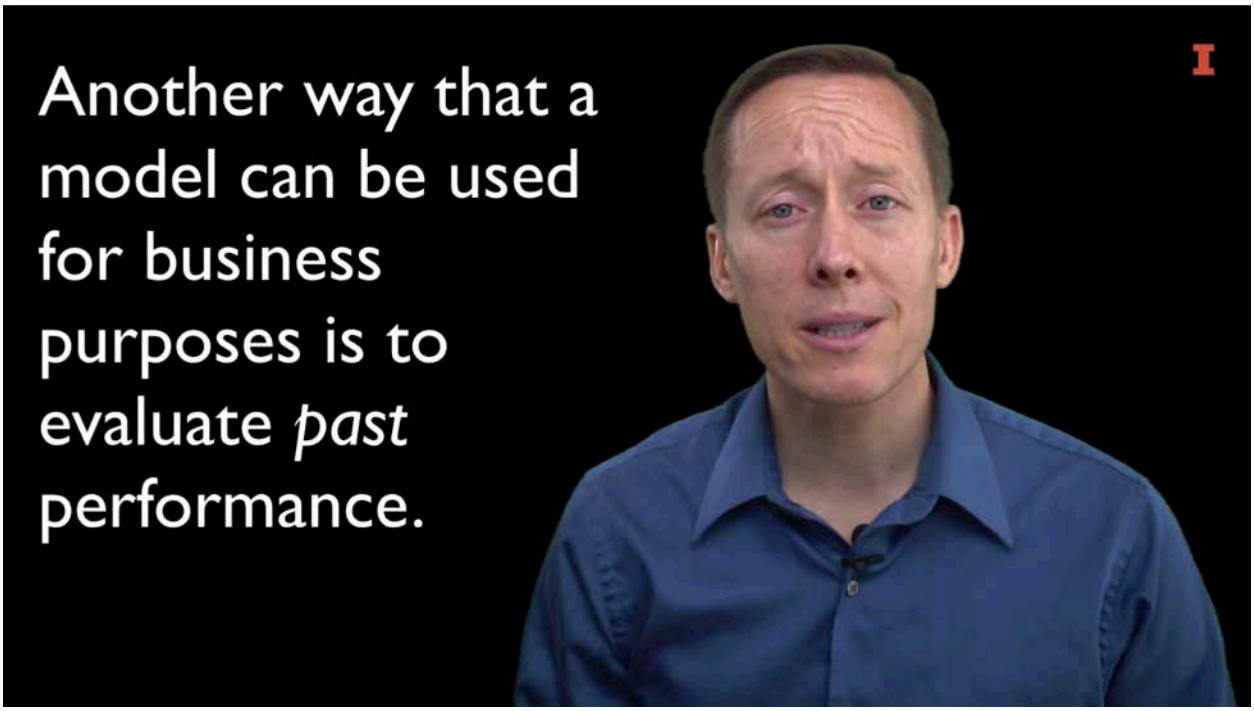
Lesson 1-4.4 Residuals and Predictions



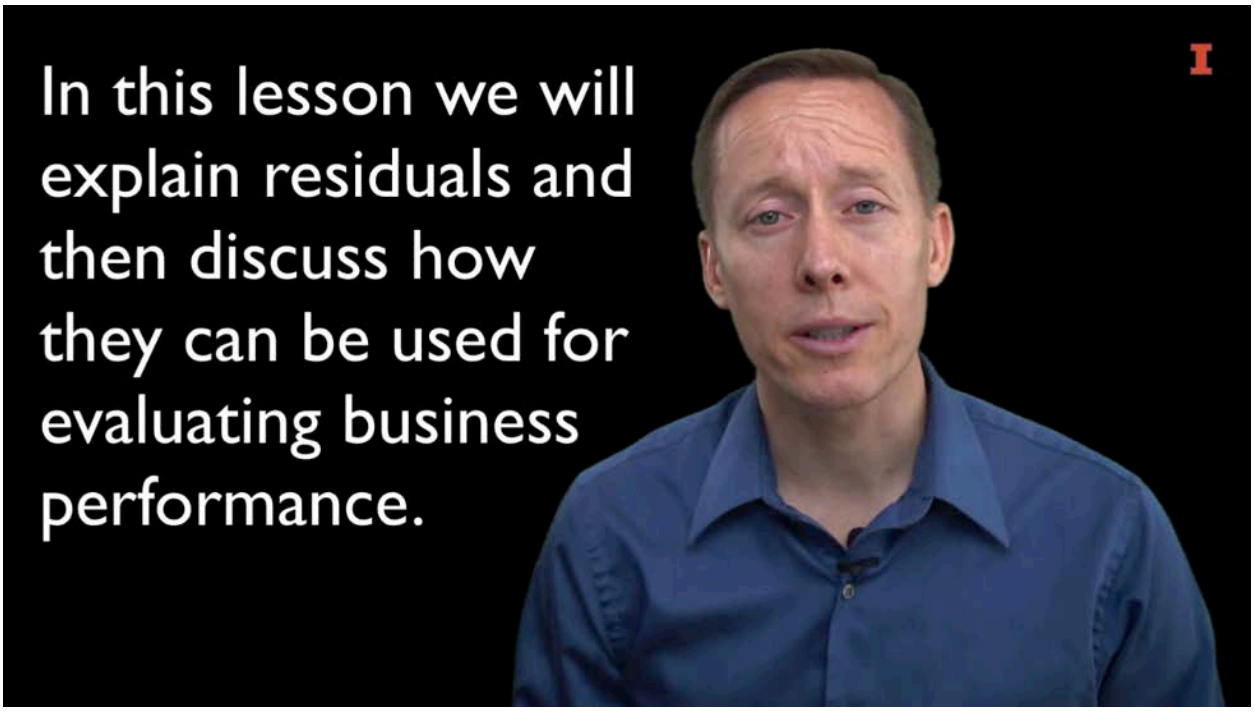
Recall that our broad business question is, how are quarterly sales affected by quarter of the year, region, and by product category?



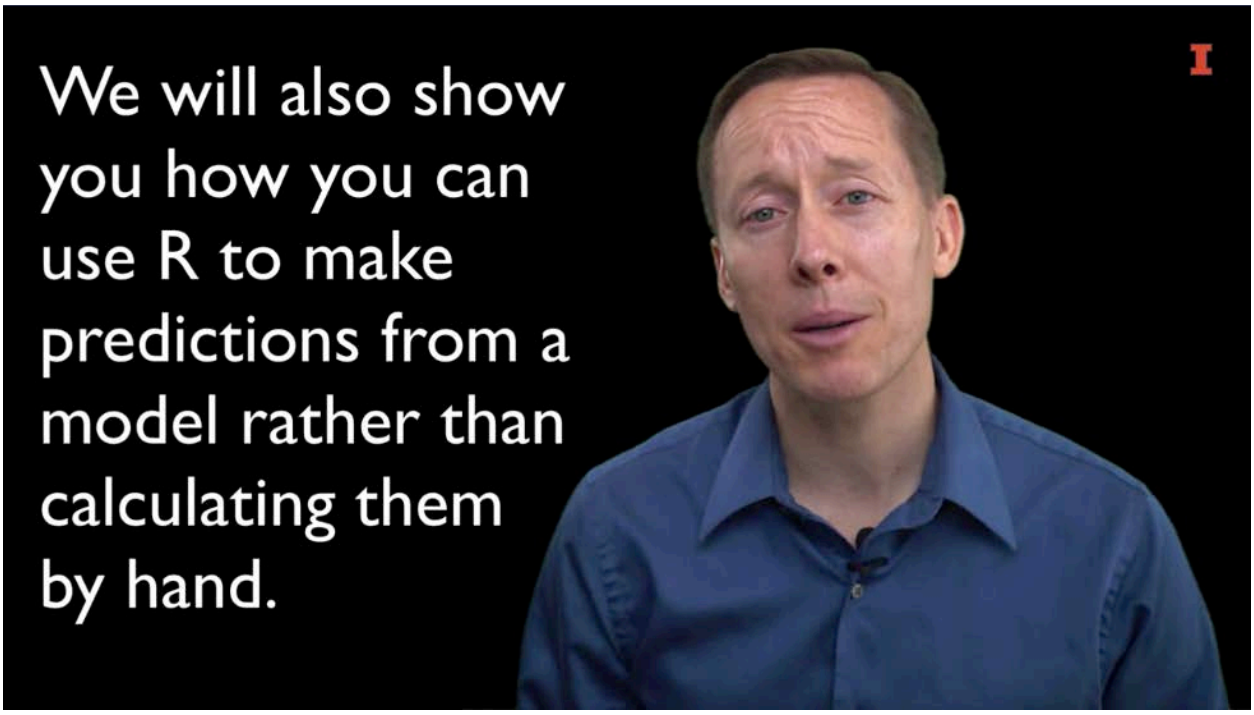
Creating a model to help answer this question can certainly be helpful for predicting future performance.



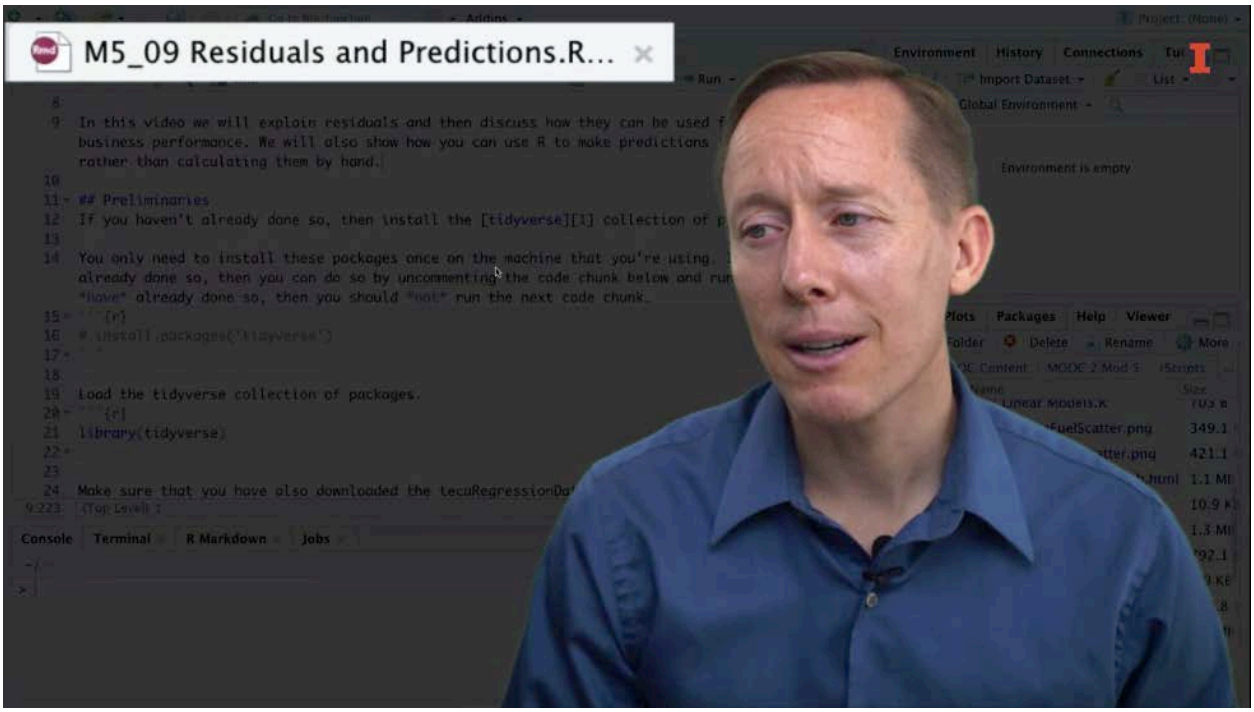
Another way in which the model can be used for business purposes is to evaluate past performance.



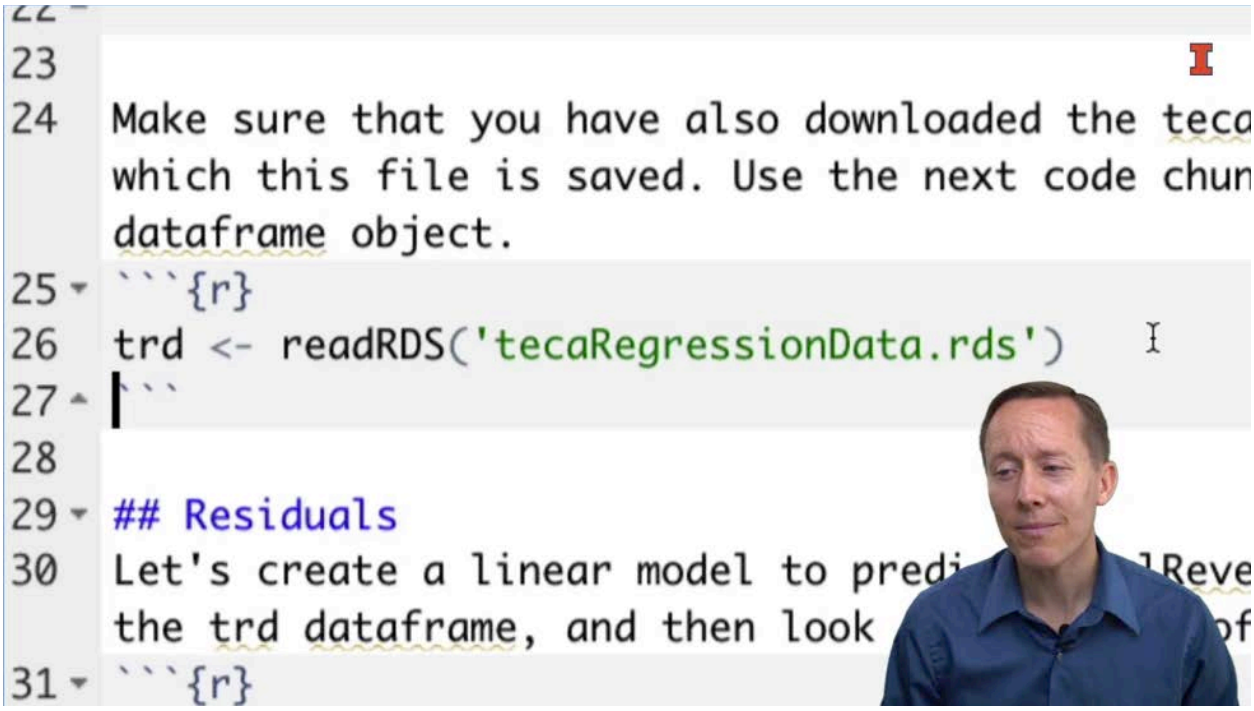
In this lesson, we will explain residuals and then discuss how they can be used for evaluating business performance.



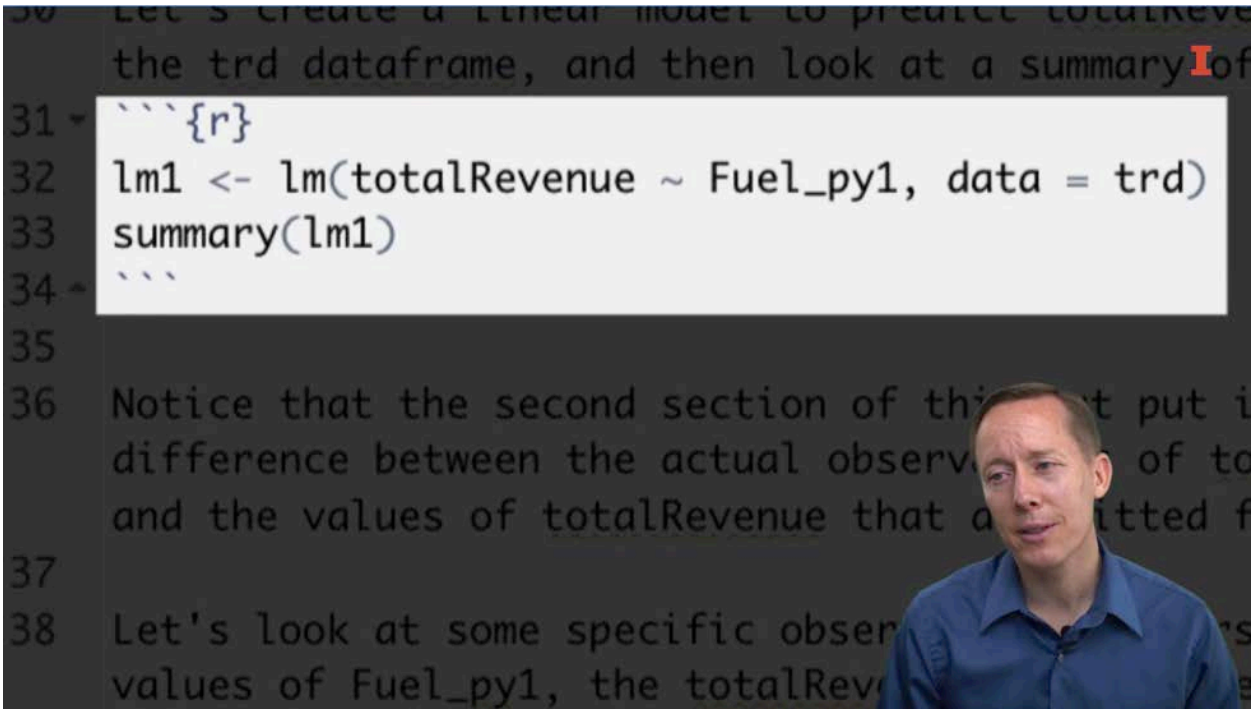
We will also show you how you can use R to make predictions from a model rather than by calculating them by hand.



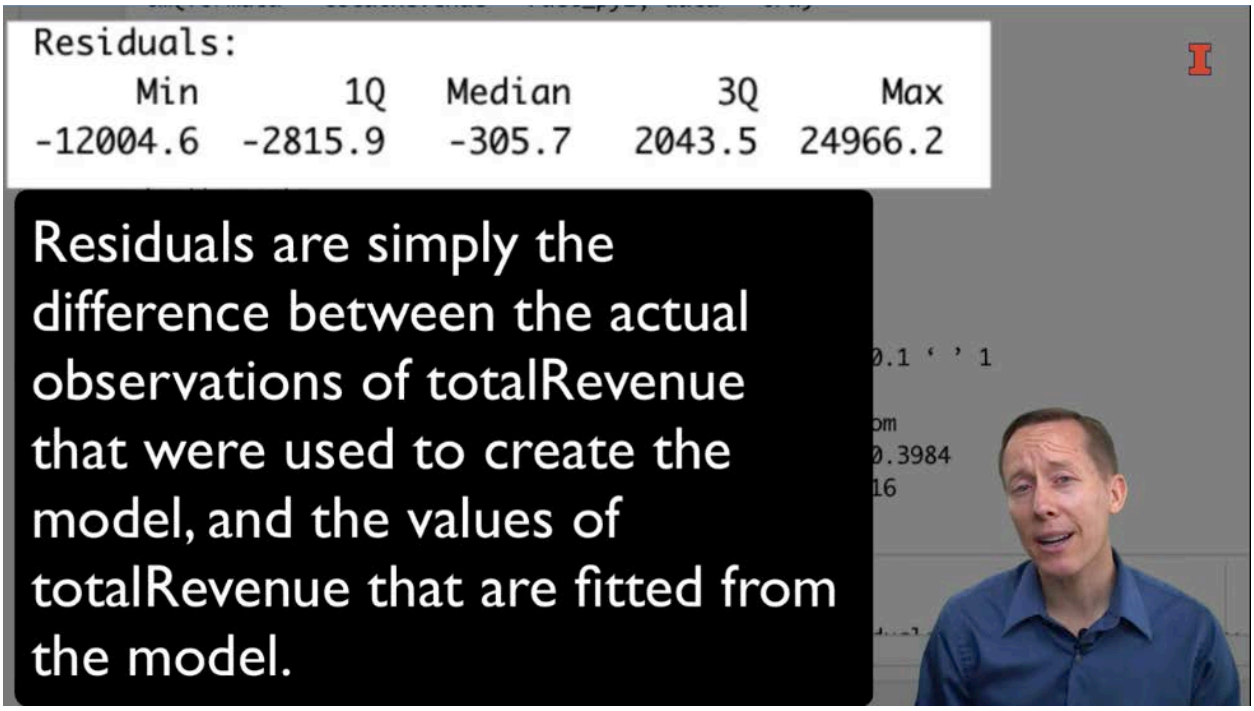
If you want to follow along, then go ahead and download the associated R Markdown file and install the tidyverse collection of packages and load those packages.



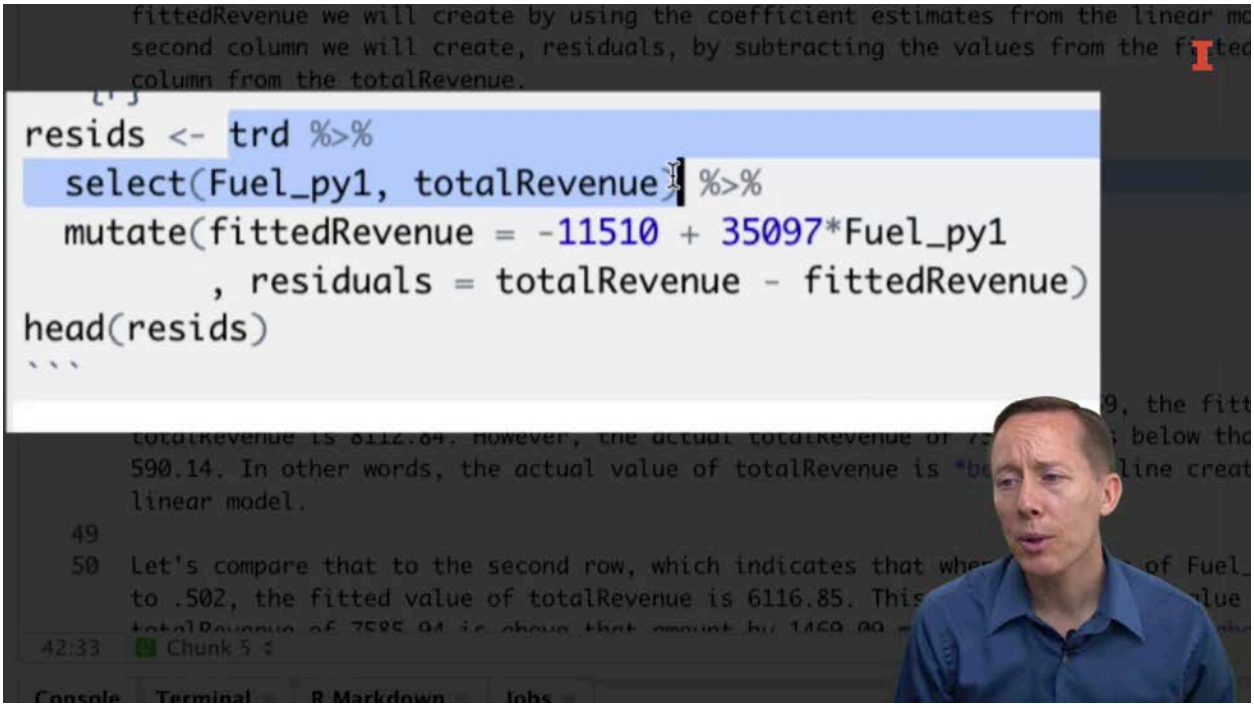
Also, make sure that you read in the tecaRegressionData, and save it as a dataframe called TRD.



Then we'll create a linear model that takes total revenue and regresses it on Fuel_py1, and we'll look at the summary of that model.

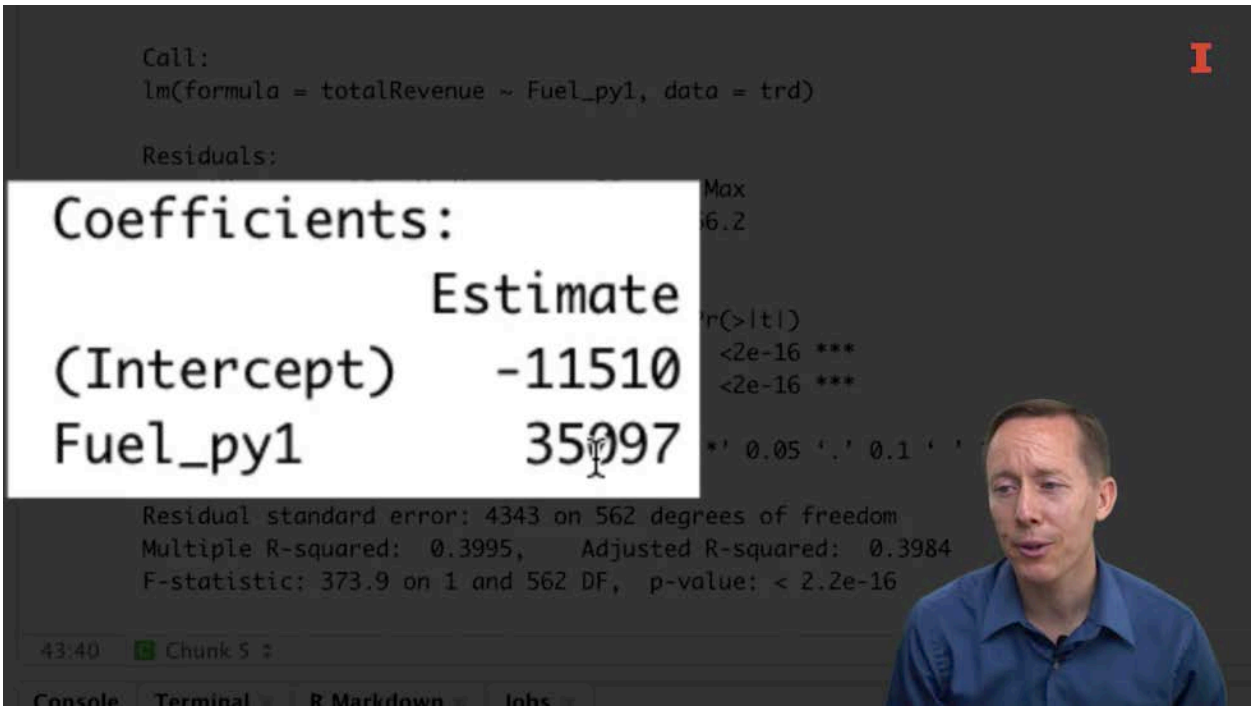


Now notice the second section in this output is residuals. Residuals are simply the difference between the actual observations of total revenue that were used to create the model, and the values of total revenue that are fitted from the model.

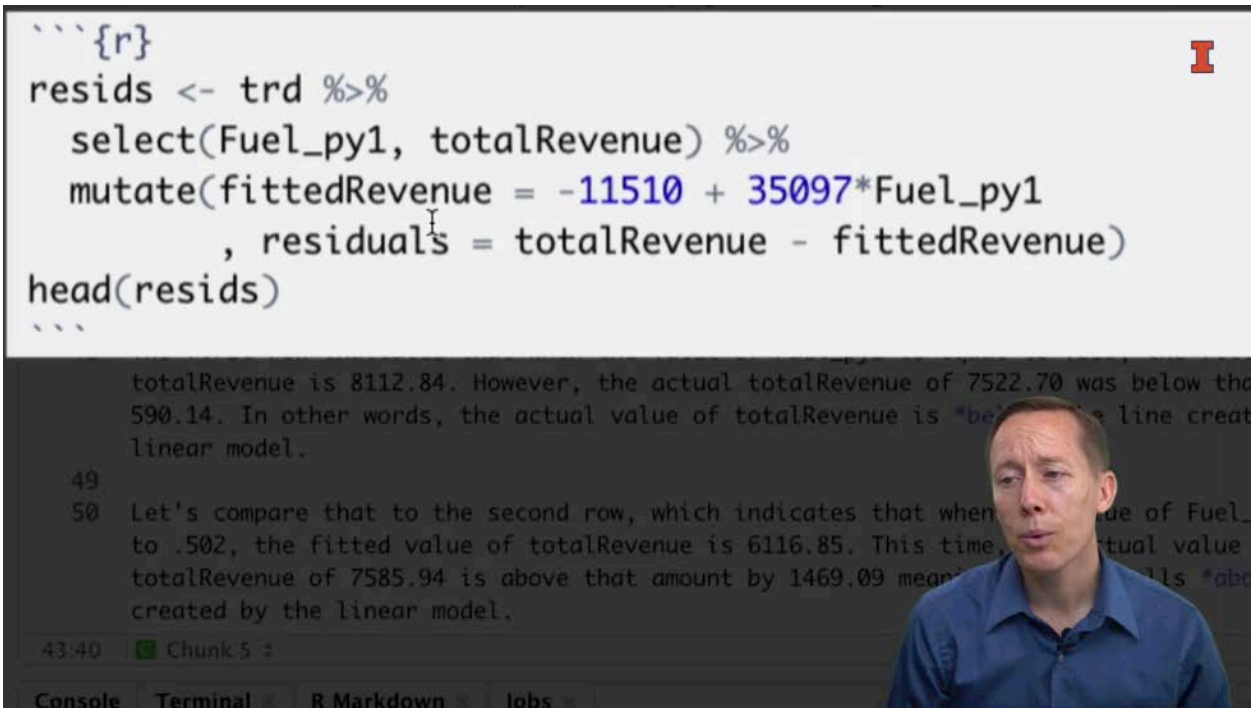


Let's look at some specific observations by first creating a DataFrame that has only Fuel_py1 and total revenue from the original trd DataFrame. Then we're going to pipe that into the mutate function, and we will create two new columns. The first one, fitted

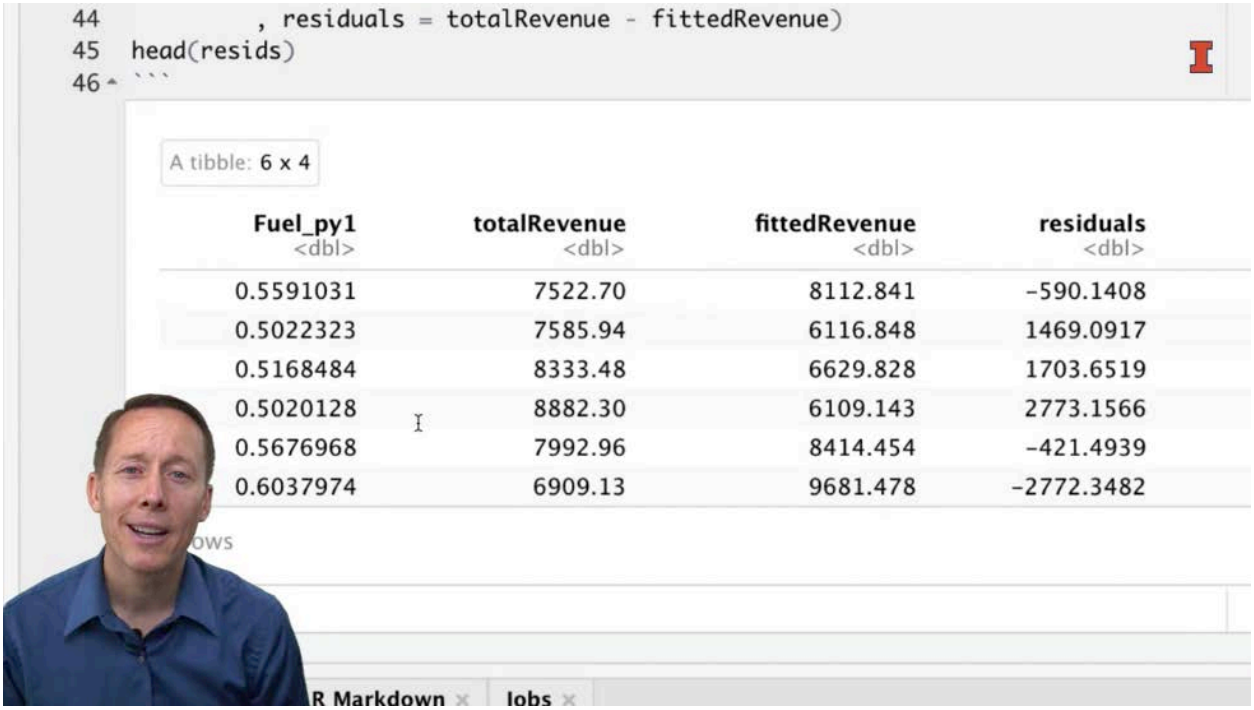
revenue, is equal to negative 11,510 plus 35,097 times the actual value of Fuel_py1. Now, you might notice that these two values right here come from the regression results.



They are the coefficient estimates for the intercept and Fuel_py1.



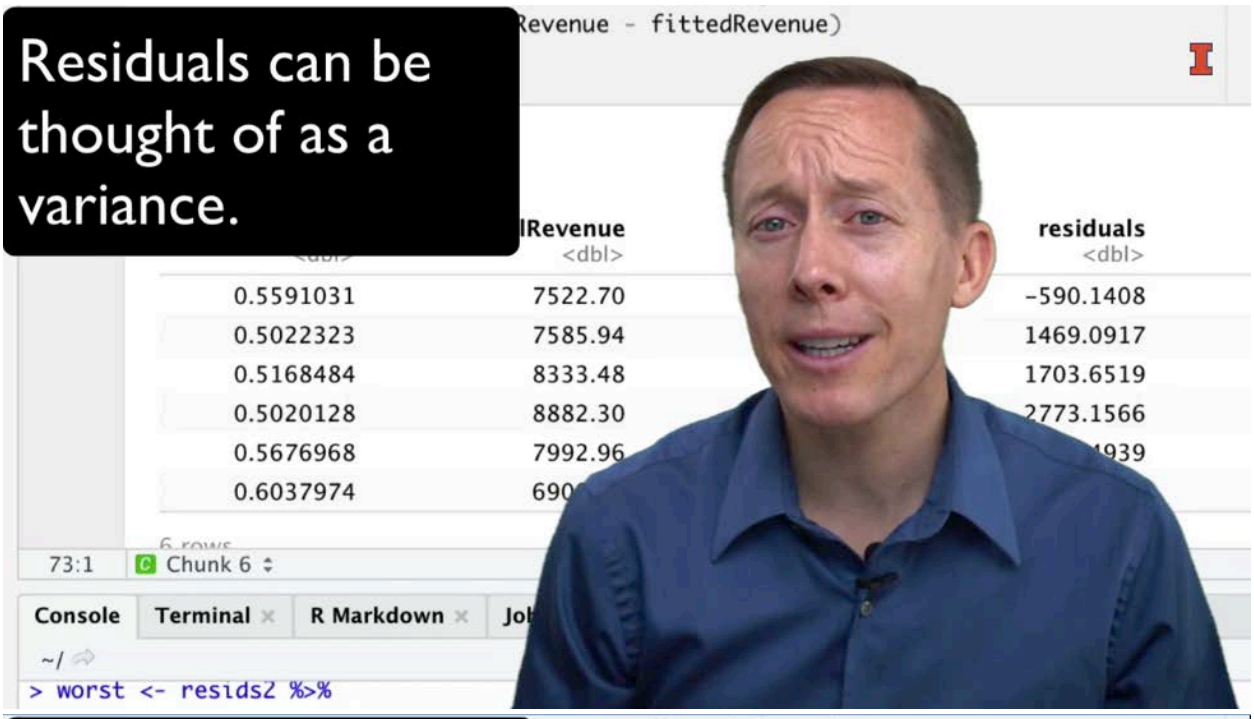
That's how we can create the fitted values for total revenue. We will then create the residuals by taking the actual value of total revenue and subtracting from that these newly created fitted values.

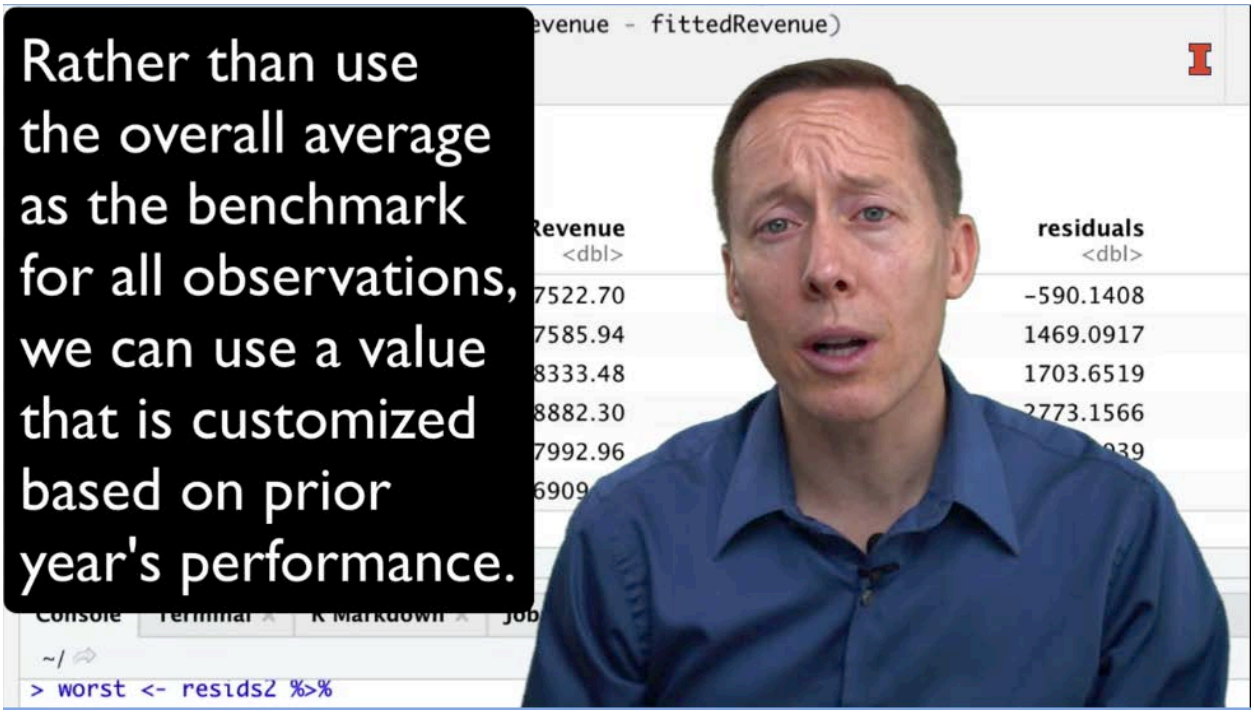


Then we'll look at the top five rows of this DataFrame. Let's just look at this first row here. This first row indicates that when the value of Fuel_py1 is equal to 0.559, the fitted value of total revenue is 8,112.84. However, the actual total revenue of 7,522.7 was

below that amount by $590.14. In other words, the actual value of total revenue is below the line created by the linear model. Let's compare that to the second row, which indicates that when the value of Fuel_py1 is equal to 0.502, the fitted value of total revenue is 6,116.85 This time, the actual value of total revenue of 7,585.94 is above that amount by $1,469.9. Meaning that it falls above the line created by the linear model.

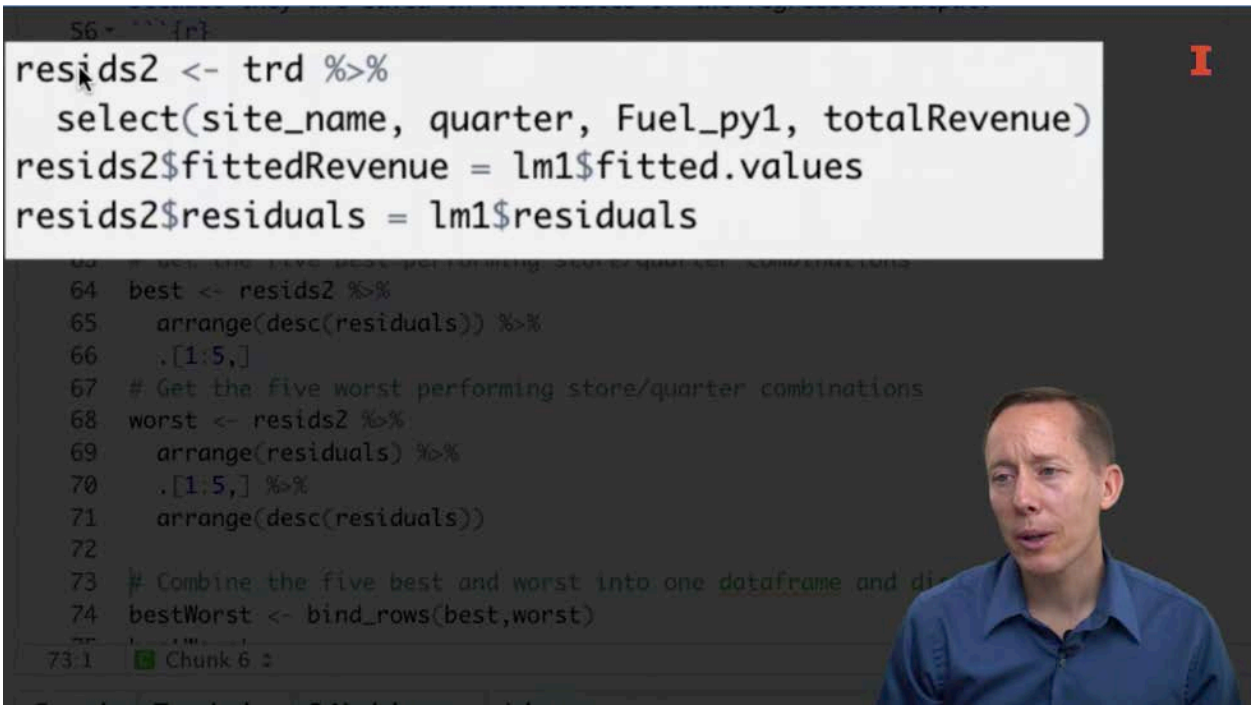
Residuals have an important use for business management. If we think of the fitted values as a target or expectation of what total quarterly revenue should be, then the residual tells us whether that revenue is more or less than expected.



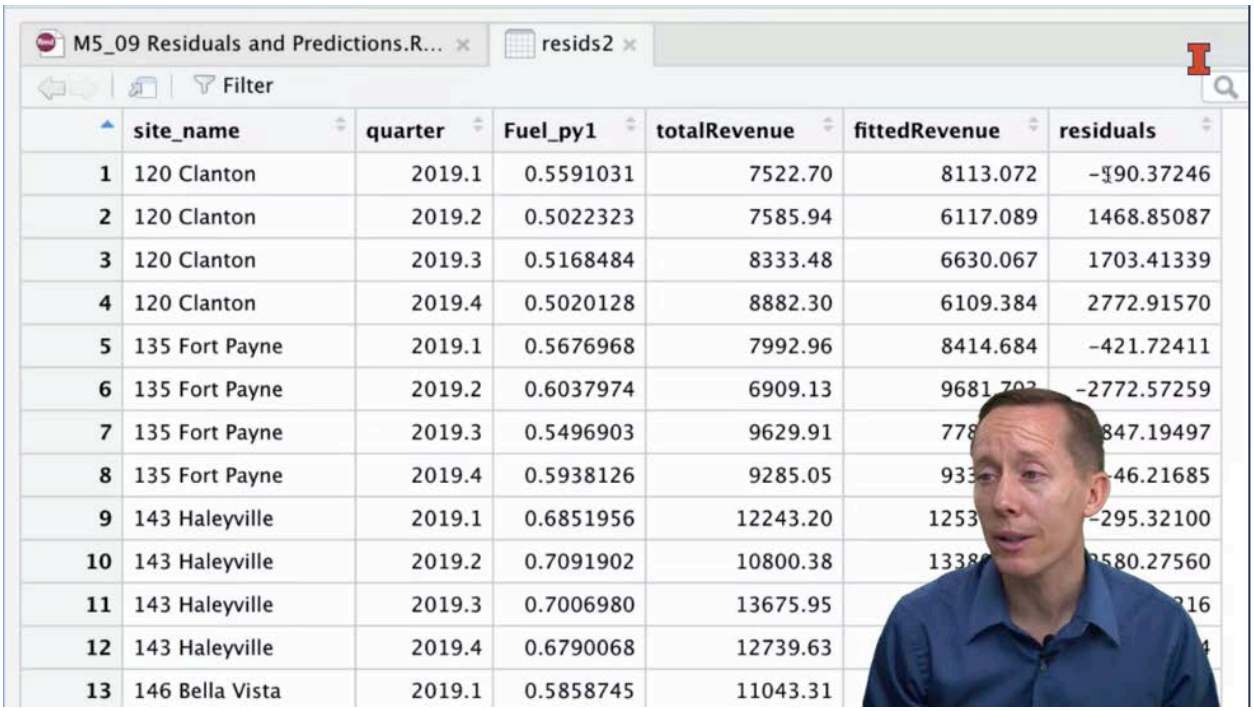


In other words, it can be thought of as a variance. Rather than use the overall average as the benchmark for all observations, we can use a value that is customized based on prior year's performance.

Let's go ahead and create a new DataFrame that we can use to identify the five best performing store quarter combinations as well as the five worst-performing store quarter combinations.

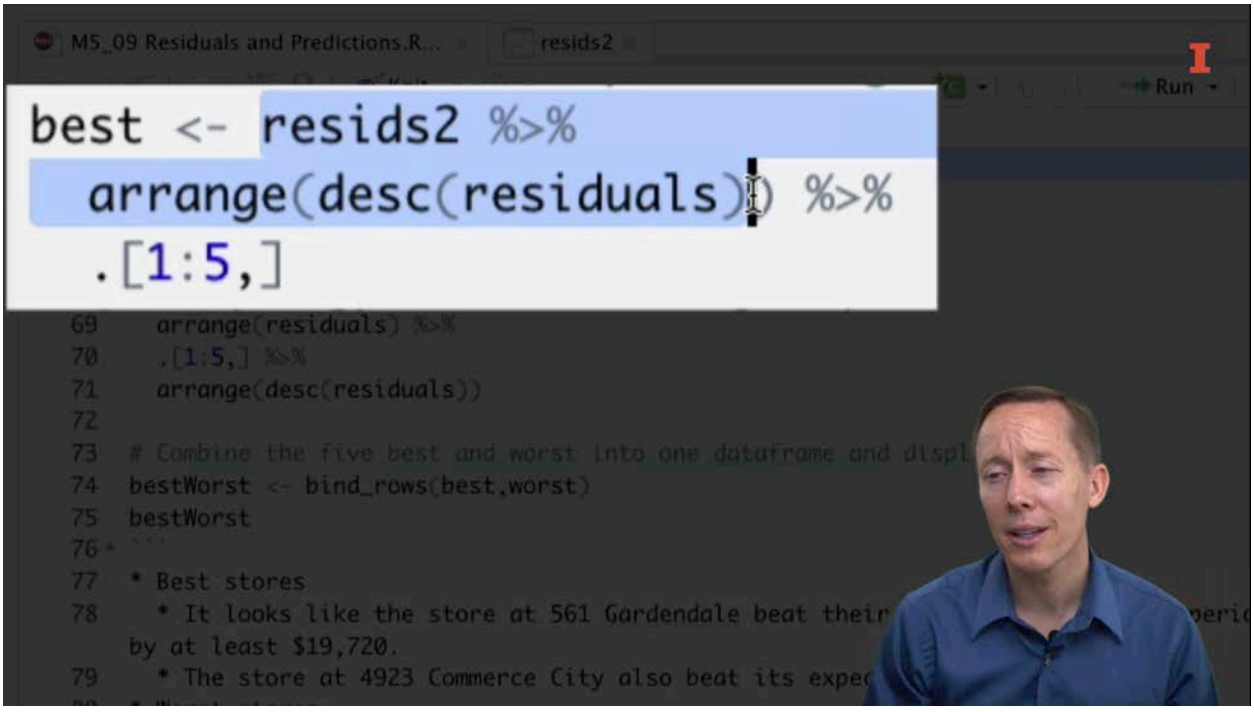


I'm going to create this resids2 DataFrame by taking the trd DataFrame and only selecting site name, quarter, Fuel_py1, and total revenue. Then I'm going to add two new columns to that. The first column, fitted revenue, is going to come from the lm1 object, that's the linear model that we just created, and we're going to take the fitted values from that object. Notice that this object from the linear model actually stores the fitted values, and we don't have to create them by using the mutate function. We can also create a residuals column by taking the residuals that are stored in that LM1 object.



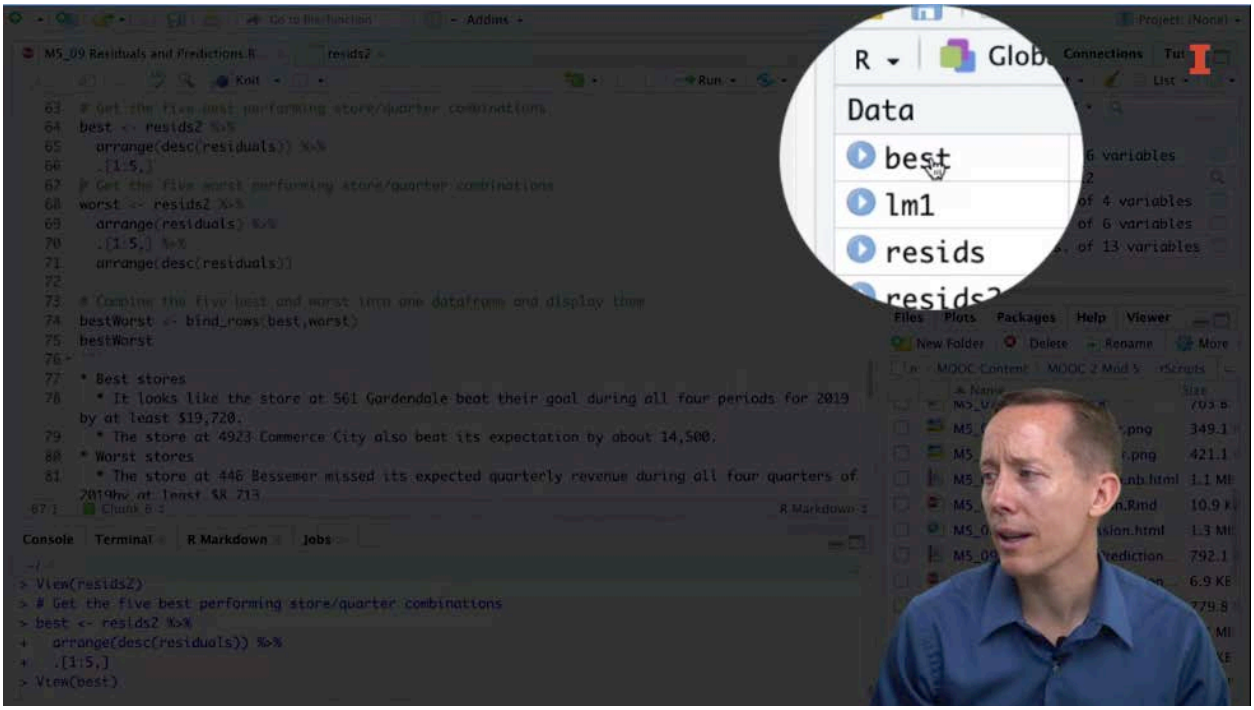
	site_name	quarter	Fuel_py1	totalRevenue	fittedRevenue	residuals
1	120 Clanton	2019.1	0.5591031	7522.70	8113.072	-590.37246
2	120 Clanton	2019.2	0.5022323	7585.94	6117.089	1468.85087
3	120 Clanton	2019.3	0.5168484	8333.48	6630.067	1703.41339
4	120 Clanton	2019.4	0.5020128	8882.30	6109.384	2772.91570
5	135 Fort Payne	2019.1	0.5676968	7992.96	8414.684	-421.72411
6	135 Fort Payne	2019.2	0.6037974	6909.13	9681.703	-2772.57259
7	135 Fort Payne	2019.3	0.5496903	9629.91	778	847.19497
8	135 Fort Payne	2019.4	0.5938126	9285.05	93	-46.21685
9	143 Haleyville	2019.1	0.6851956	12243.20	1253	-295.32100
10	143 Haleyville	2019.2	0.7091902	10800.38	1338	580.27560
11	143 Haleyville	2019.3	0.7006980	13675.95		216
12	143 Haleyville	2019.4	0.6790068	12739.63		4
13	146 Bella Vista	2019.1	0.5858745	11043.31		

Let's go ahead and create that resids2 DataFrame, and let's look at that to verify what we've done. That looks correct. We can see the site name, quarter, Fuel_py1, total revenue, then the fitted revenue and the residuals, and we can look and compare these numbers to verify that that residual is the difference.

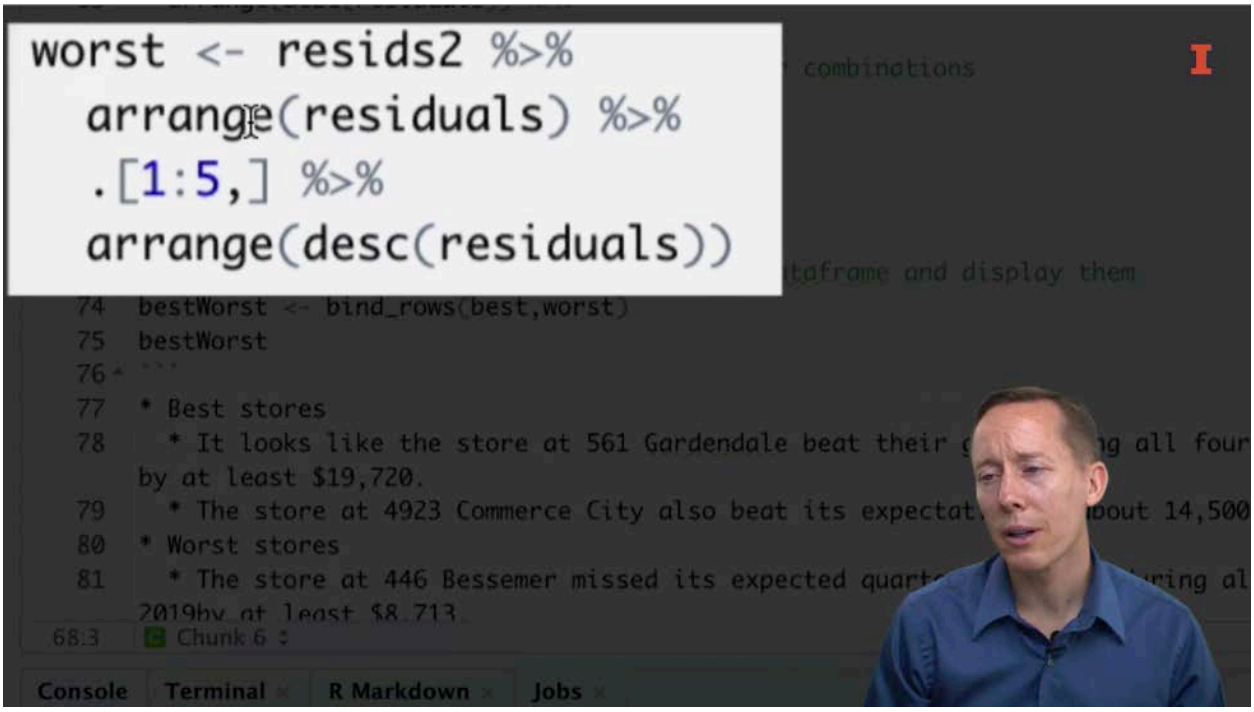


Now, let's take that resids2 DataFrame and only extract the five best stores and the five worst stores. The way we will do that is we will sort it first in descending order of

residuals and that will give us those observations that have the highest residuals, meaning they beat expectations by the most. We will only keep the first five rows.

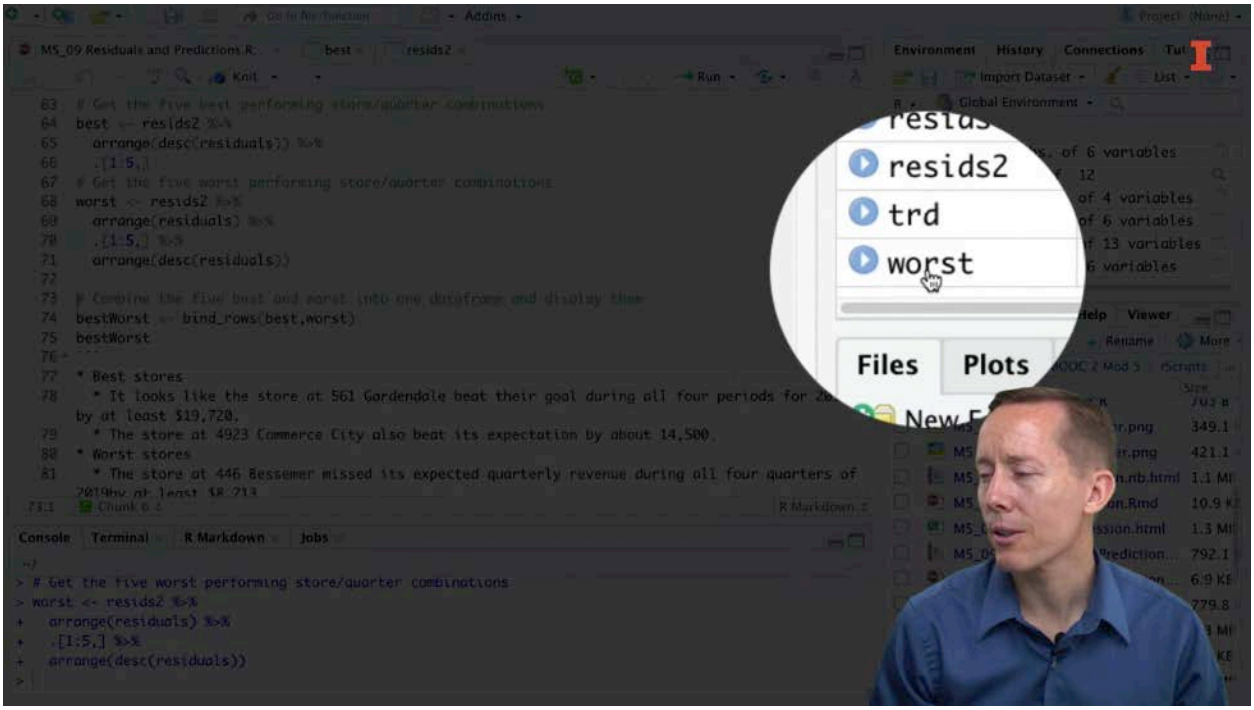


I'll do that and look at best here and there they are.

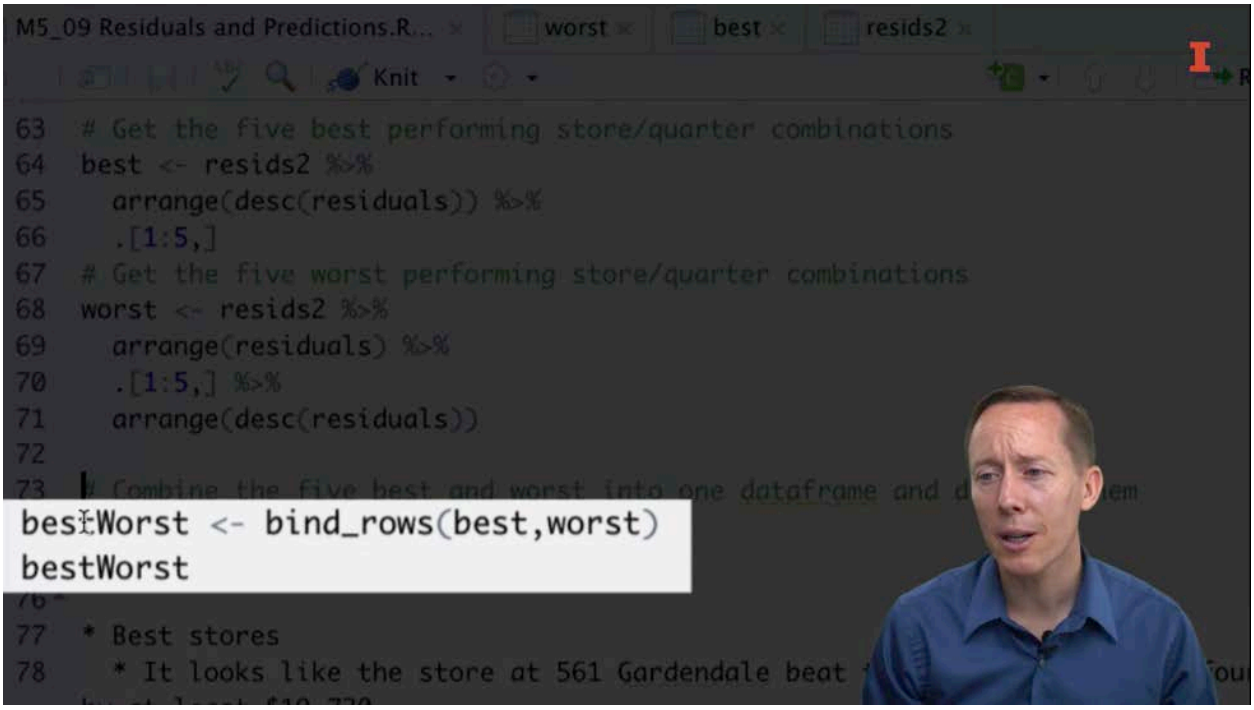


Then I will do a similar thing for the worst, but I will arrange the observations in ascending order of residual, so start with the lowest residuals. I'll keep only the first five

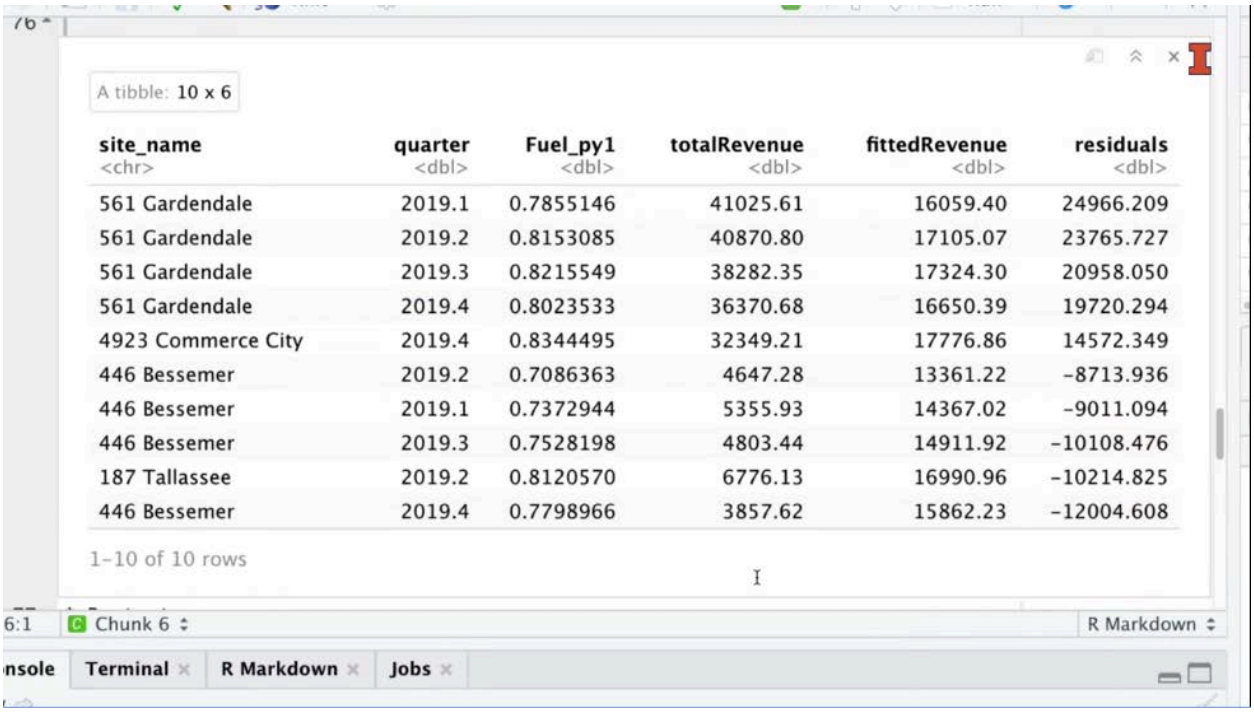
rows, and then I will sort them in descending order so that the worst observation is at the bottom.



There's the worst observations.



To make it easier to look at all of these observations at once, we will bind those rows together; best on top, worst on the bottom, and then display that DataFrame.

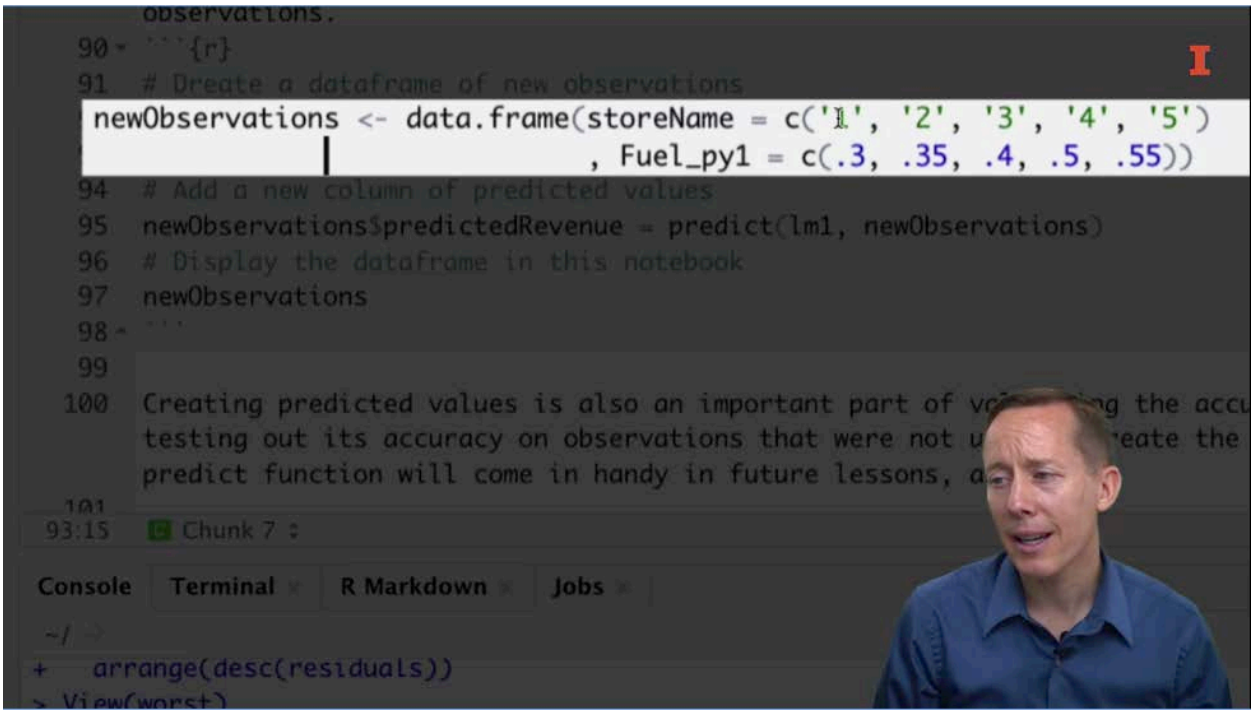


A tibble: 10 x 6

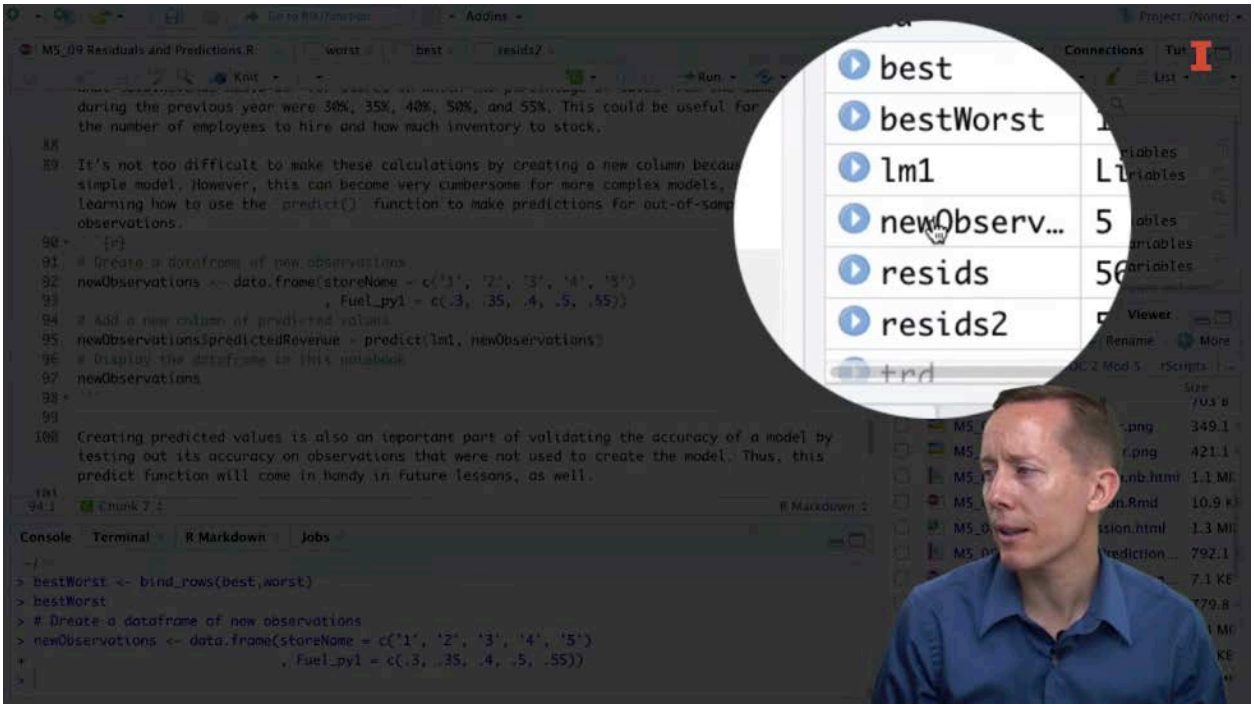
site_name <chr>	quarter <dbl>	Fuel_py1 <dbl>	totalRevenue <dbl>	fittedRevenue <dbl>	residuals <dbl>
561 Gardendale	2019.1	0.7855146	41025.61	16059.40	24966.209
561 Gardendale	2019.2	0.8153085	40870.80	17105.07	23765.727
561 Gardendale	2019.3	0.8215549	38282.35	17324.30	20958.050
561 Gardendale	2019.4	0.8023533	36370.68	16650.39	19720.294
4923 Commerce City	2019.4	0.8344495	32349.21	17776.86	14572.349
446 Bessemer	2019.2	0.7086363	4647.28	13361.22	-8713.936
446 Bessemer	2019.1	0.7372944	5355.93	14367.02	-9011.094
446 Bessemer	2019.3	0.7528198	4803.44	14911.92	-10108.476
187 Tallassee	2019.2	0.8120570	6776.13	16990.96	-10214.825
446 Bessemer	2019.4	0.7798966	3857.62	15862.23	-12004.608

1-10 of 10 rows

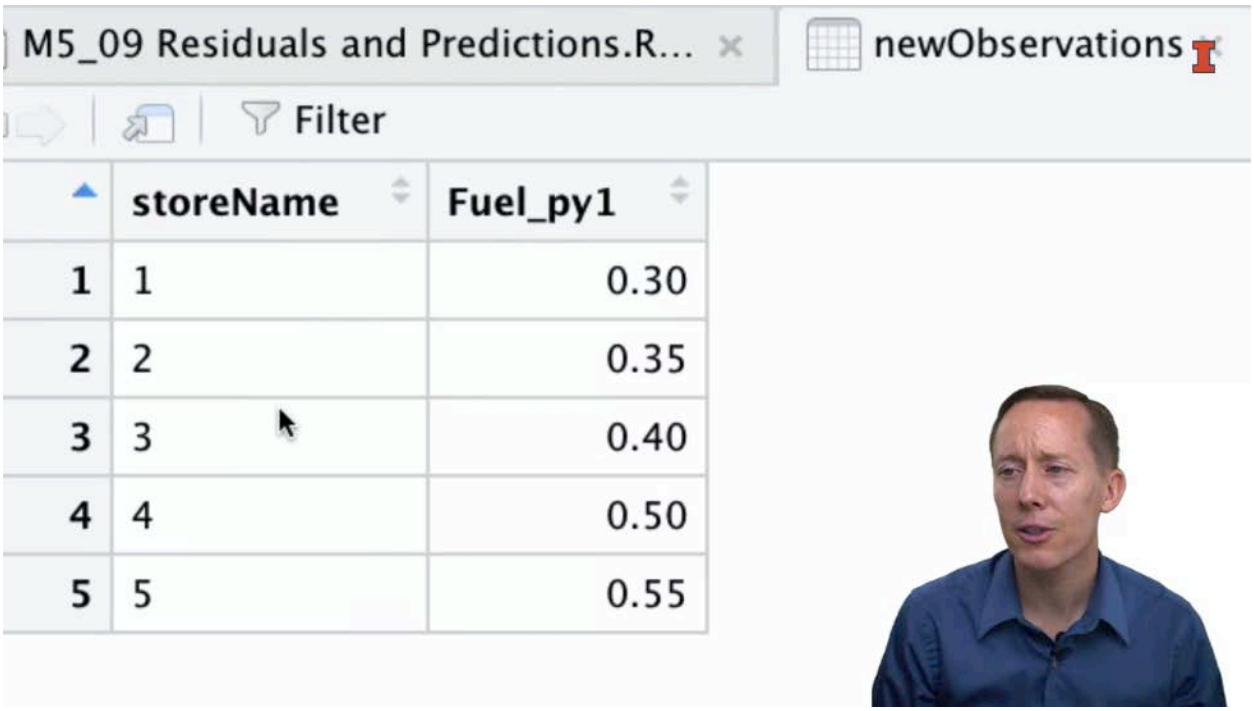
It looks like the store at 561 Gardendale beat their goal during all four periods for 2019 by at least $19,720. The store at 4923 Commerce City also beat its expectation by about 14,500. The store at 446 Bessemer missed its expected quarterly revenue during all four quarters of 2019 by at least $8,713, but it did get up to $12,000. Then the store at 187 Tallahassee missed its expected revenue by $10,214. Now, if I trust those expectations then as a manager, I may want to look into the two stores that underperformed during 2019 to work on improving their performance. In contrast, I may want to look at the two stores that outperformed during 2019 to find out if their best practices can be replicated in other locations. This linear model can also be used to predict future values. Let's say that we want to find out what total revenue would be for stores in which the percentage of sales from the same quarter during the previous year were 30 percent, 35 percent, 40 percent, 50 percent, and 55 percent. This could be useful for planning the number of employees to hire and how much inventory to stock. While it's not too difficult to make these calculations by hand or by creating a new DataFrame, this can be very cumbersome for more complex models. So it's worth learning how to use the predict function to make predictions for out-of-sample observations.



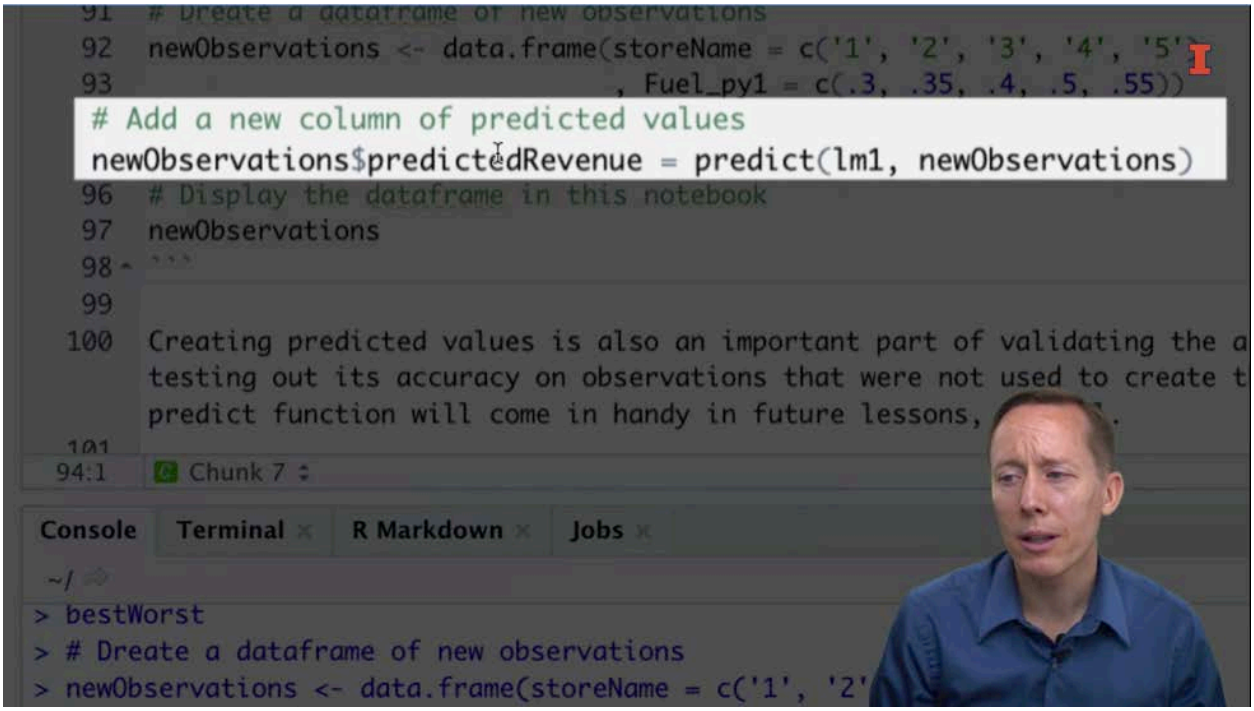
The first thing I'm going to do is I'm going to create a DataFrame, newObservations, that has five fictitious store names numbered 1-5 and then those percentages for Fuel_py1 that I mentioned.



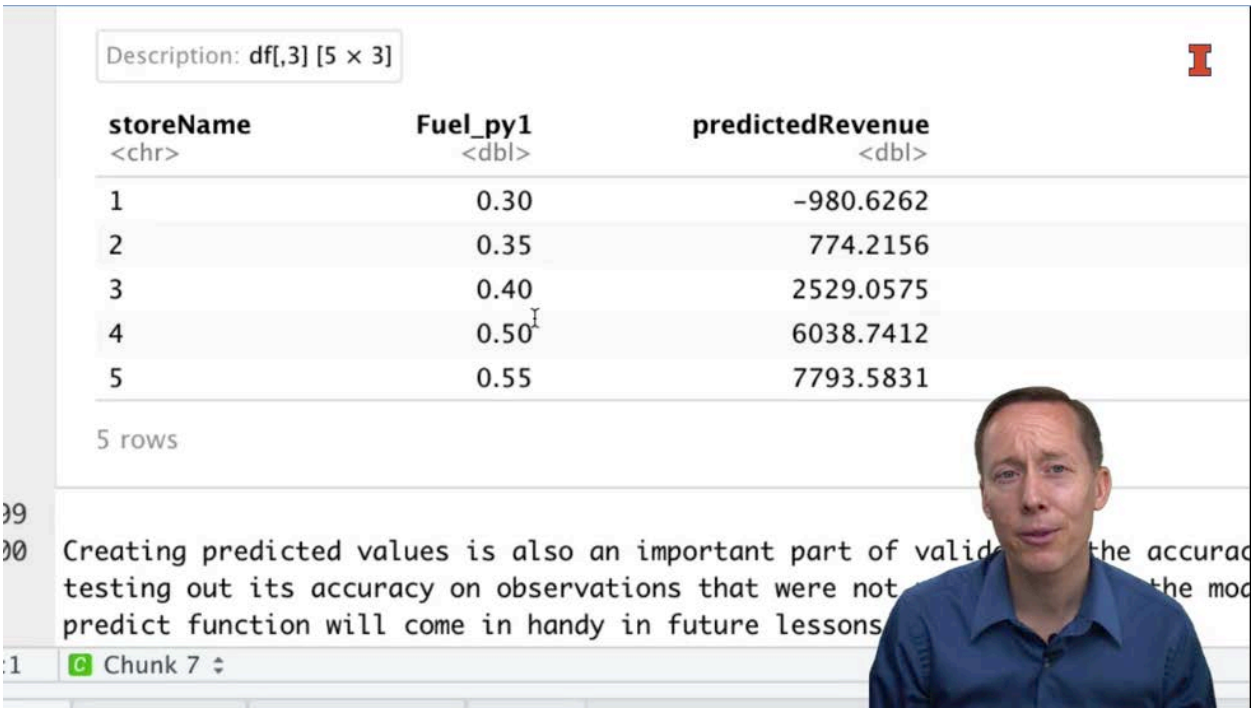
I'll go ahead and create that DataFrame and let's look at it.



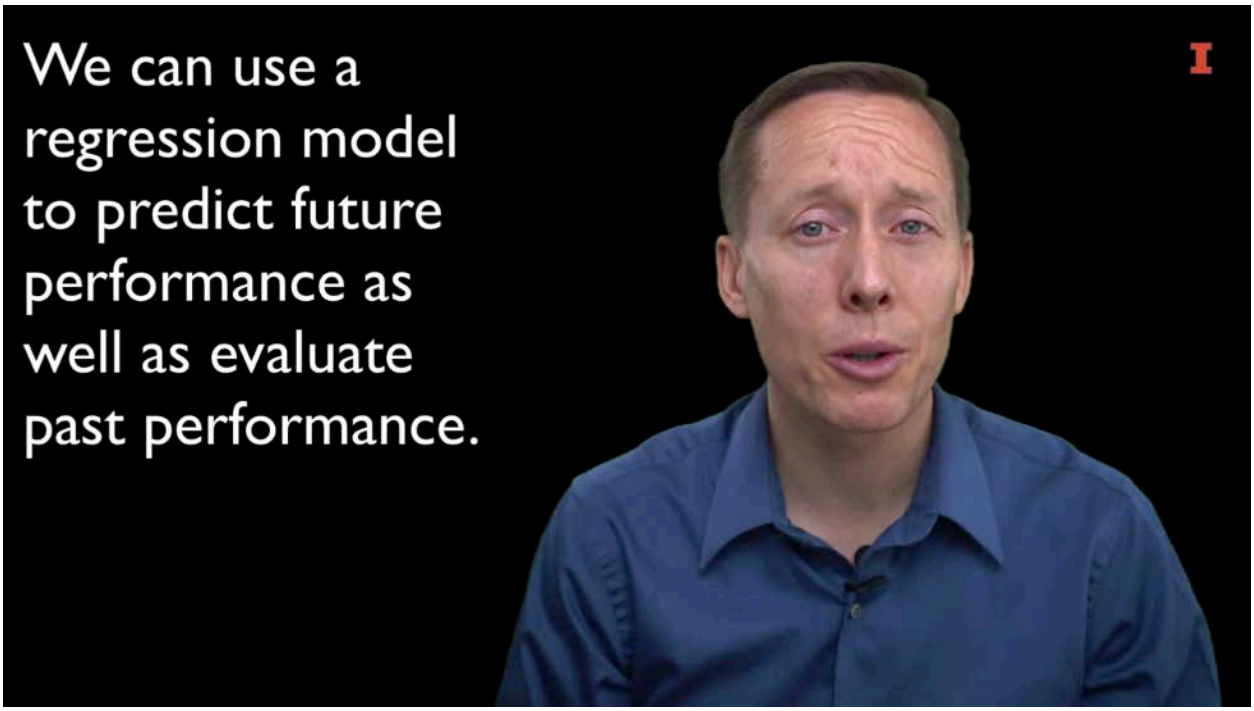
A very simple DataFrame.



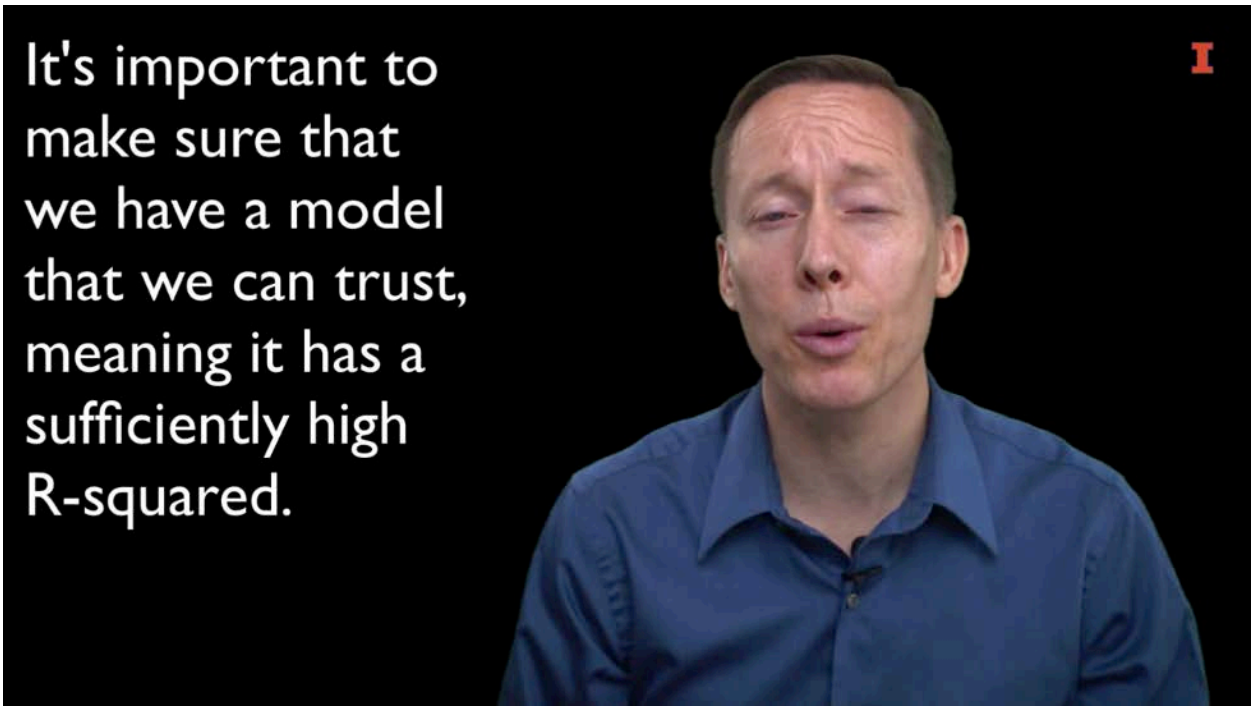
Then I'm going to add a new column of predicted values to that DataFrame. This is the key, I'm going to take the predict function, I'm going to base it off of my linear model 1 object, and I'm going to make the predictions using the values in this newObservations DataFrame.



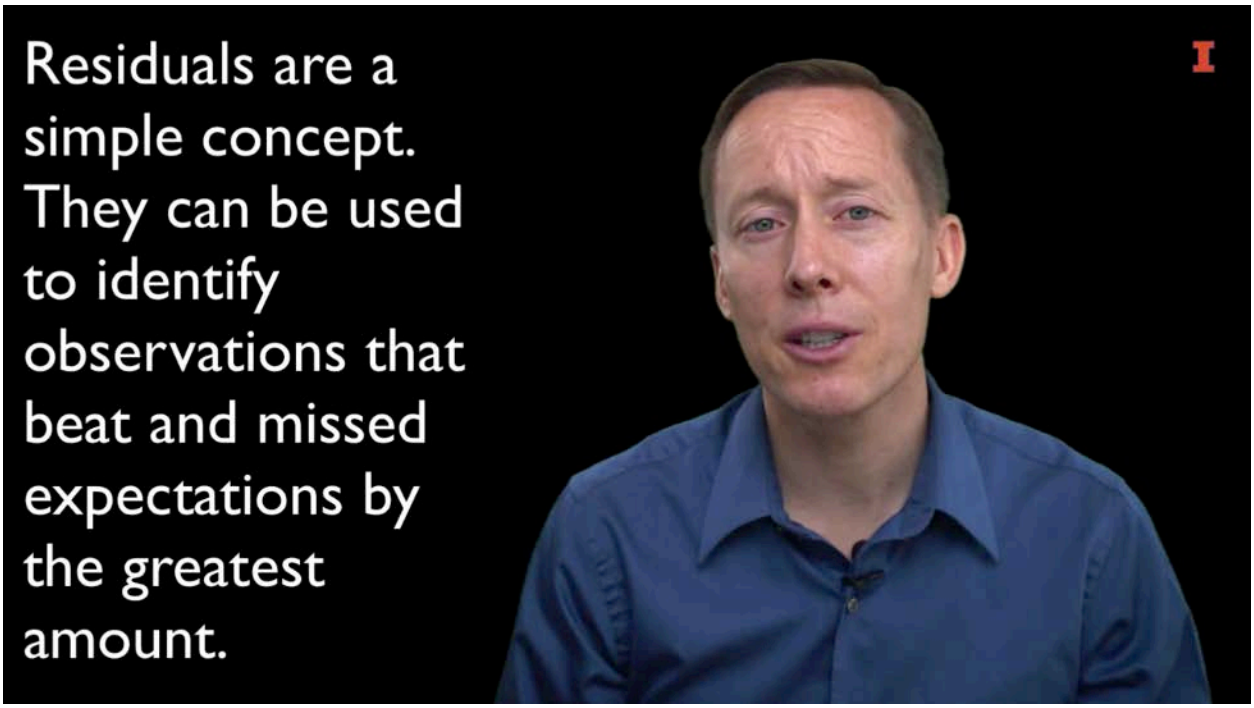
Now, let's go ahead and do that and then explore this DataFrame. We can see that I've got these predictions very quickly for those five different levels of Fuel_py1. Creating predicted values is also an important part of validating the accuracy of a model by testing out its accuracy on observations that were not used to create the model. Thus this predict function will come in really handy in future lessons as well.



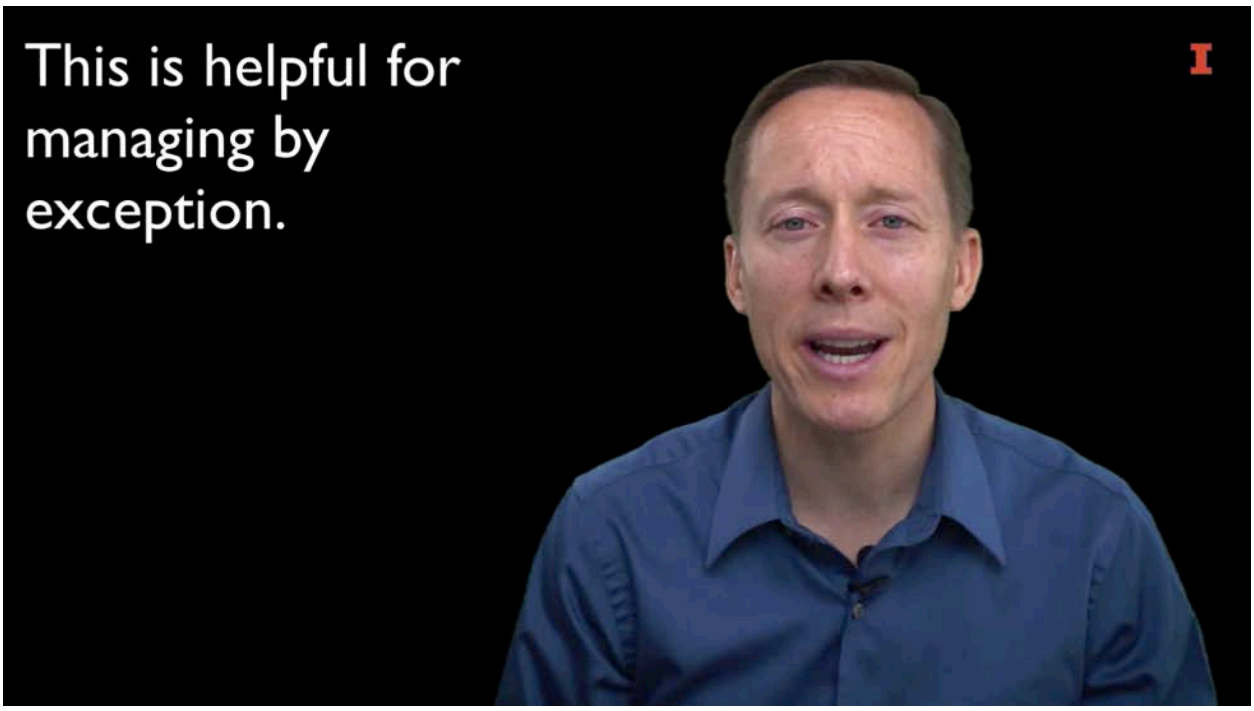
In conclusion, we can use a regression model to predict future performance as well as evaluate past performance.



However, when we do that, it's important to make sure that we have a model that we can trust, meaning it has a sufficiently high R-squared.

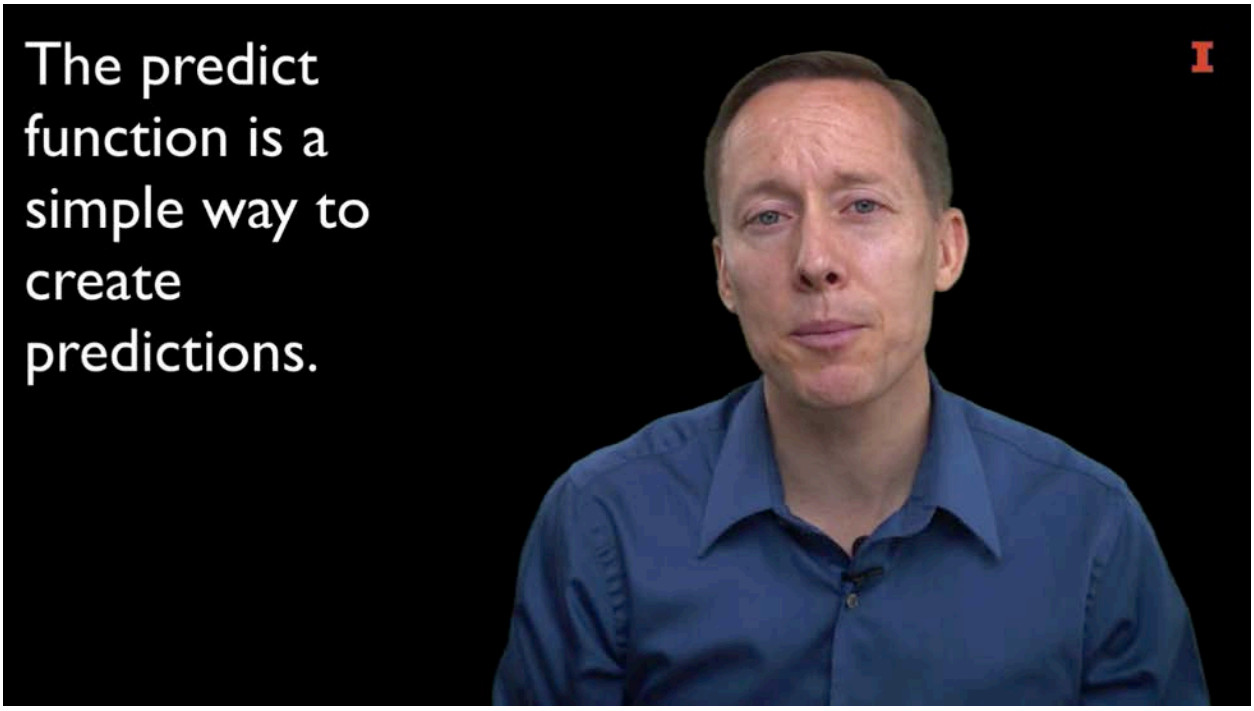


Residuals are a simple concept and can be used to identify observations that beat and missed expectations by the greatest amount.



This is helpful for managing by exception.

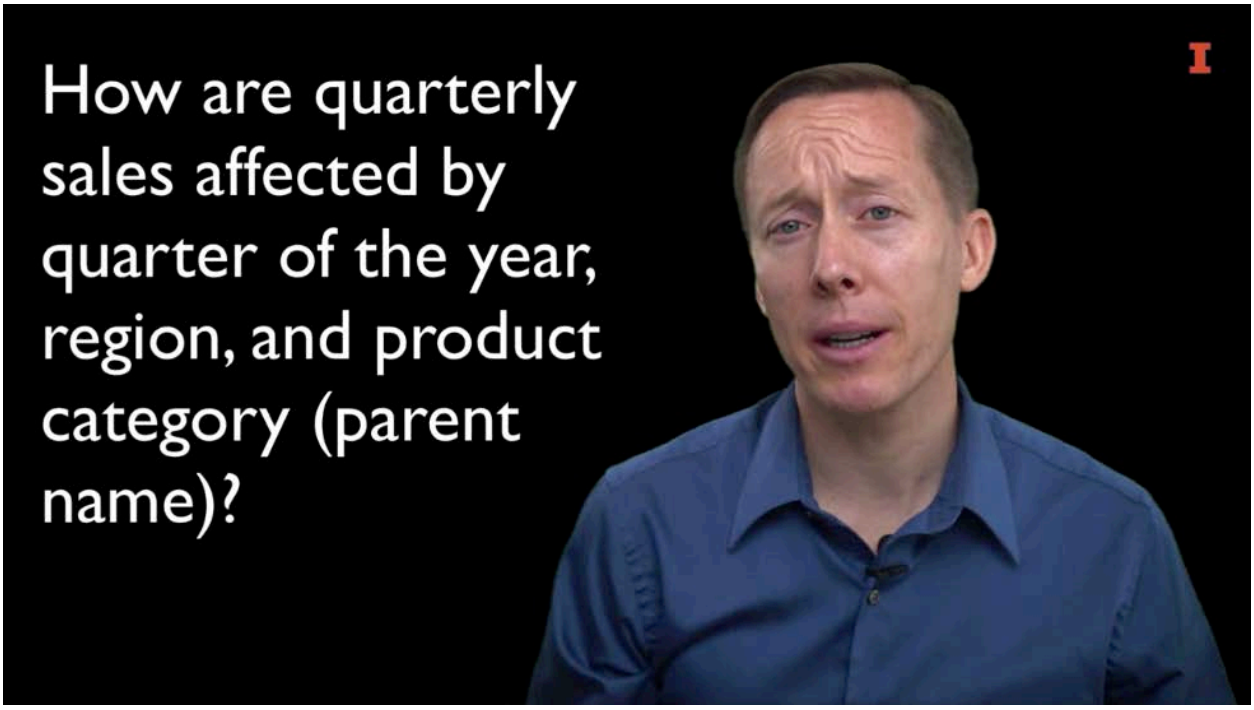
Making predictions helps forecast future performance and improve plans.



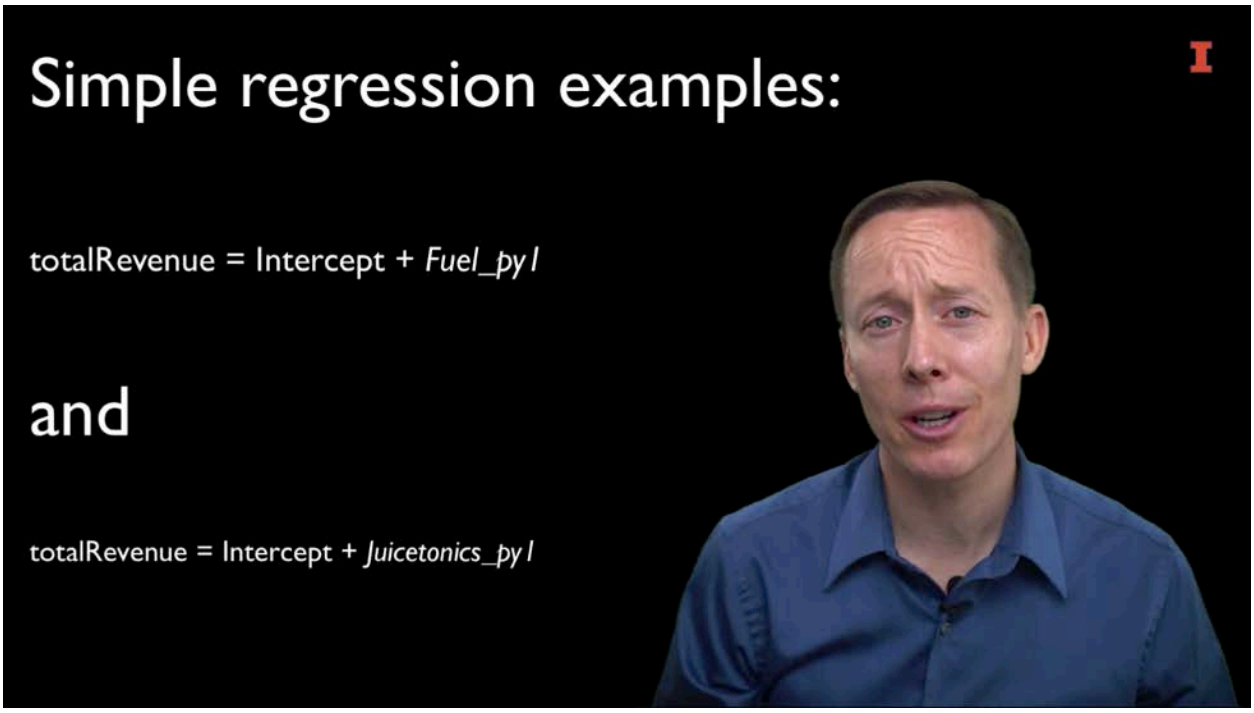
The predict function is a simple way to create these predictions.

This function will also come in handy for validating model accuracy using out-of-sample data in future lessons.

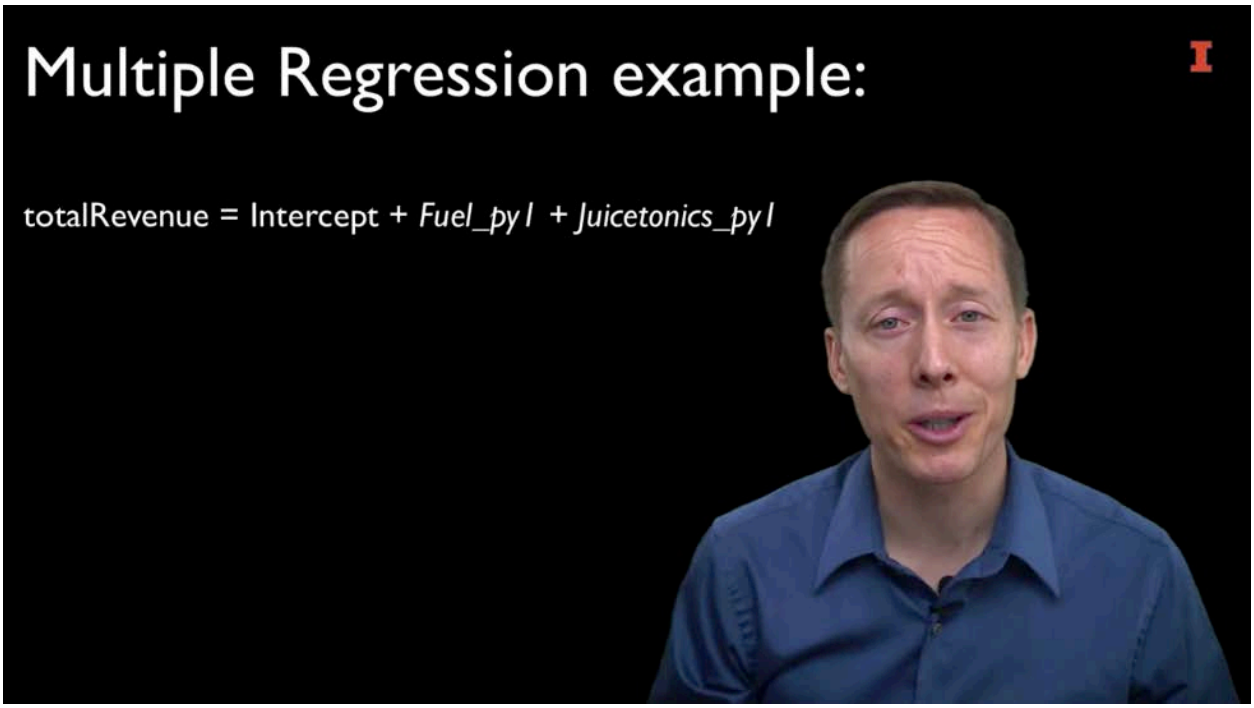
Lesson 1-4.5 Multiple Regression



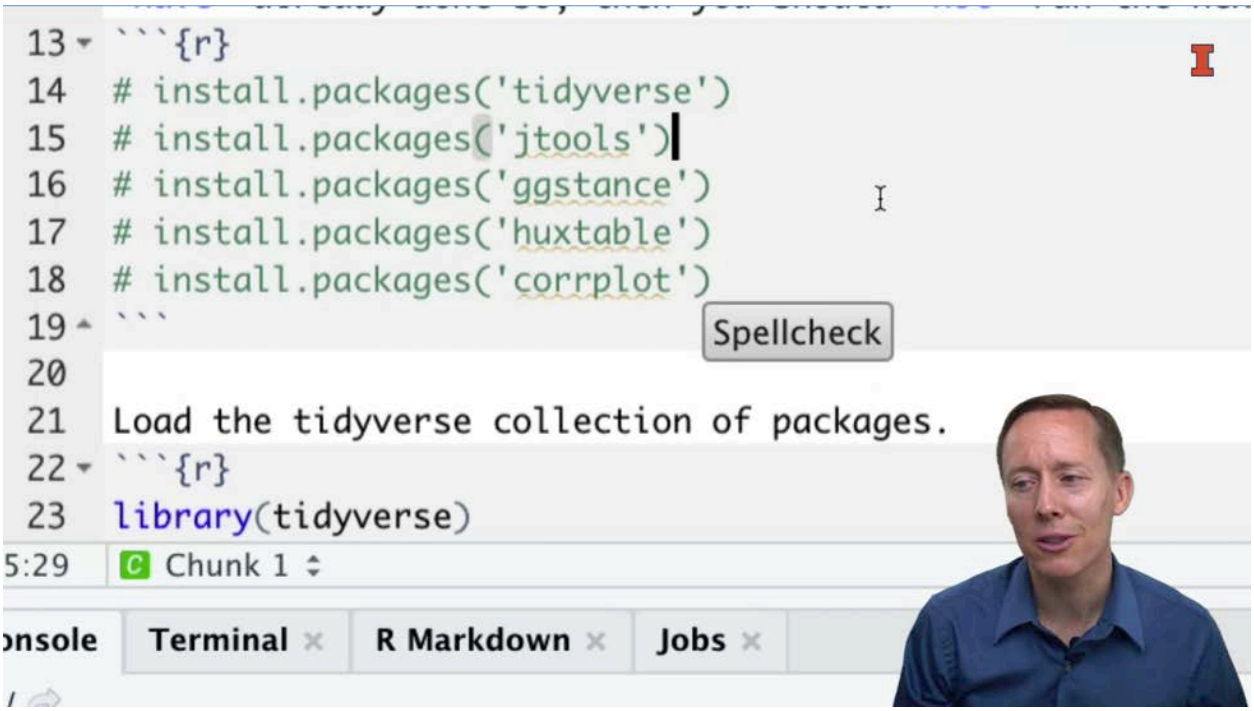
Recall that are broad business question is how are a quarterly sales affected by quarter of the year, region and by product category?



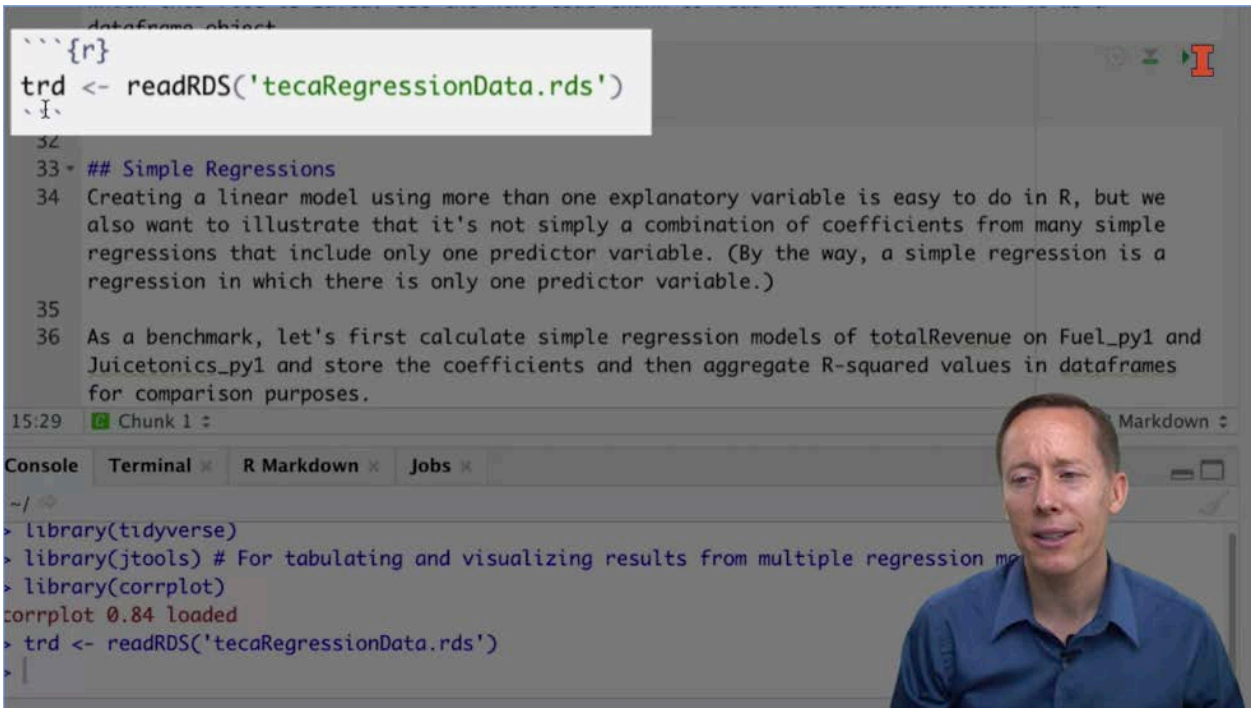
Up to this point, we have analyzed the ability of predictor variables to create forecasts of quarterly revenue independently of each other. In other words, we have used one predictor variable in a regression model. This is known as simple regression.



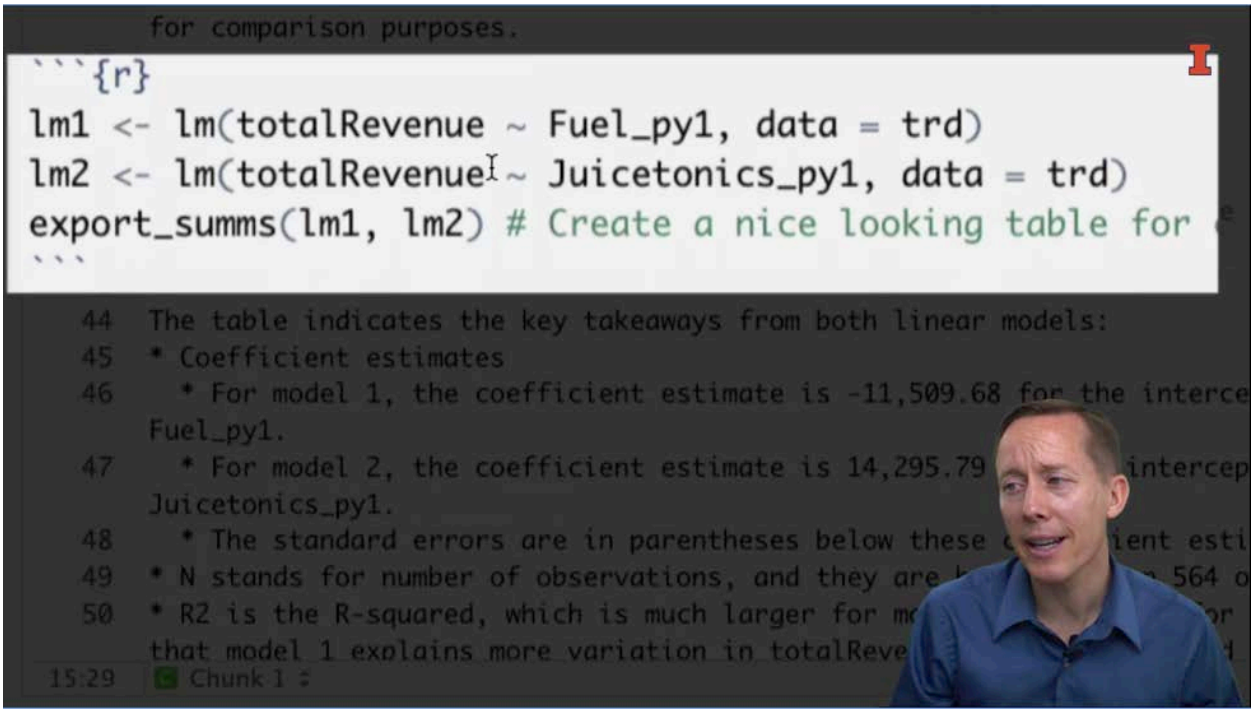
In this lesson, we will investigate the benefits of using multiple variables in the same linear model to create those forecasts. When we do that, it's known as multiple regression.



Now if you want to follow along then go ahead and download the associated r Markdown file and you'll need to install several packages. So first we have the tidyverse collection of packages and then the jtools package which we will use for tabulating regression results in an easy to read format. And that package depends on the ggstance, huxtable packages. And then we'll also use corrplot package as well.

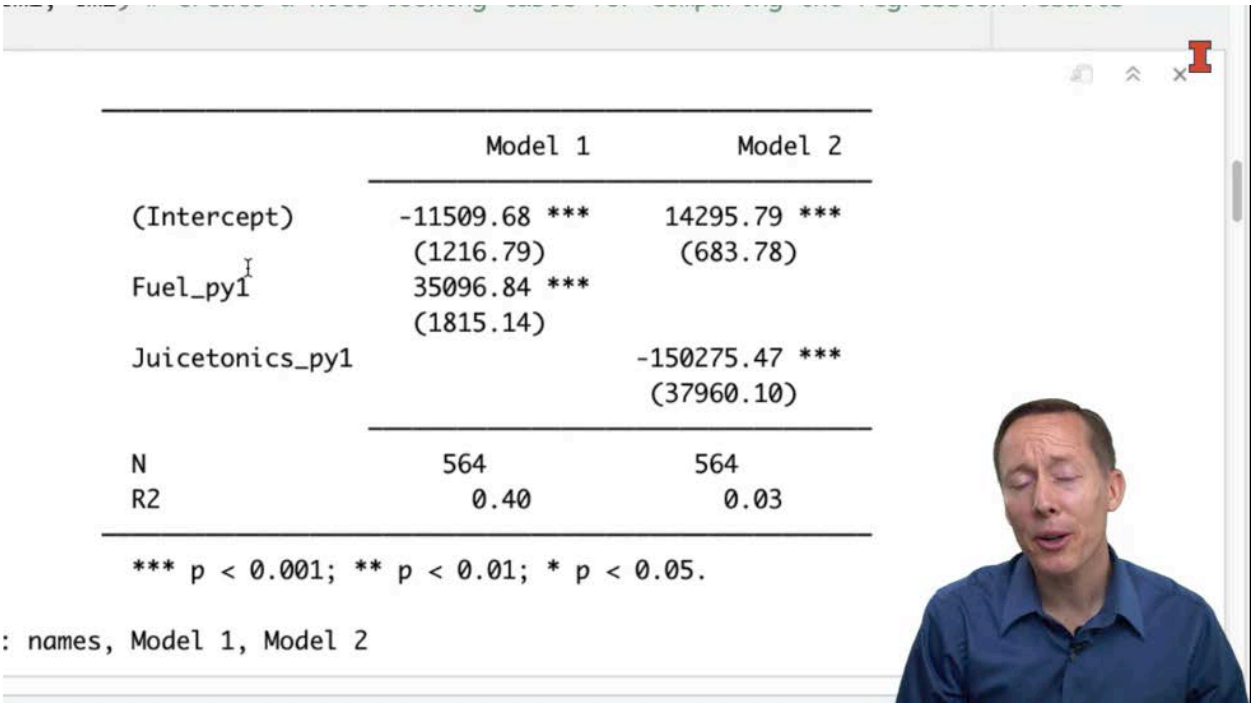


Once you've installed those packages go ahead and make sure to load those packages and also make sure to read in the tecaRegressionData.rds file and save it as a data frame called trd.



Now creating a linear model using more than one explanatory variable is easy to do in r. But we also want to illustrate that it's not simply a combination of coefficients from many simple regressions that include only one predictor variable. So as a benchmark, let's

first calculate simple regression models of total revenue on fuel_py1 and juicetonics_py1. And then store those coefficients and aggregate the r squared values and data frames for comparison purposes. So Lm1 will be a regression of totalRevenue on Fuel_py1and Lm2 will be a regression of totalRevenue on juicetonics_py1 and then we're going to use the J tools package to summarize that information.

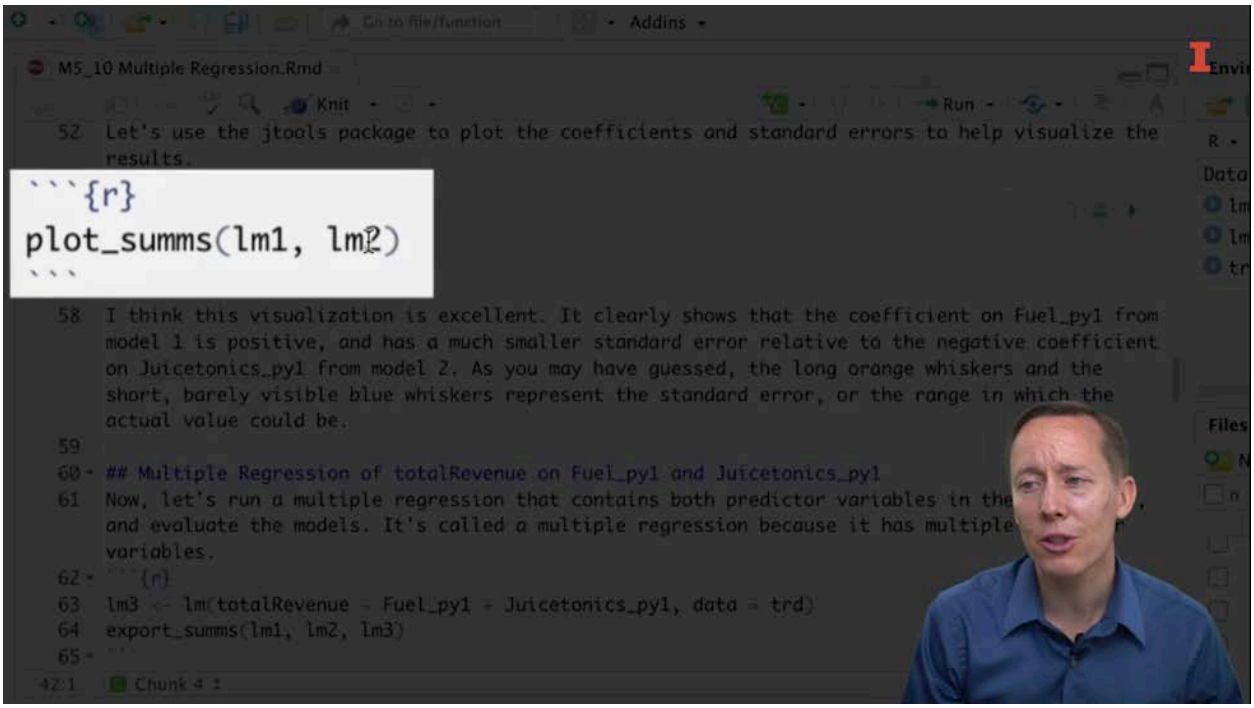


	Model 1	Model 2
(Intercept)	-11509.68 ***	14295.79 ***
	(1216.79)	(683.78)
Fuel_py1	35096.84 ***	
	(1815.14)	
Juicetonics_py1		-150275.47 ***
		(37960.10)
N	564	564
R2	0.40	0.03

*** p < 0.001; ** p < 0.01; * p < 0.05.

: names, Model 1, Model 2

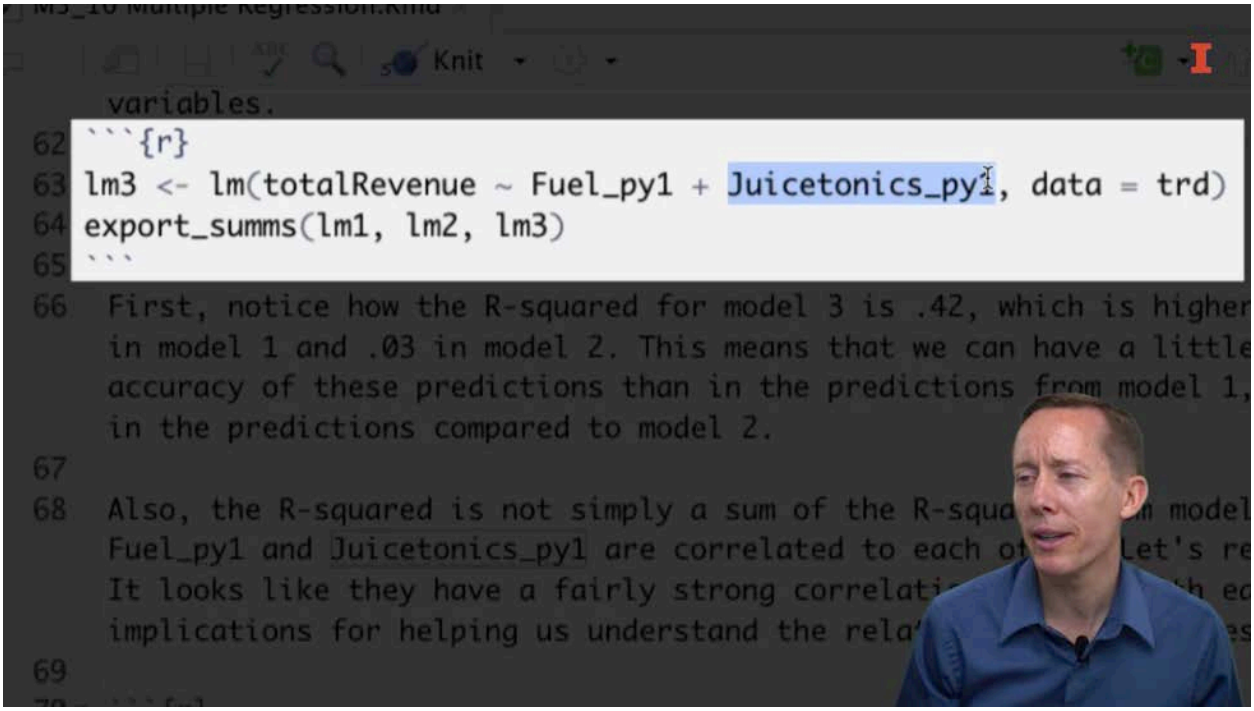
All right, this table indicates the key takeaways from both linear models. For Model 1 the coefficient estimate is negative 11509.68 for the intercept and is 35096.84 for fuel_py1. For model to the coefficient estimate is 14,295.79 for the intercept and is negative 150,275.47 for juicetonics_py1. The standard errors are in parentheses below these coefficient estimates. N stands for the number of observations and they are both based on 564 observations. R2 is the R squared which is much larger for model 1, 40% and for model 2 which is just 3%. This means that model 1 explains more variation in total revenue than model 2 and is better for making predictions. Let's use the J tools package to plot the coefficients and standard errors to help visualize these results.



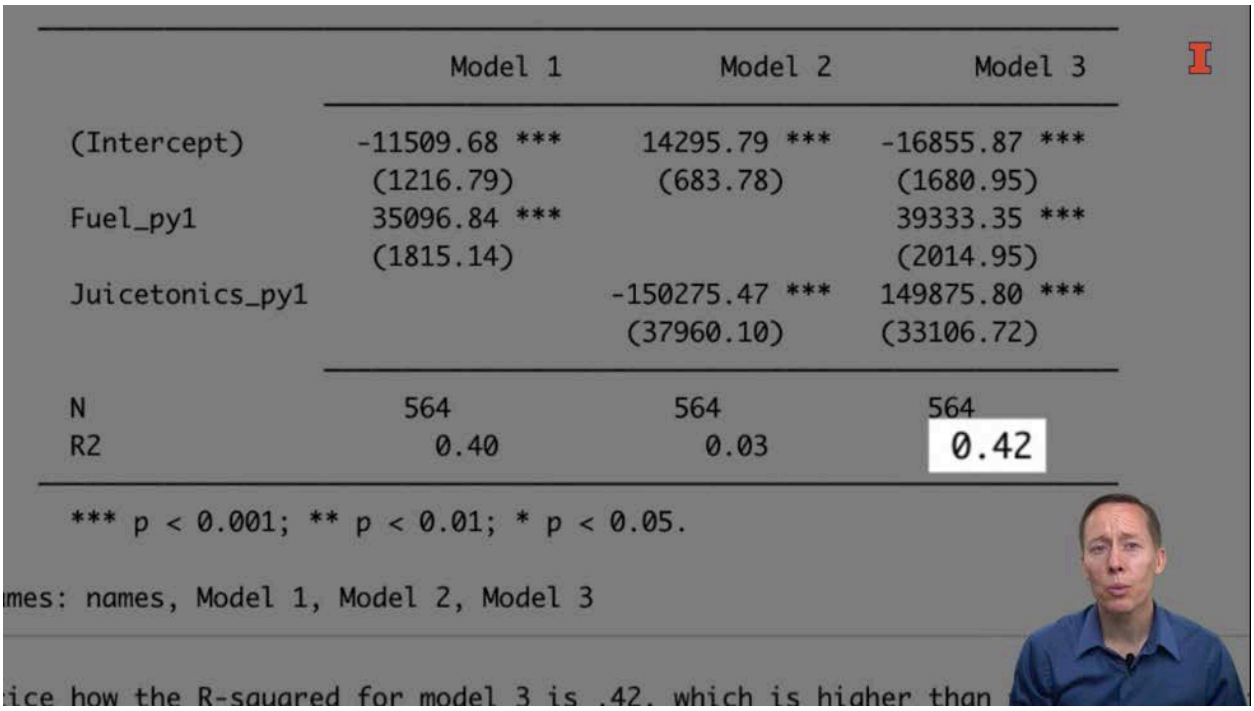
I will use this plot sums function on the Lm1 and Lm2 objects.

I think this visualization is excellent. It clearly shows that the coefficient on fuel_py1 from model 1 is positive and has a much smaller standard error relative to the negative coefficient on Juicetonics_1 from model 2. As you may have guessed the long orange whiskers and the short barely visible blue whiskers represent the standard error or the

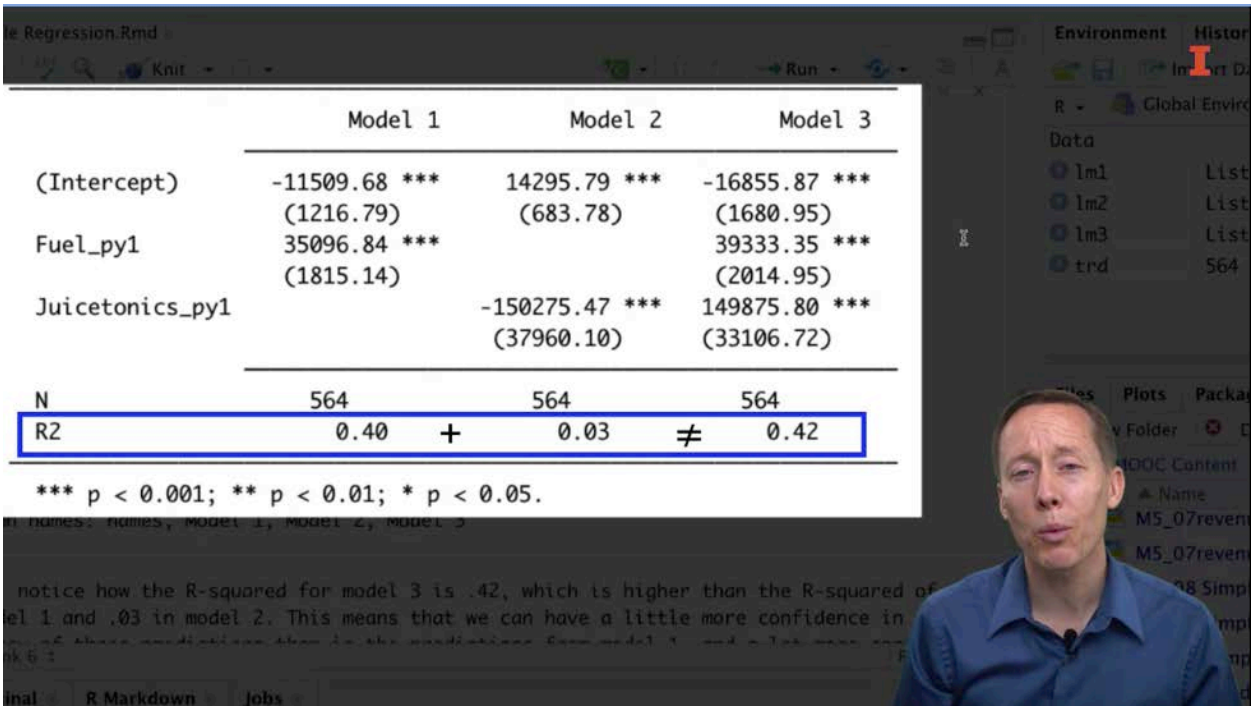
range in which the actual value could be. Now let's run a multiple regression that contains both predictor variables in the same model and evaluate all 3 models.



So, I'm going to run this model right here. That has Fuel_py1 and Juicetonics_py1 in it and then we'll tabulate those results.

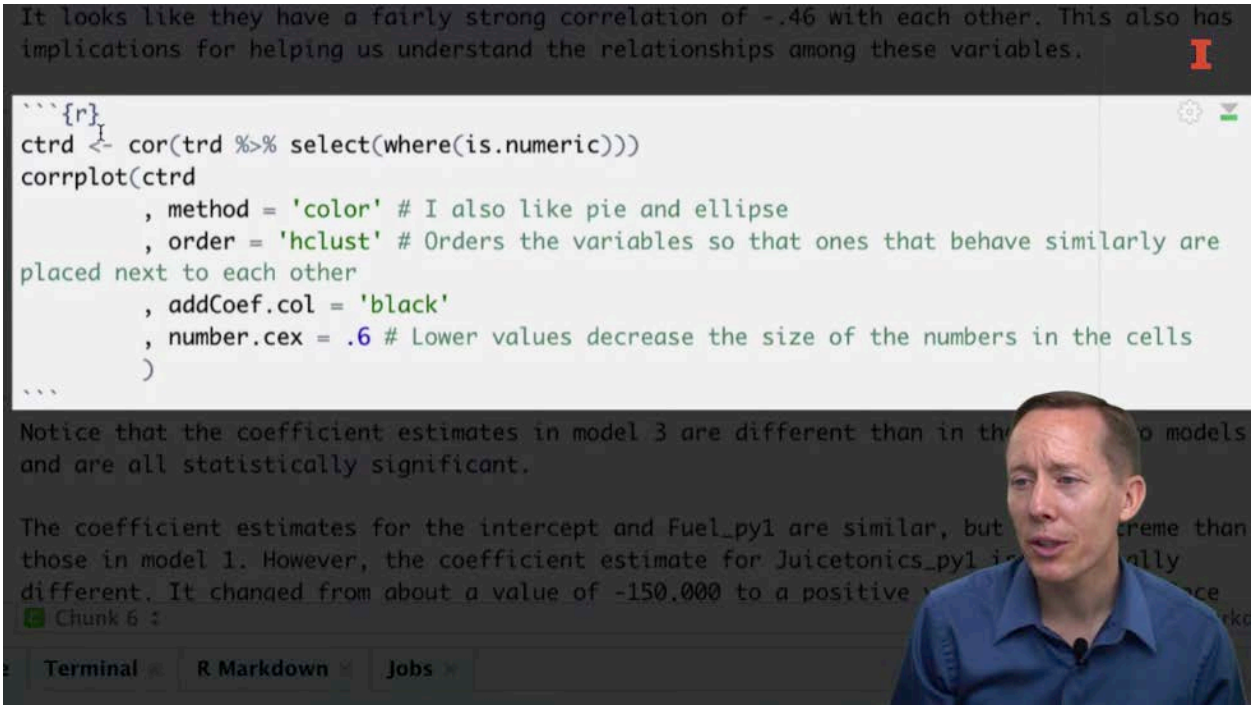


Okay, first notice how the R squared from model three is 0.42, which is higher than the R squared of 0.40 in model 1 and then 0.03 in model 2.

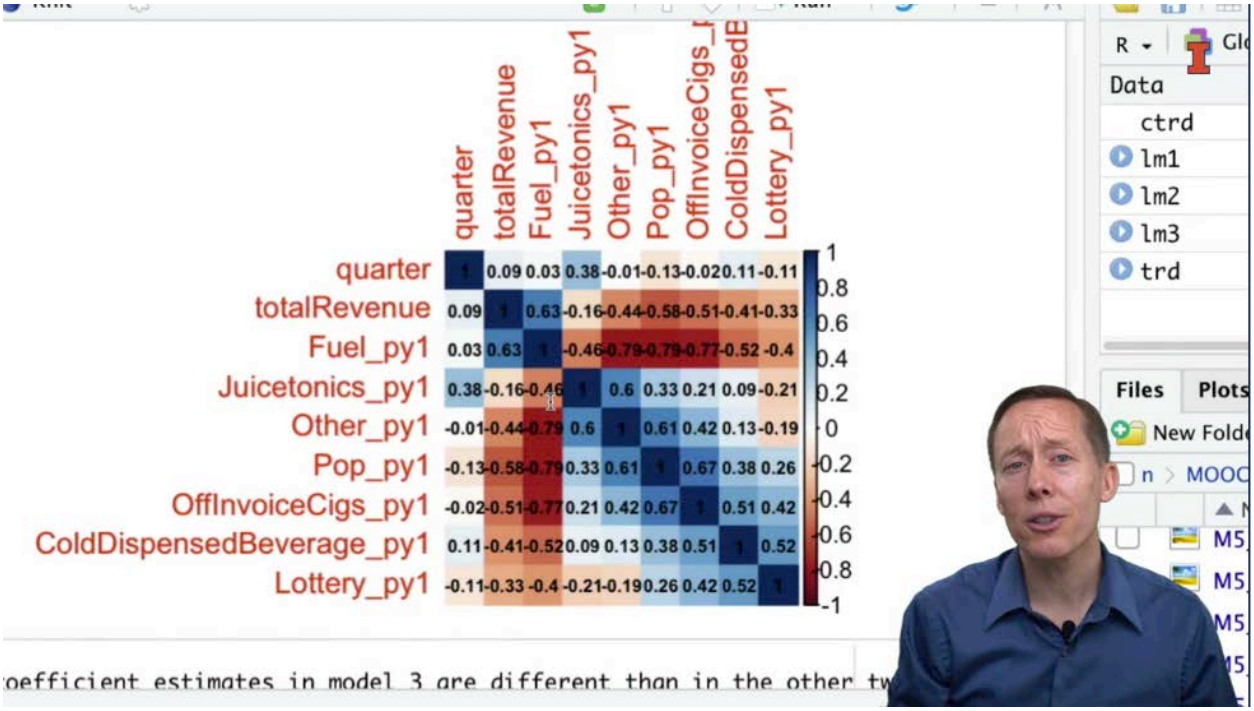


This means that we can have a little more confidence in the accuracy of these predictions than in the predictions from model 1 ,and a lot more confidence in the predictions compared to the predictions from model 2. Also the R squared is not simply

as some of the R squared from models 1 and 2. This is because fuel_py1 and Juicetonics_py1 are correlated with each other. Let's review the correlation matrix.

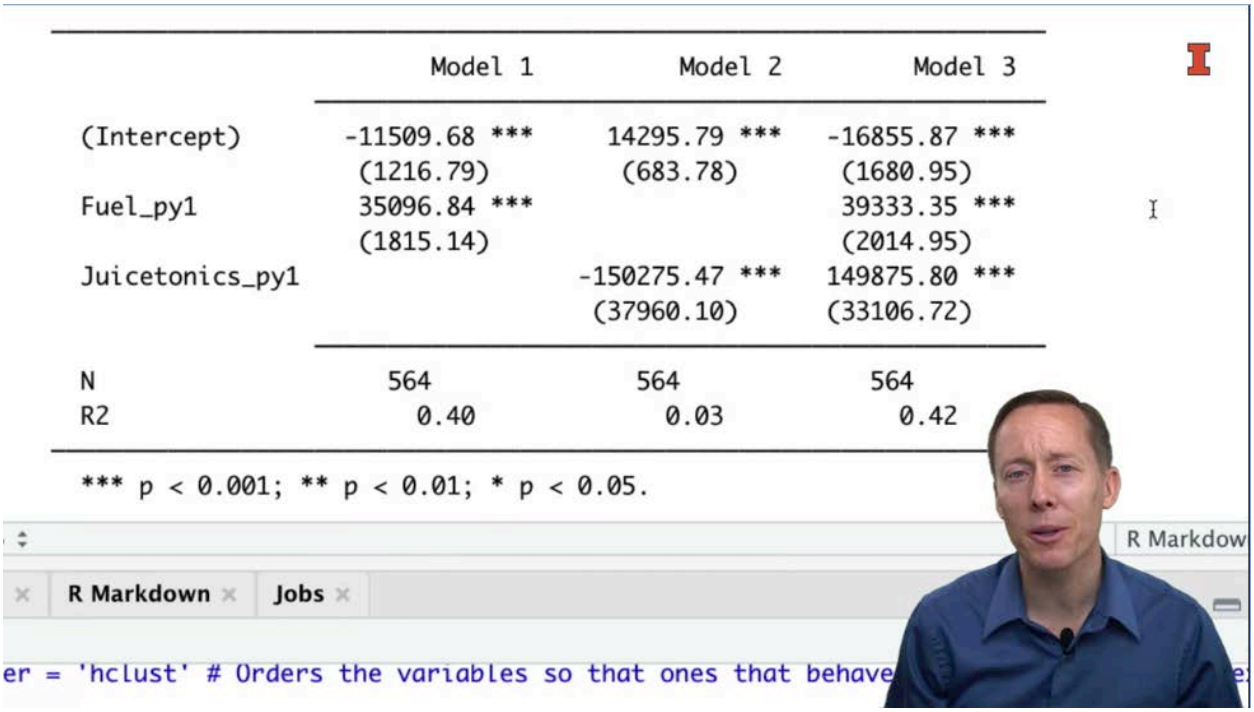


So I'm going to create this correlation plot,



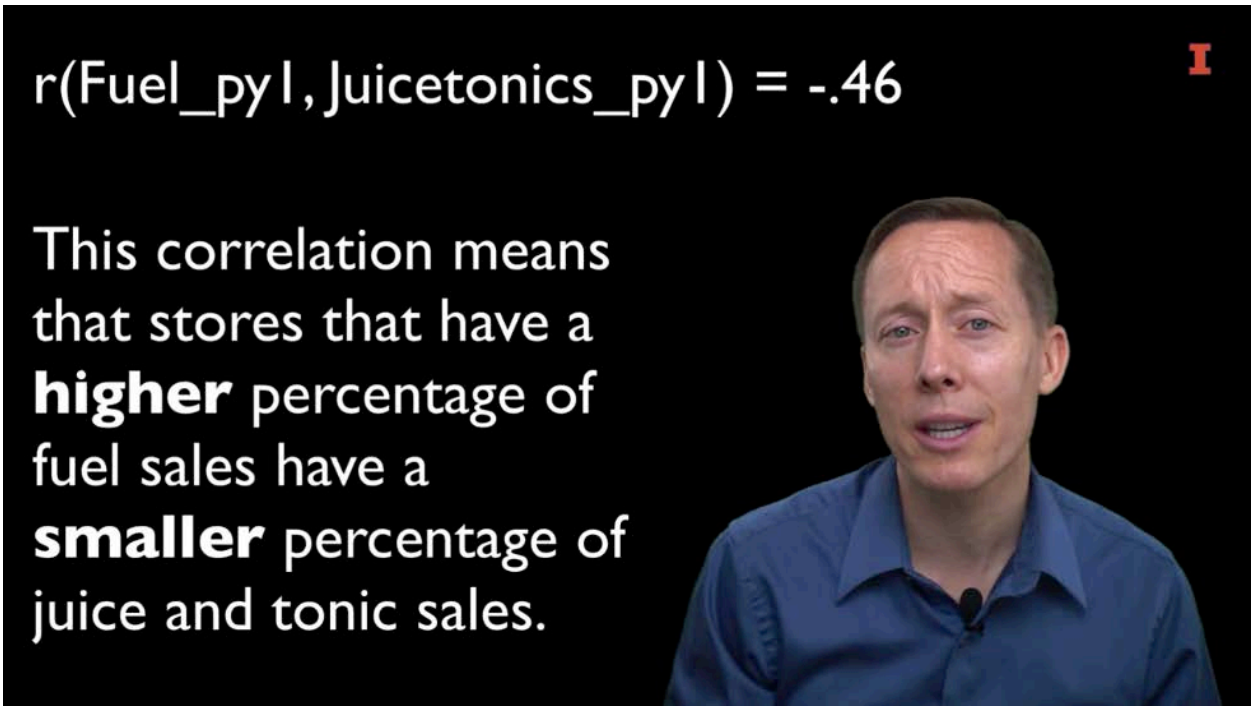
And let's look at the correlation between Fuel_py1 and Juicetonics_py1. It's negative 0.46, so that's a pretty strong negative correlation that they have with each other. This

also has implications for helping us understand the relationships among all 3 of these variables.

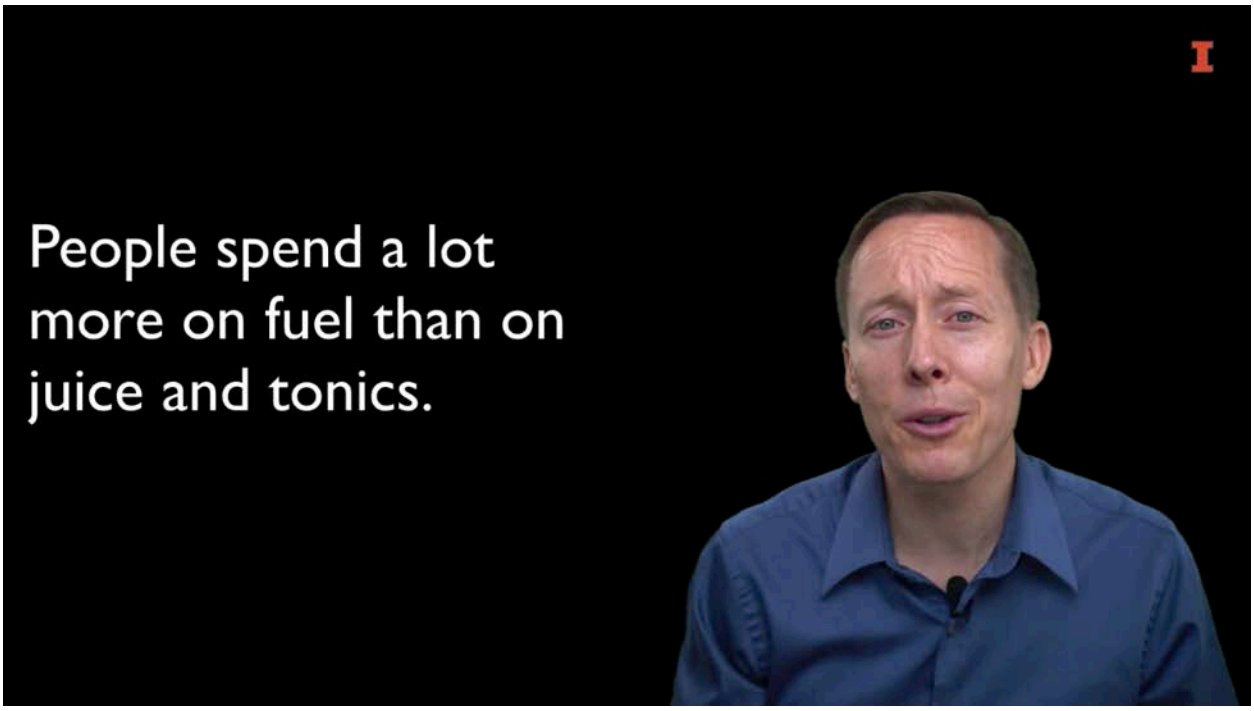


Notice that the coefficient estimates and model 3 are different than in the other 2 models and are all statistically significant. The coefficient estimates for the intercept and Fuel_py1 are similar but more extreme than those in model 1. However, the coefficient estimate for Juicetonics_py1 is drastically different. It changed from about the value of negative 150,000 to a positive value of 150,000. Once again, this is because Fuel_py1 and Juicetonics_py1 are correlated with each other. And the coefficient estimates represent the unique effect of each predictor variable, after considering the effects of the other predictor variables.

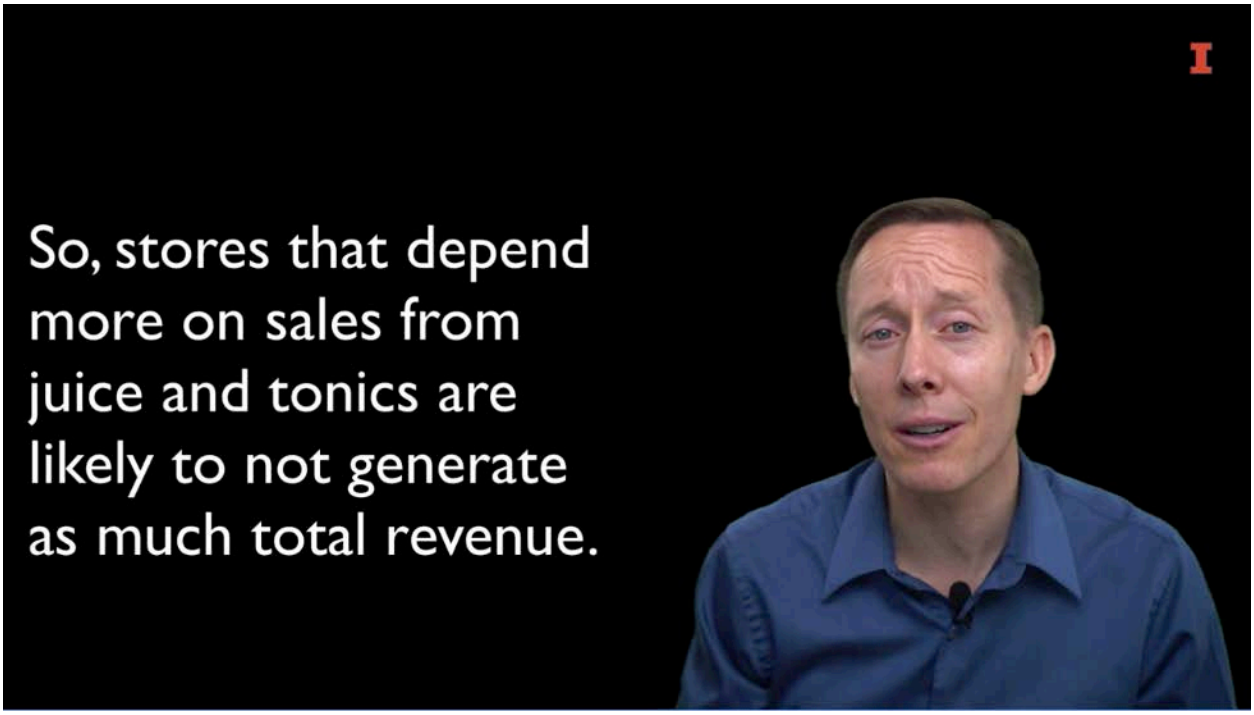
This change is really effectively summarized by creating a plot from the J tools package. Conceptually this change and the Juicetonics_py1 coefficient makes sense as well.



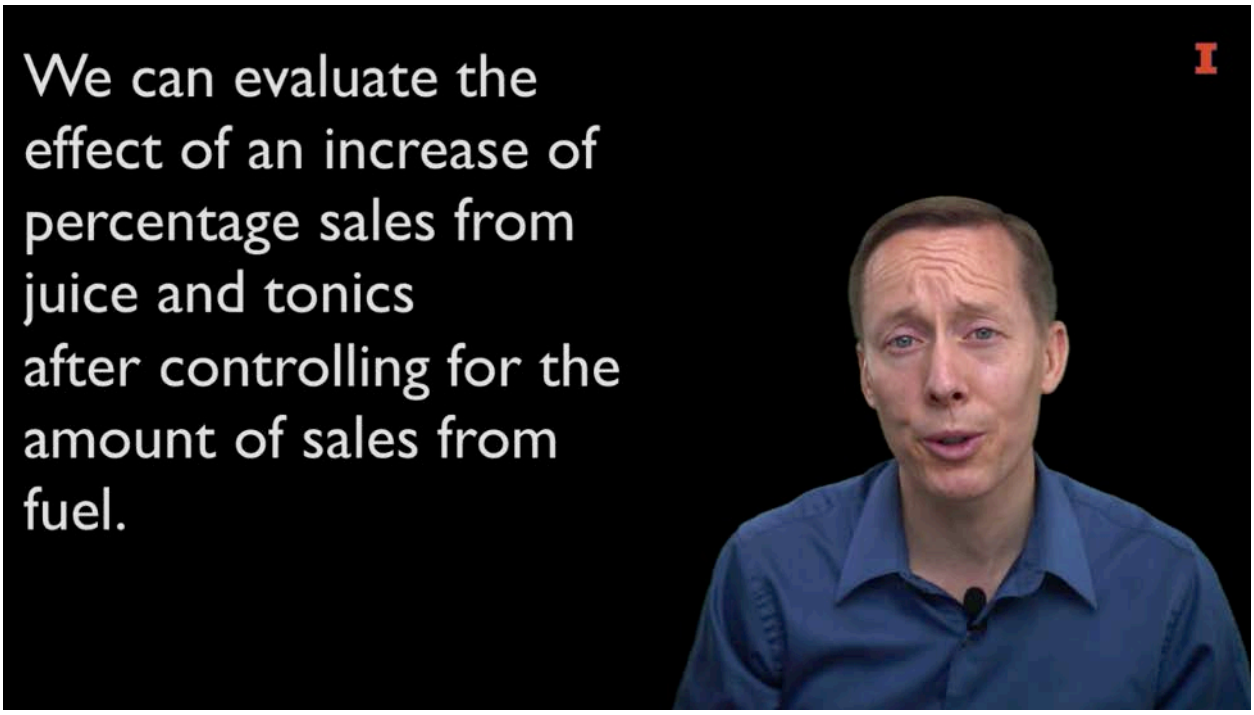
Specifically the negative correlation between Fuel_py1 and Juicetonics_py1 means that stores that have higher percentages of fuel sales have a smaller percentage of juice and tonic sales.



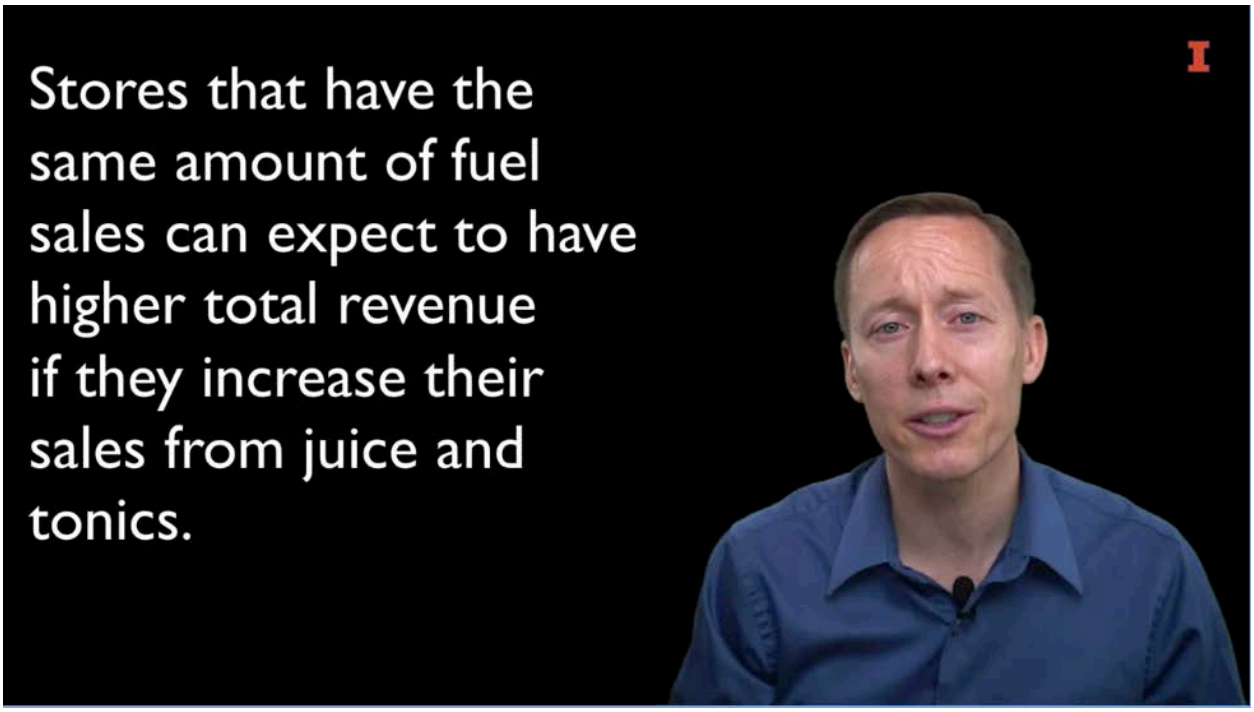
People spend a lot more on fuel than on juice and tonic.



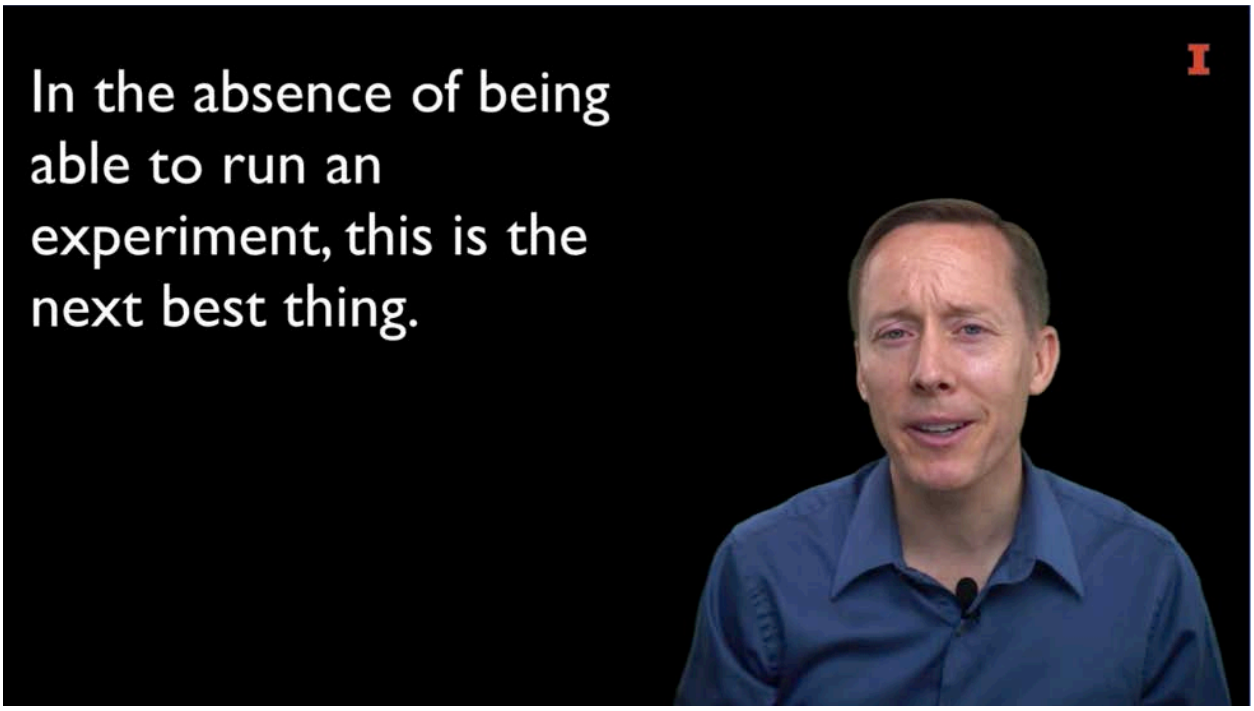
So stores that depend more on sales of juice and tonics are likely to not generate as much total revenue.



So the really great thing for explanatory purposes is that we can evaluate the effect of an increase of percentage sales from juice and tonics after controlling for the amount of sales from fuel.

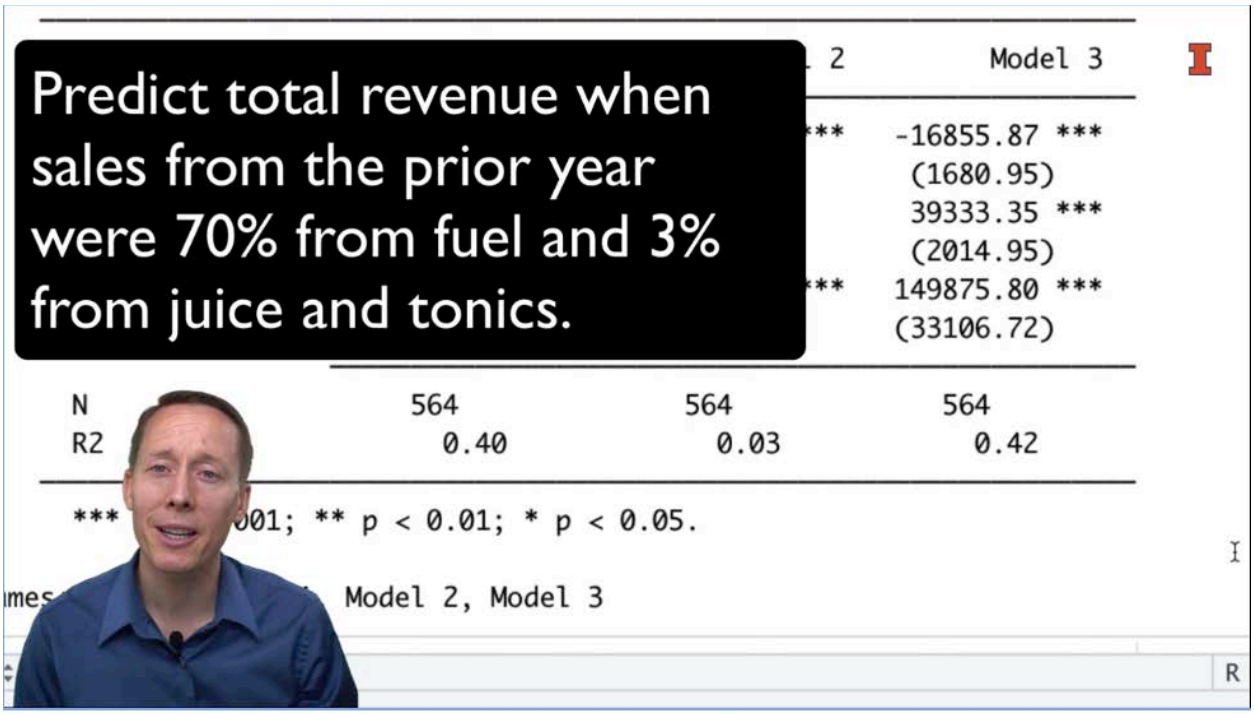


In other words, stores that have the same amount of fuel sales can expect to have higher total revenue if they increase their sales from juice and tonics.

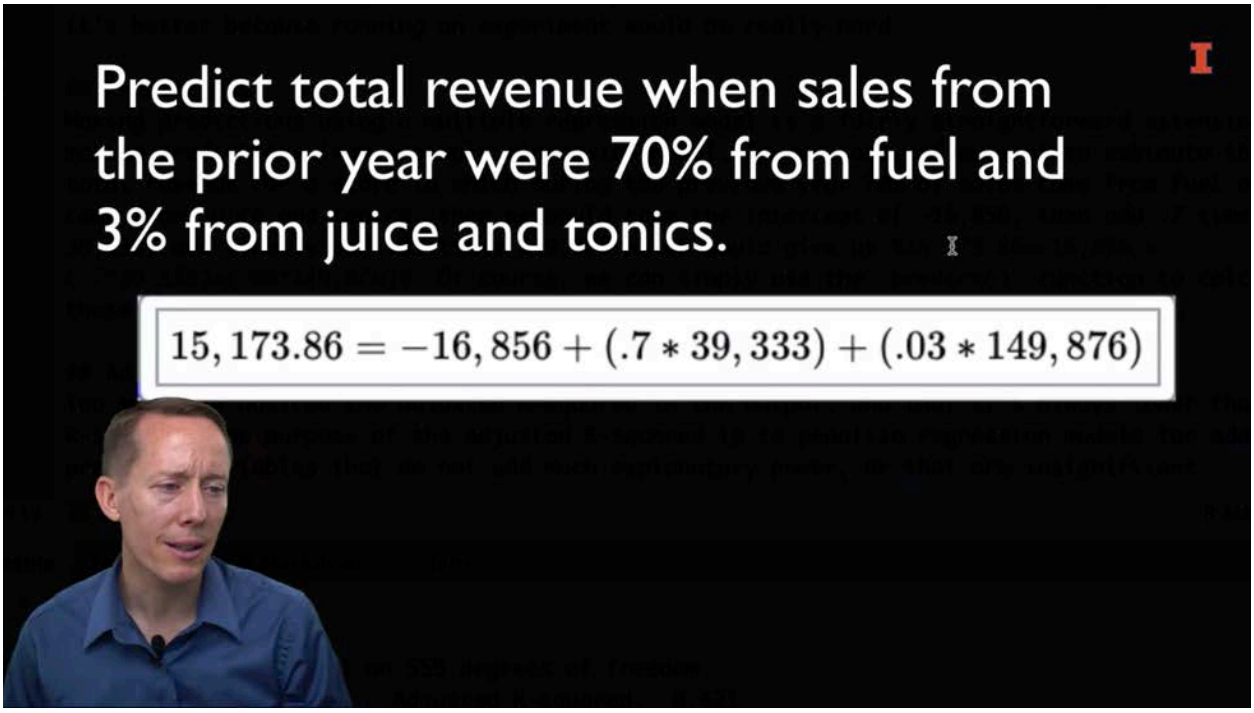


In the absence of being able to run an experiment, this is the next best thing.

In some ways, it's better because running an experiment would be really hard. Making predictions, using a multiple regression model is a fairly straightforward extension of making predictions from a simple regression model.

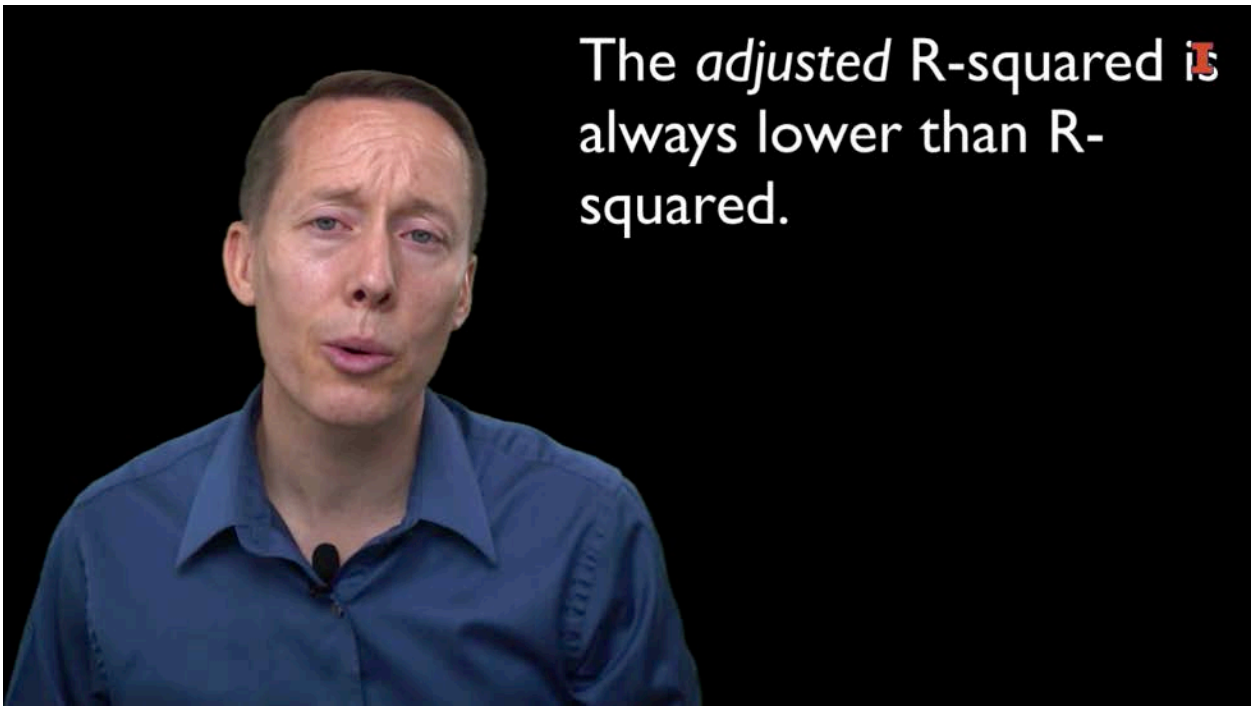


In our case, if we want to predict total revenue for a year, in which sales during the prior year were 70% from fuel and 3% from juice and tonics. Then we can take these coefficients for model 3 and calculate the predicted value in the following way.

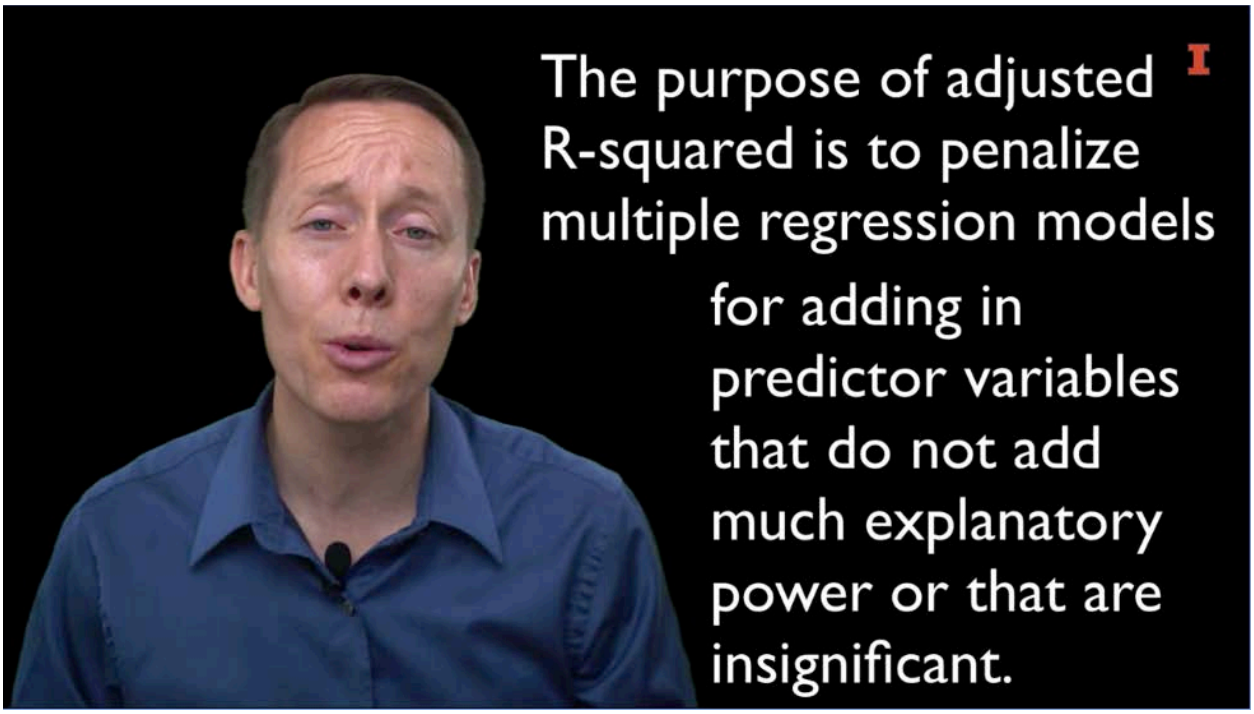


And I have this typed up nicely here. So we can take the intercept of negative 16,856, and add to it the product of 0.7, meaning the 70% of sales from fuel during the previous year. Times that coefficient from fuelpy1 of 39,333 plus the product of 0.3 times the

coefficient on juice tonics_py1 of 149.876. And that would give us an estimate of 15,174 approximately. Now, of course we can simply use the predict function to calculate this force, but conceptually, that's how you would do it.

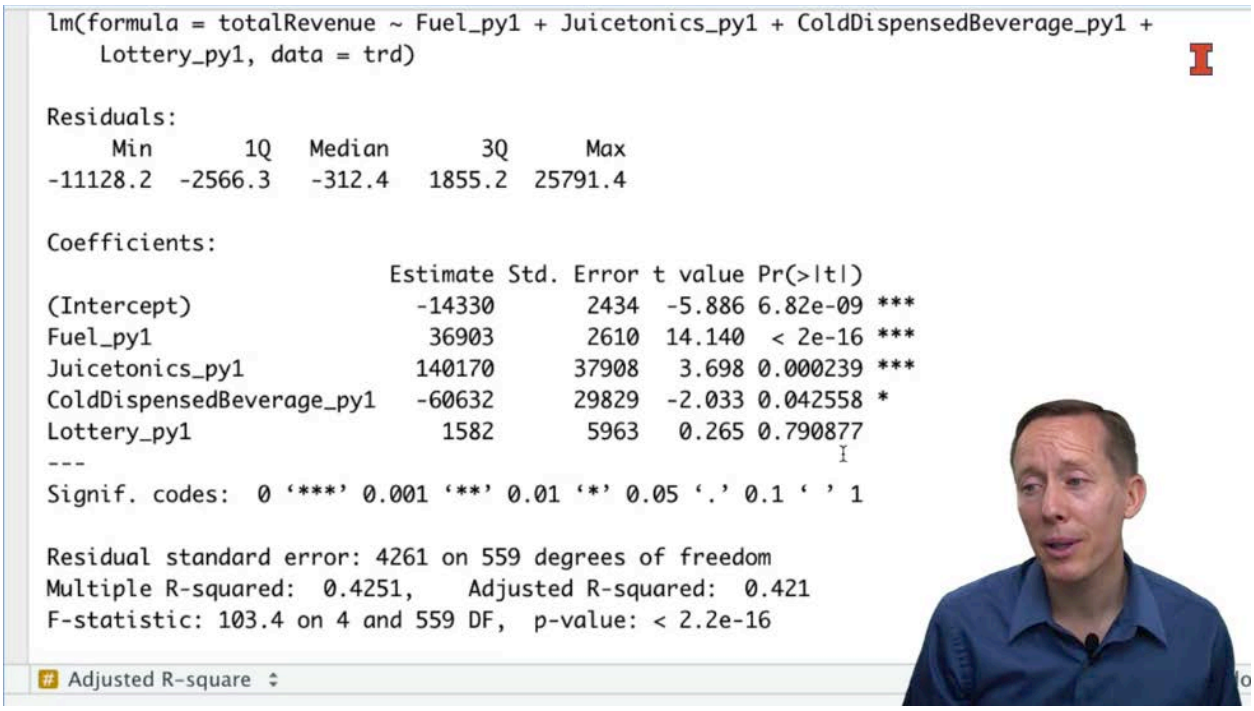


You may have noticed in the regression output that the adjusted R squared is always lower than the R squared.

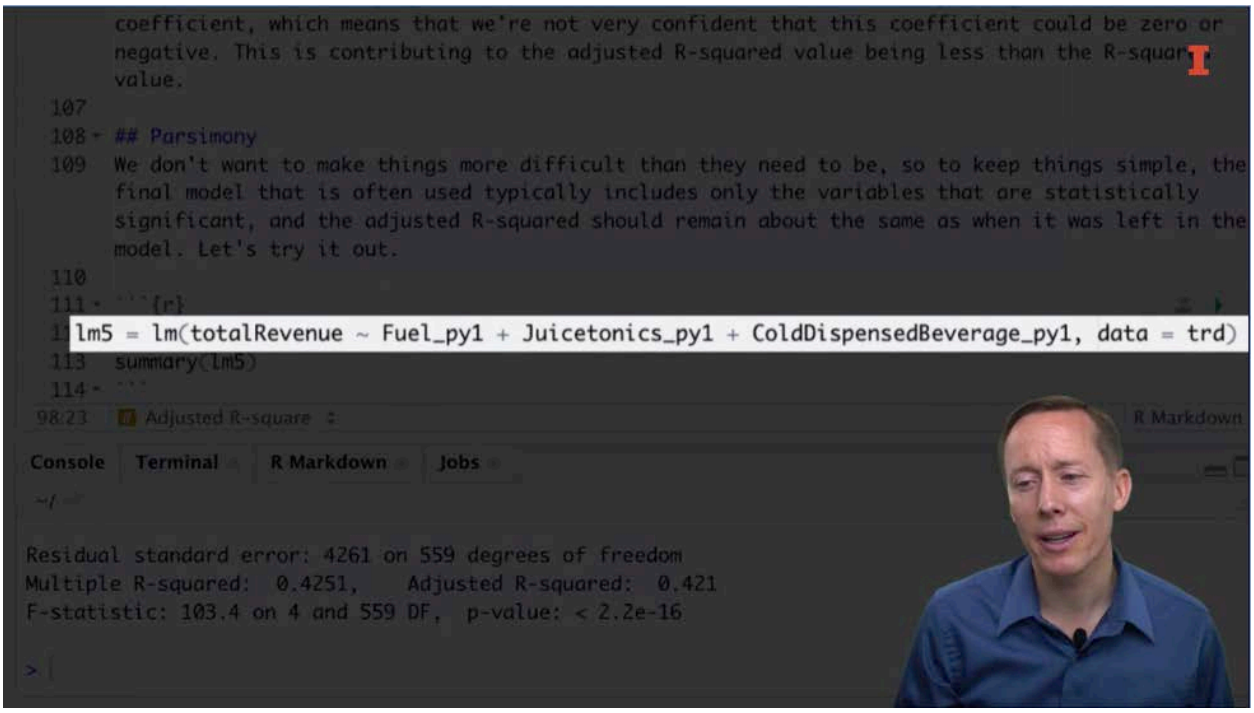


The purpose of the adjusted R squared is to penalize multiple regression models for adding in predictor variables that do not add much explanatory power or that are insignificant.

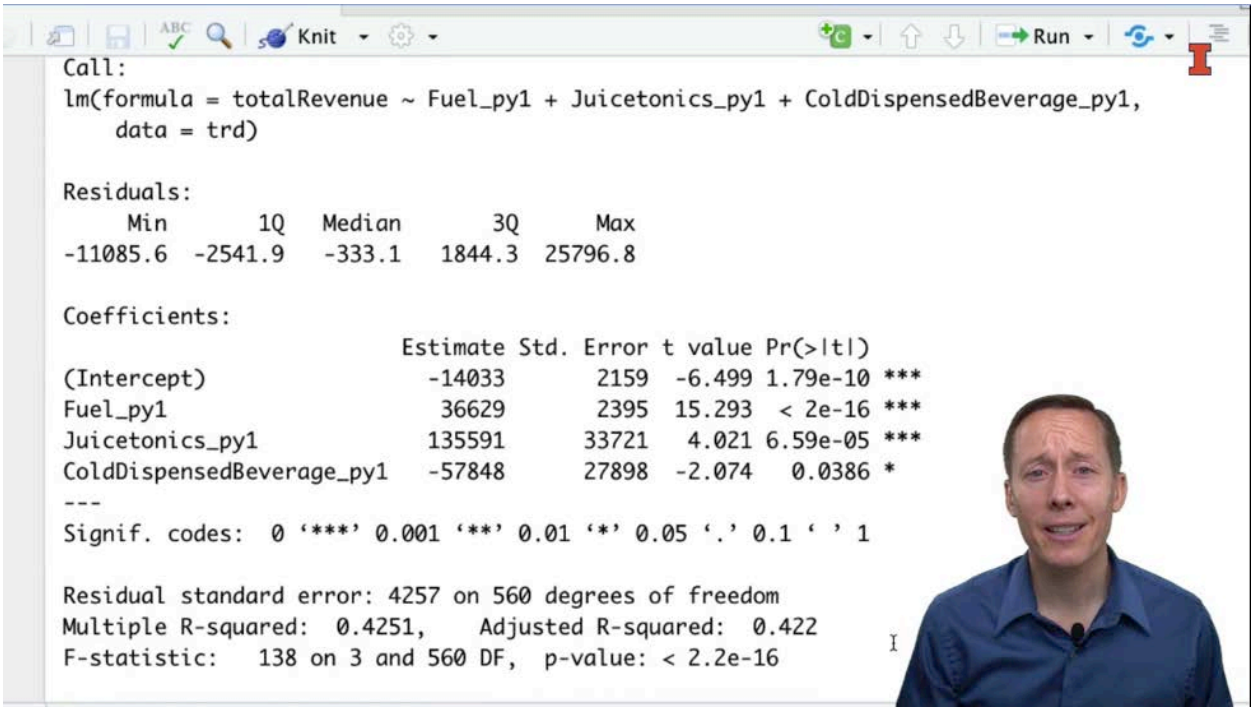
So let's test this out by running another regression model. And in this case we will include colddispenseBeverage py1 and Lotterypy1 in this model.



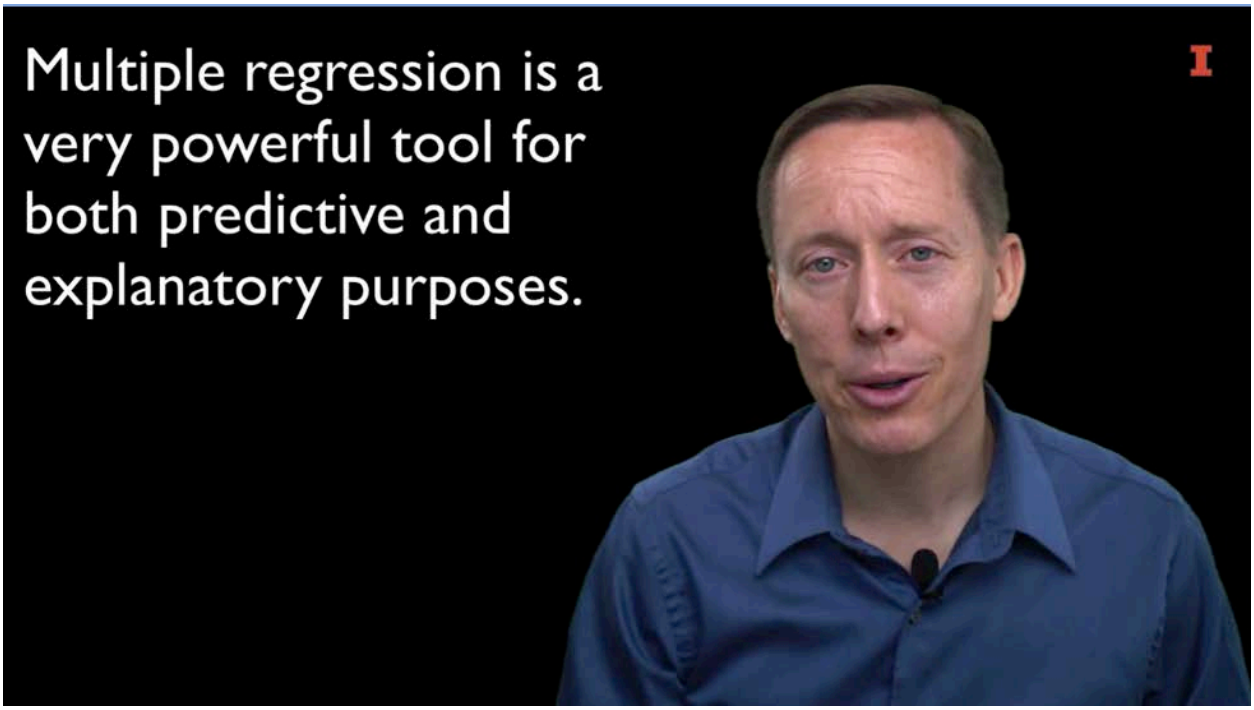
And when we do that, you can see that the coefficient on Lotterypy1 is statistically insignificant. This means that we cannot be confident that this estimate for Lotterypy1 is positive, in fact it could even be negative. This is contributing to the adjusted R squared value being less than the R squared value by just a little bit. Now we don't want to make models any more complex than they need to be. So to keep things simple, the final model that is often used typically includes only the variables that are statistically significant. And you'll notice that the adjusted R squared should remain about the same as when that variable was left in the model. Let's go ahead and try this out.



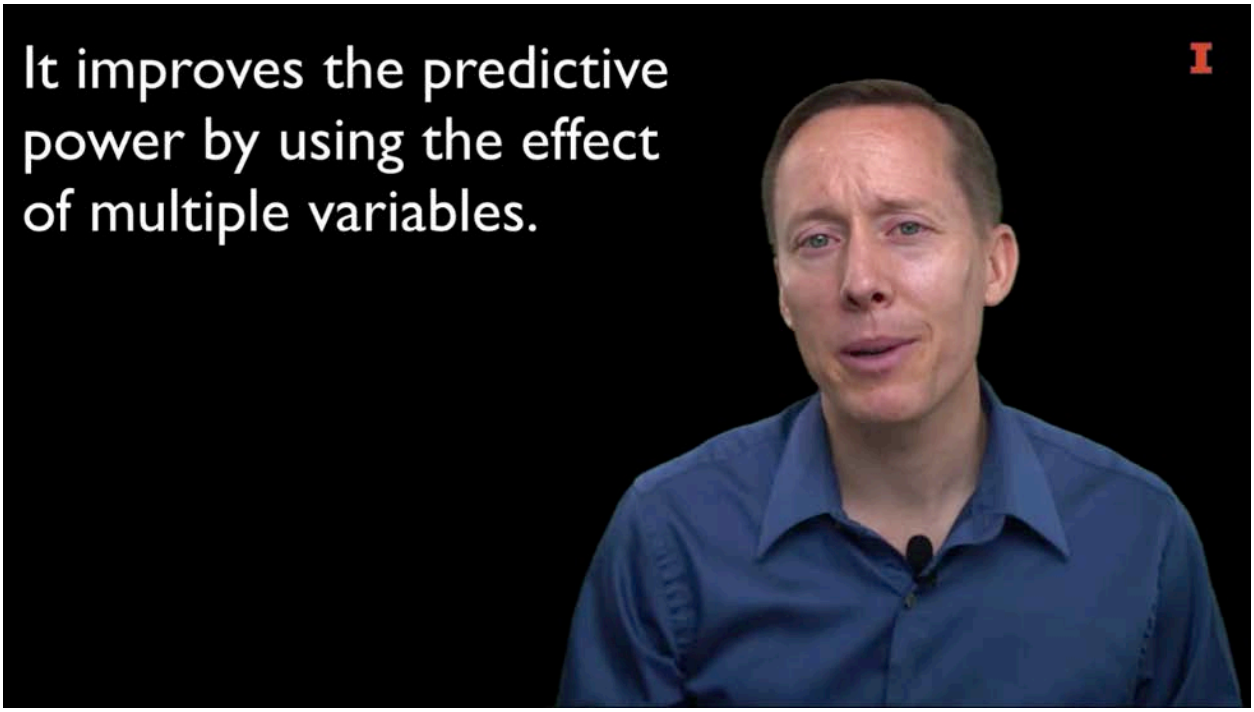
So we will create this Lm5 object in which we regress total revenue on fuel, Juicetonics and coldDispensedBeverages, but we leave out lottery.



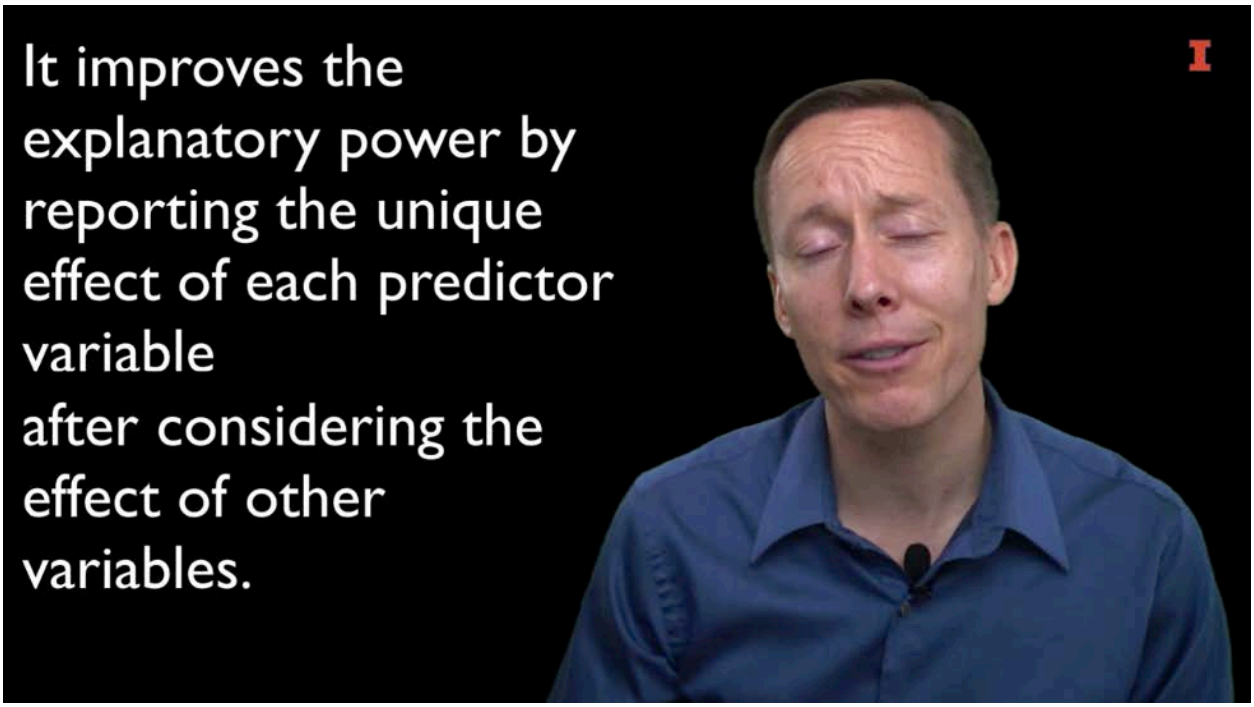
All right, in this case, the adjusted R squared not only stayed the same but even increased slightly.



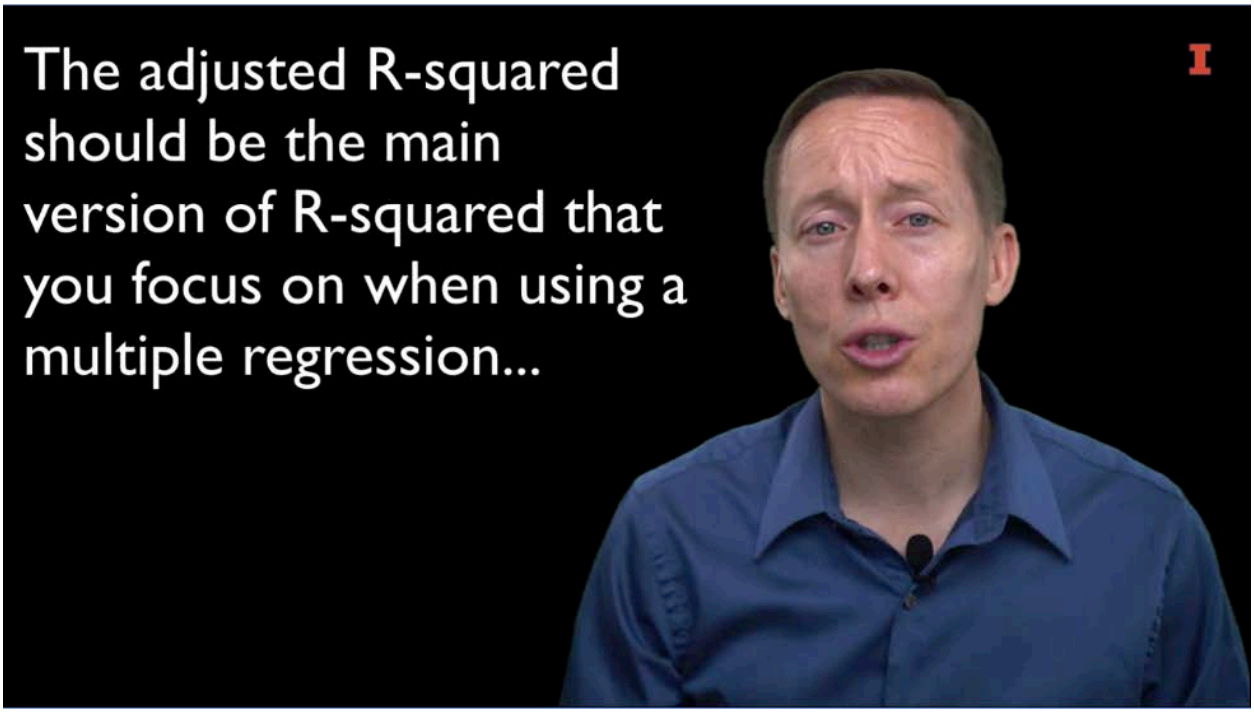
In conclusion, multiple regression is a very powerful tool for both predictive and explanatory purposes.



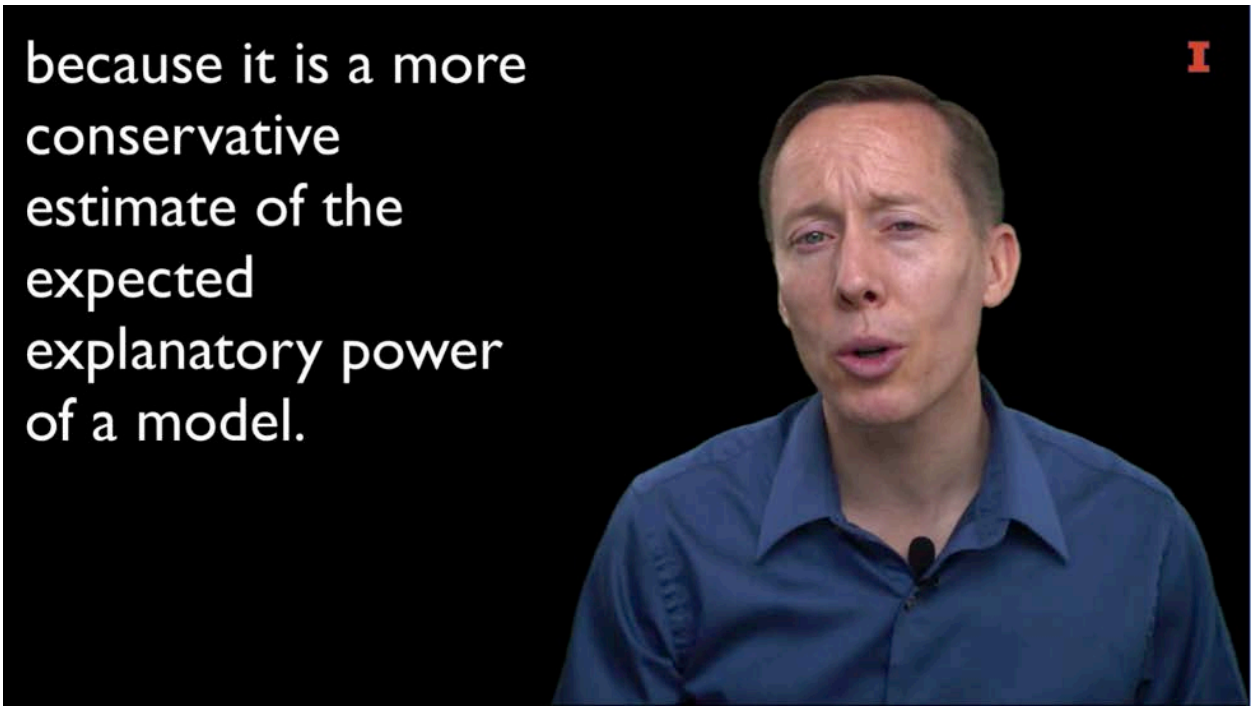
It improves the predictive power by using the effect of multiple variables.



It improves the explanatory power by reporting the unique effect of each predictor variable, after considering the effect of other variables.

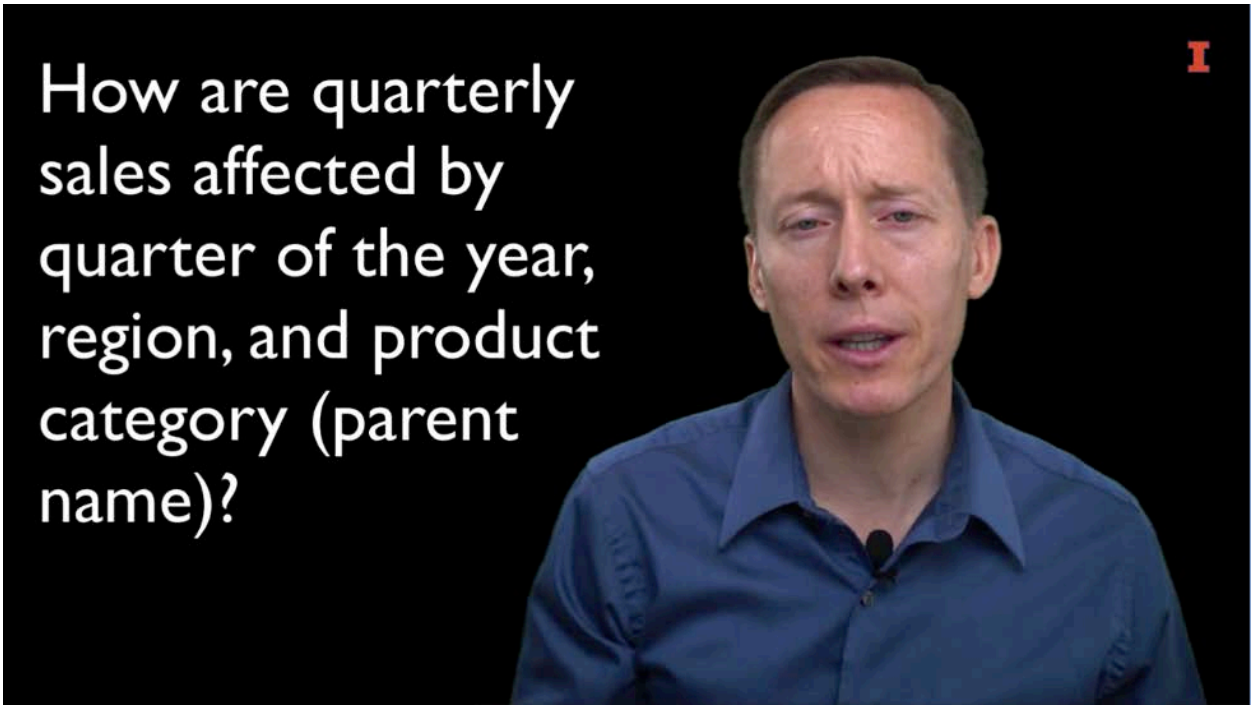


The adjusted R squared should be the main version of R squared that you focus on when using a multiple regression.

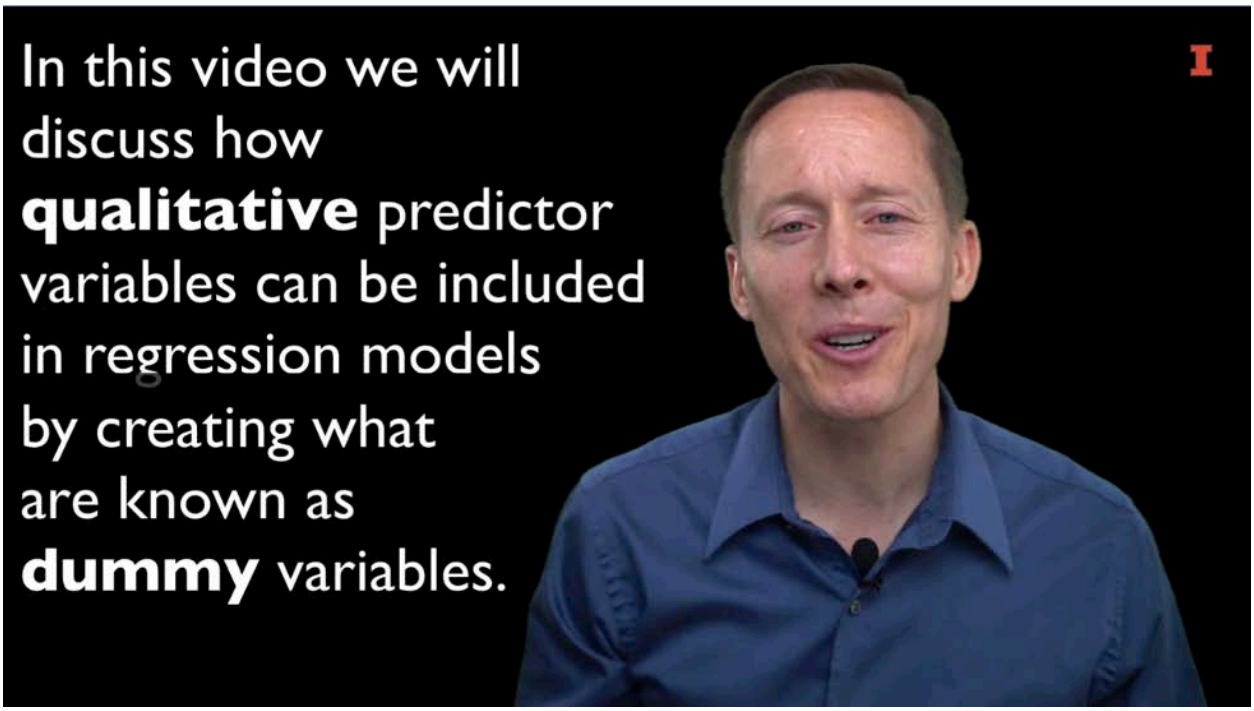


Because it is a more conservative estimate of the expected explanatory power of a model.

Lesson 1-4.6 Dummy Variables

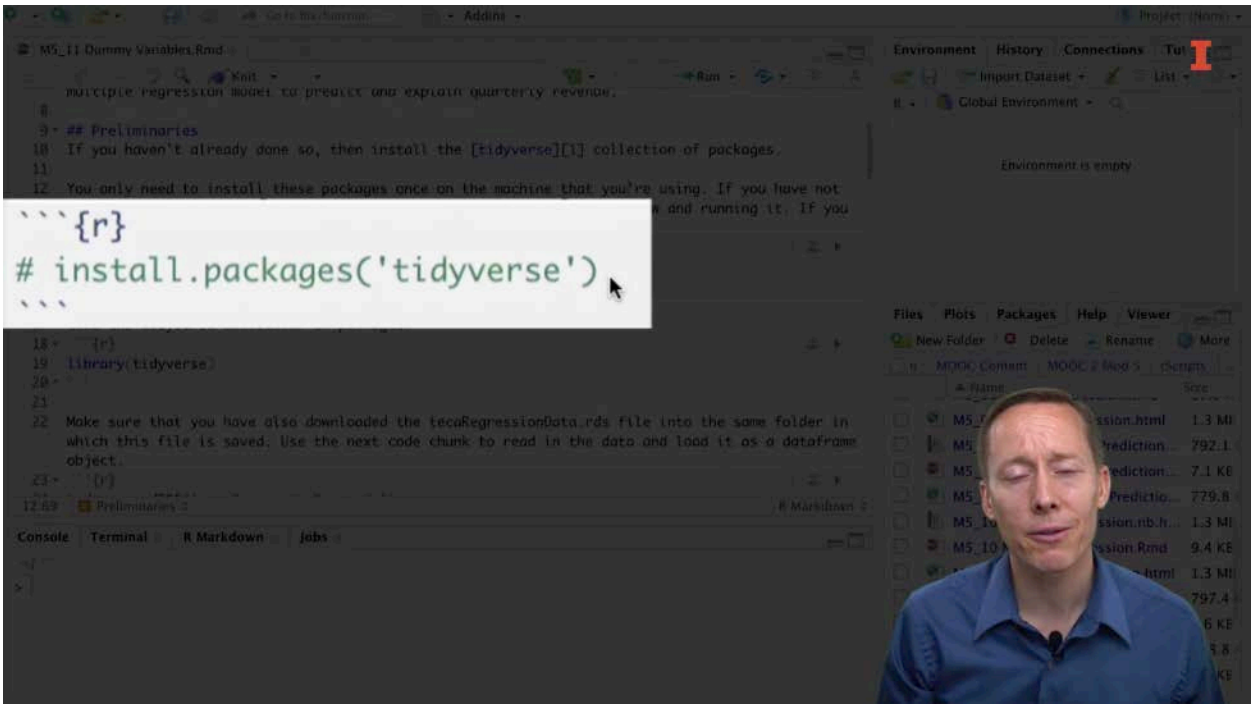


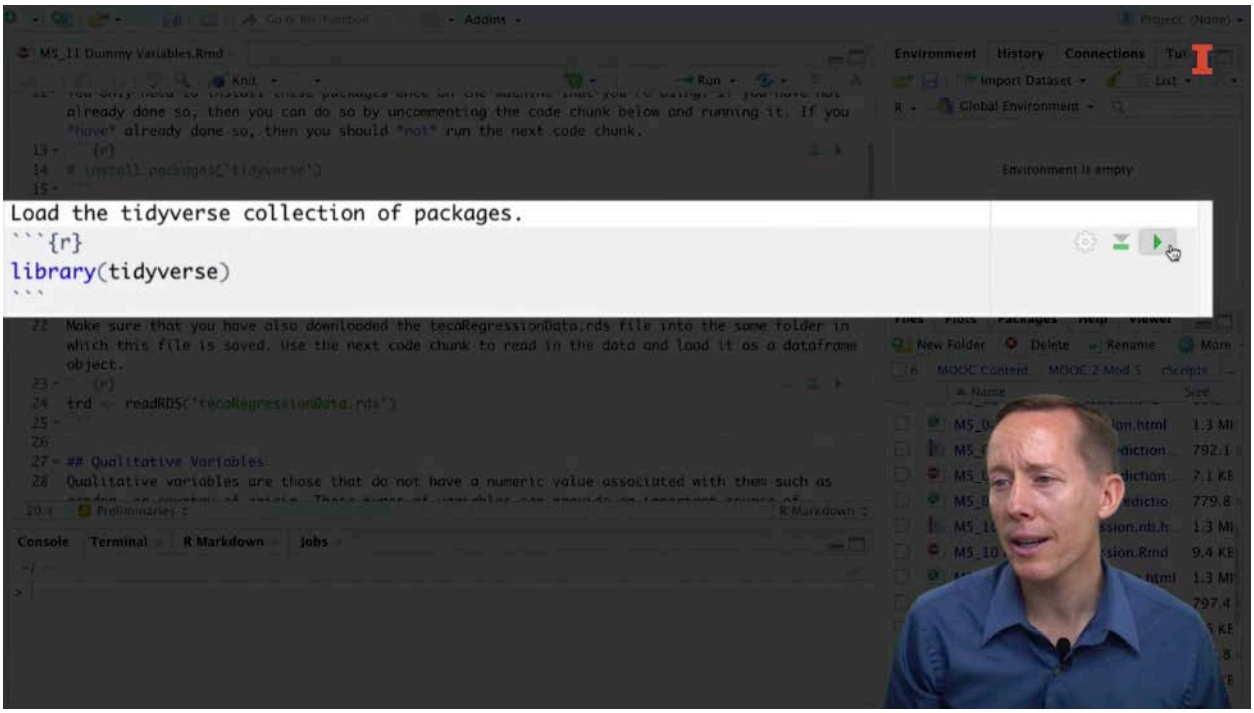
Recall that our broad business question is how are quarterly sales affected by quarter of the year, region and by product category.



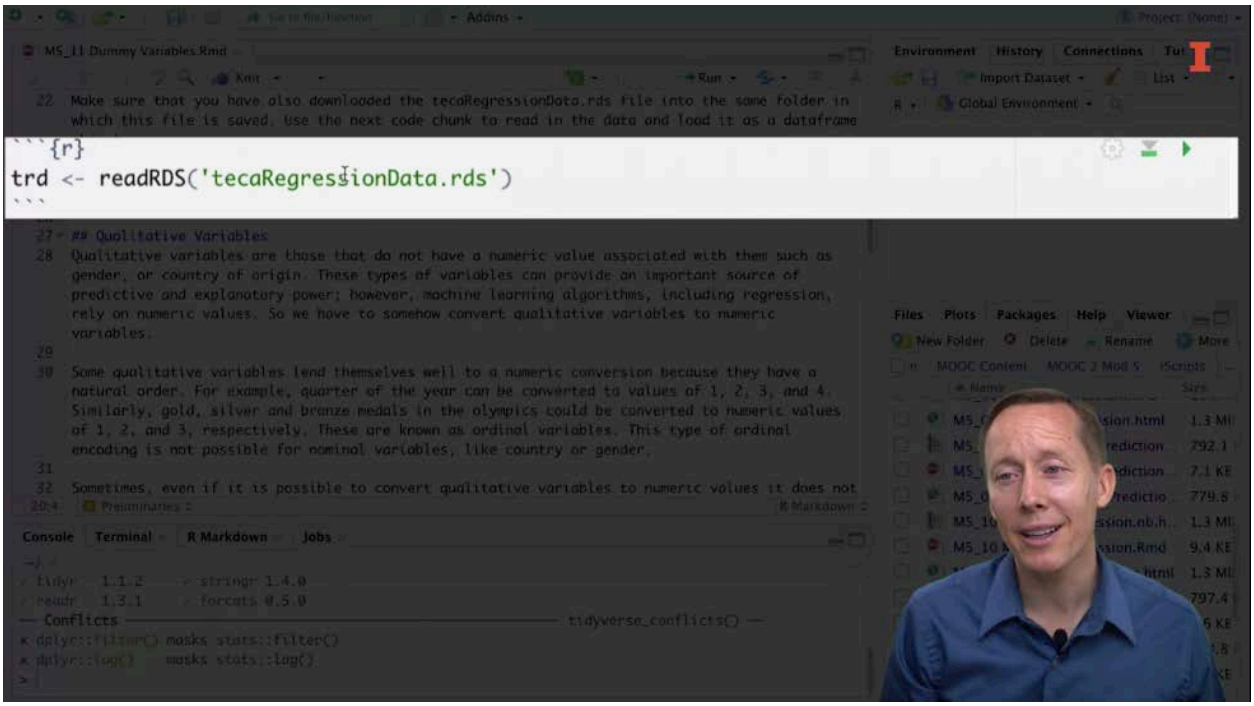
Up to this point, we have analyzed the ability of quantitative predictor variables to improve predictions. In this video, we will discuss how qualitative predictor variables can be included in regression models by creating what are known as dummy variables.

This will allow us to incorporate quarter of the year in a multiple regression model to predict and explain quarterly revenue.

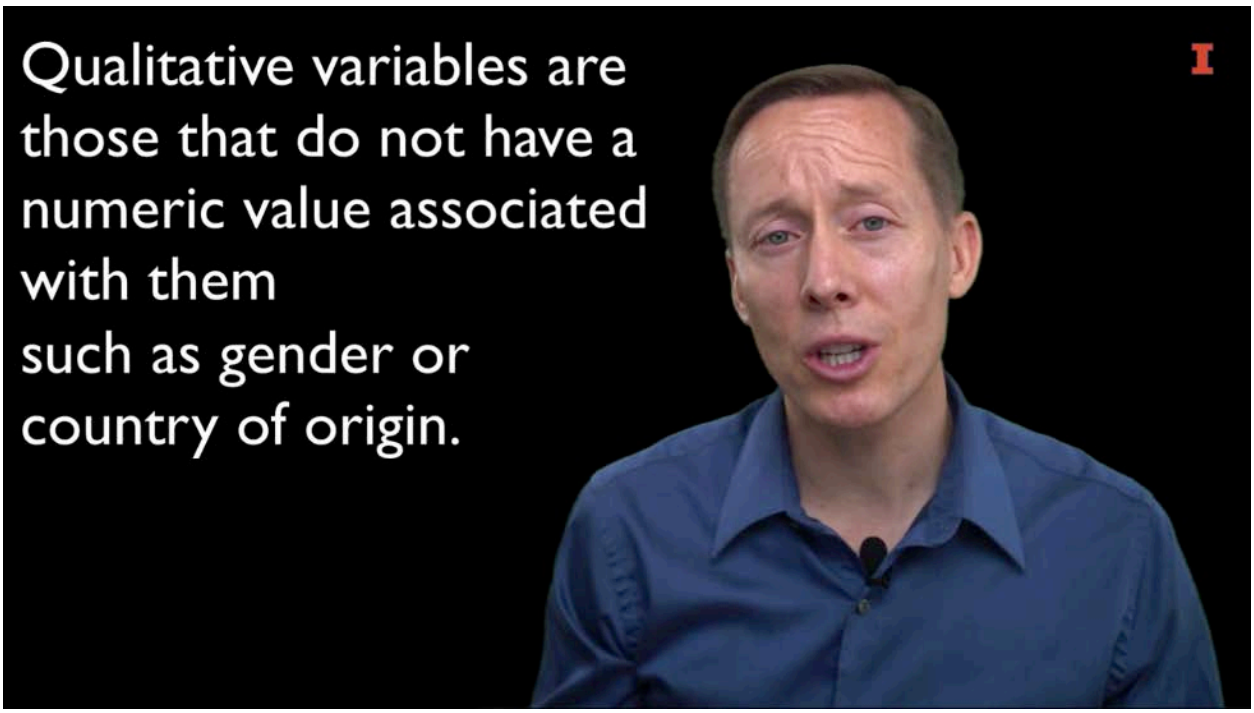




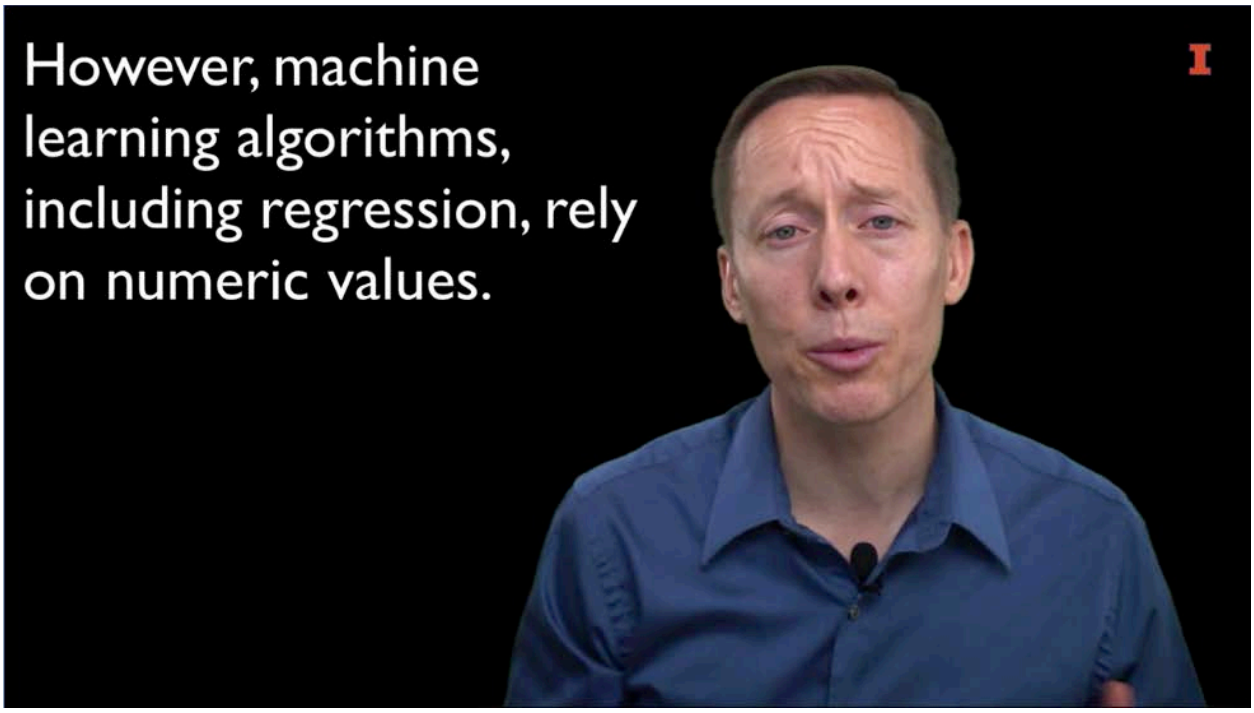
I encourage you to follow along as we go through this lesson. To do so, you'll want to download the associated R Markdown file and the teca regression data. Also make sure that you have already installed the tidyverse collection of packages and then go ahead and load those packages.



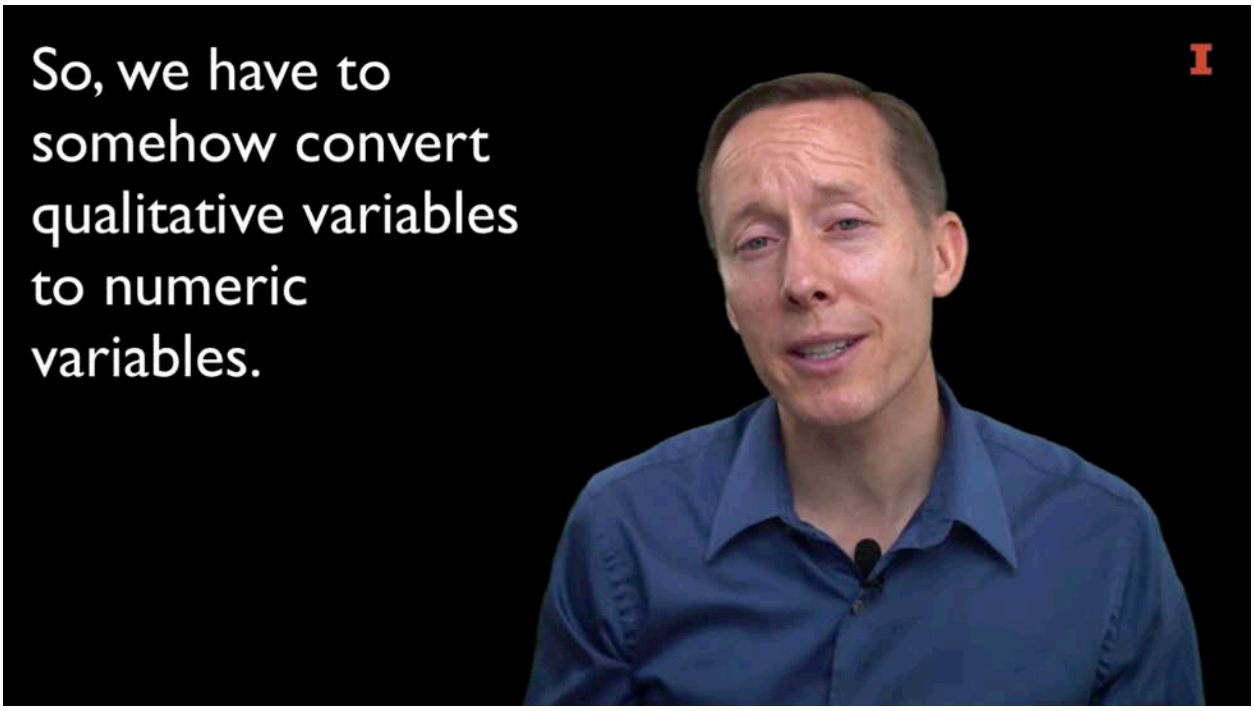
Also, make sure that you read in the tecaRegressionData.rds file and save it as a TRD data frame.



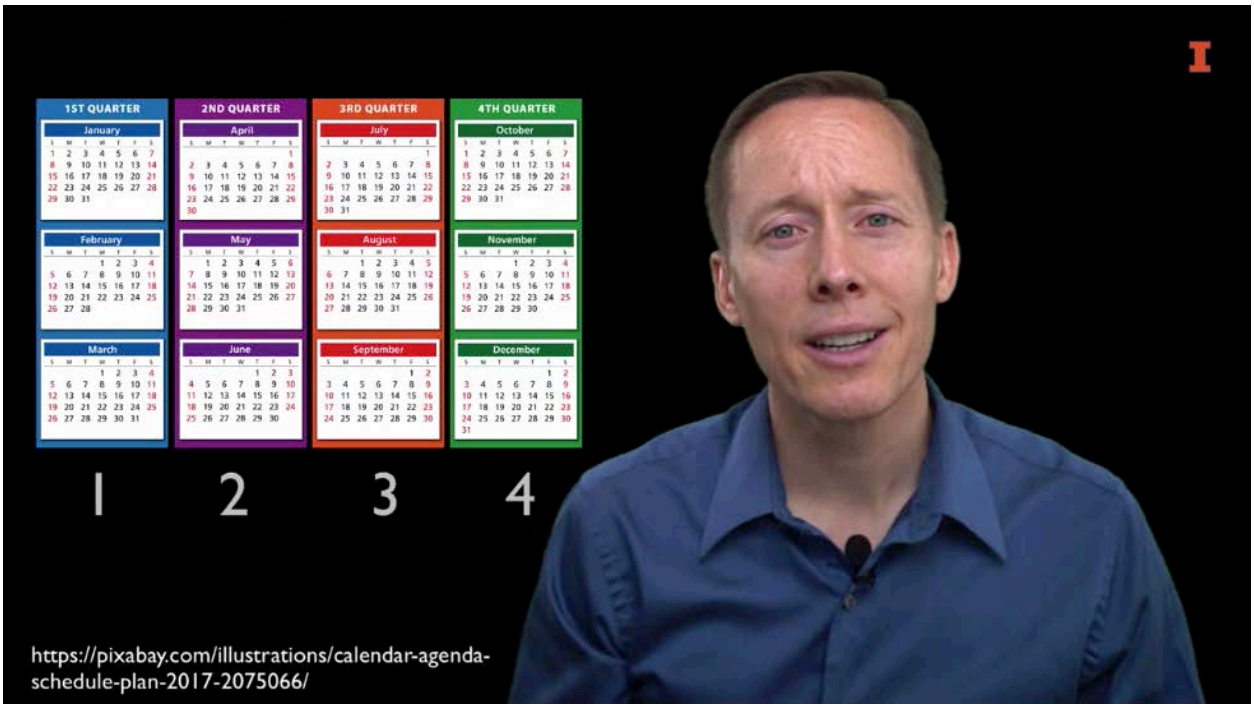
Qualitative variables are those that do not have a numeric value associated with them, such as gender or country of origin. These types of variables can provide an important source of predictive and explanatory power.



However, machine learning algorithms, including regression, rely on numeric values.

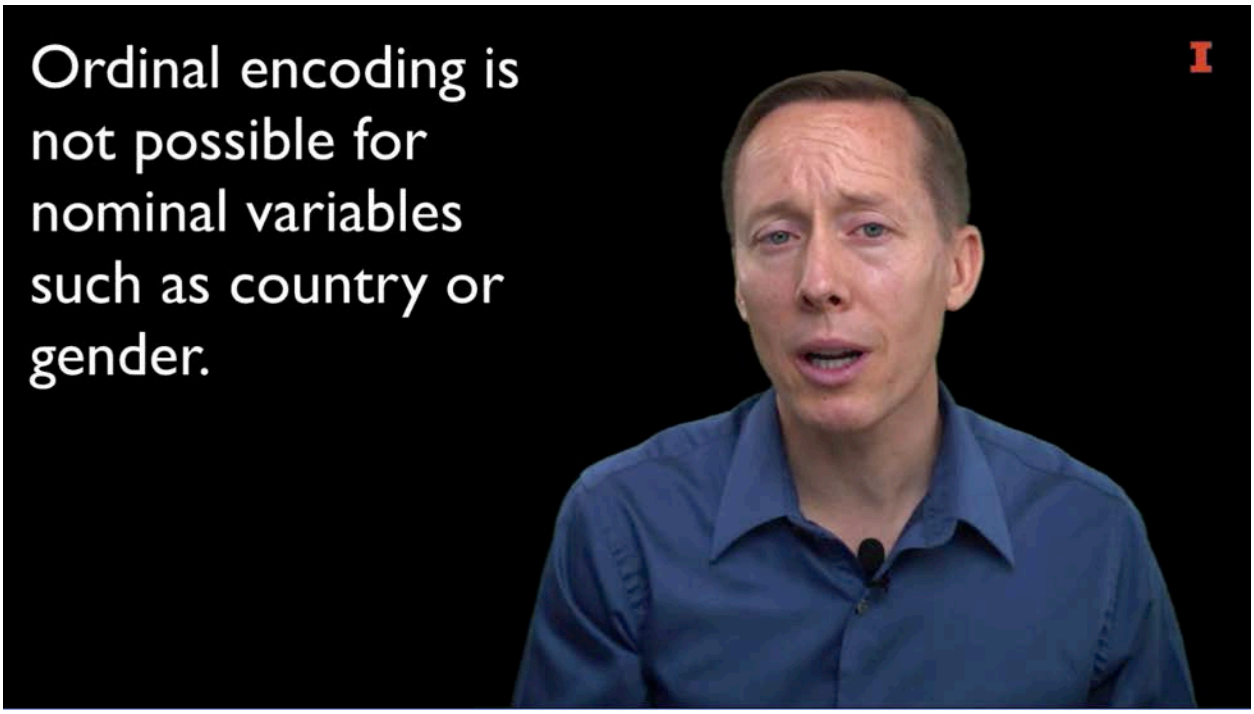


So we have to somehow convert qualitative variables to numeric variables. Some qualitative variables lend themselves well to numeric conversion because they have a natural order.

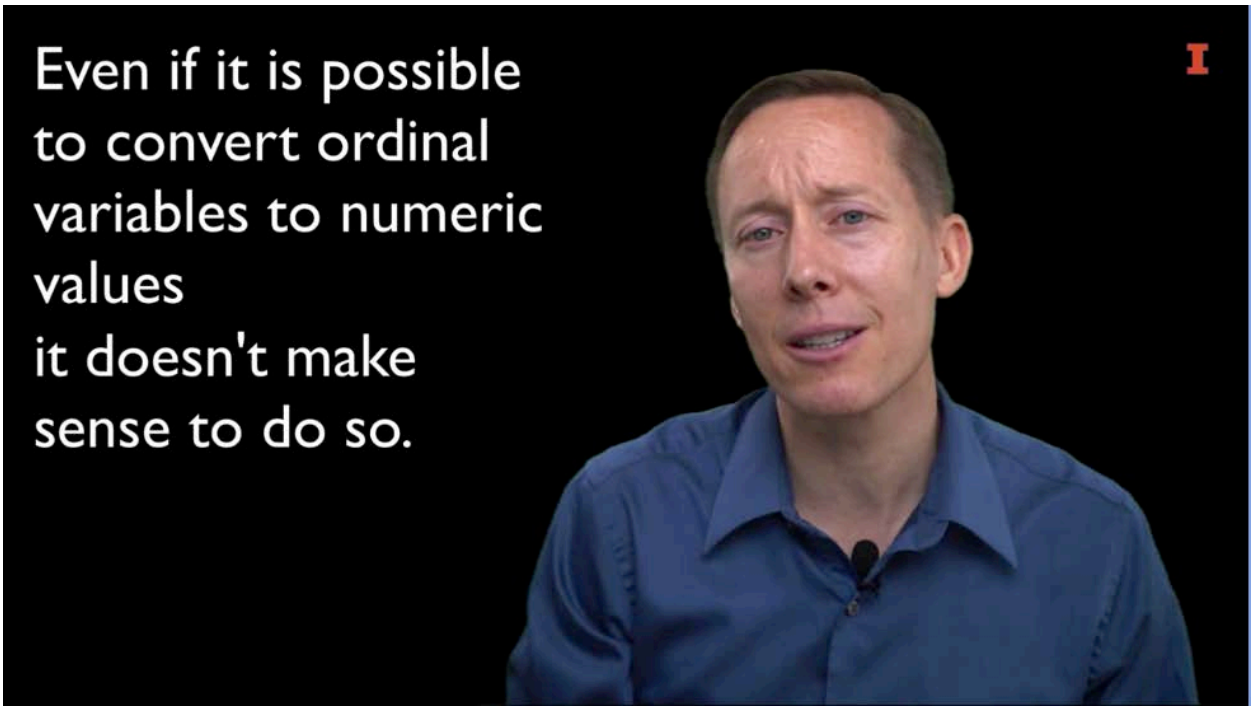


For example, quarter of the year can be converted to values of 1, 2, 3, and 4.

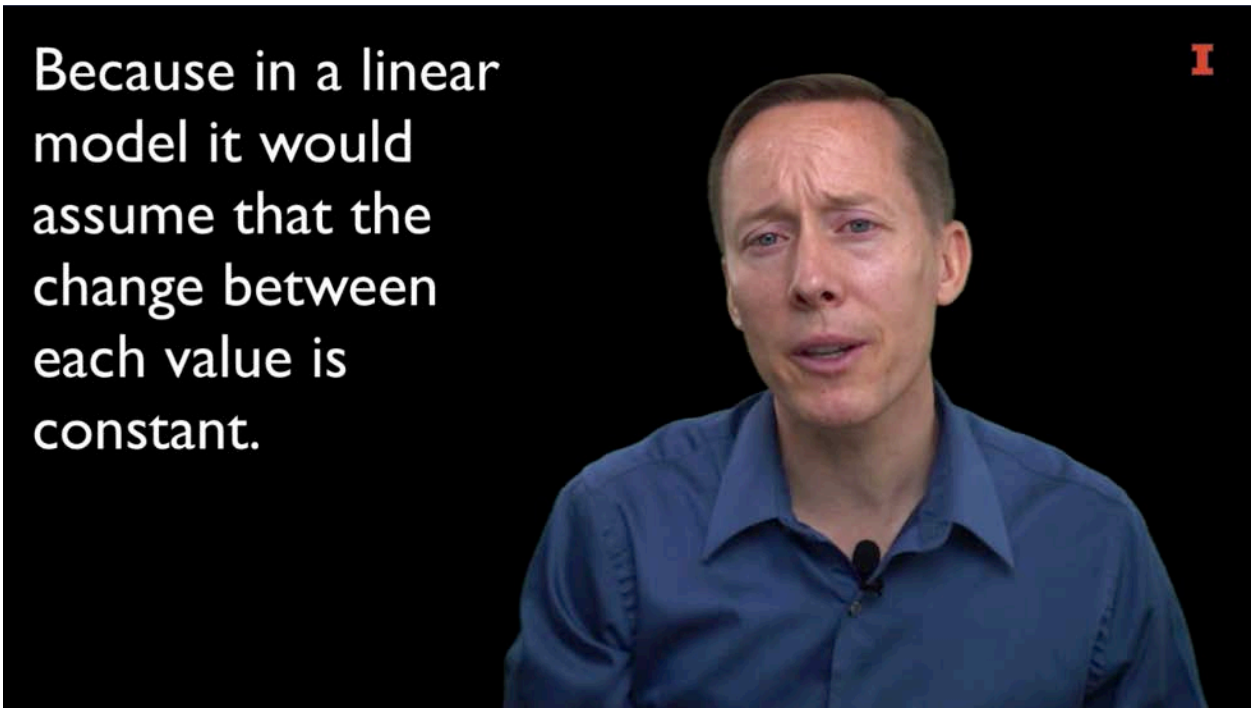
Similarly, gold, silver, and bronze medals in the Olympics could be converted to numeric values of 1, 2, and 3 respectively. These are known as ordinal variables.



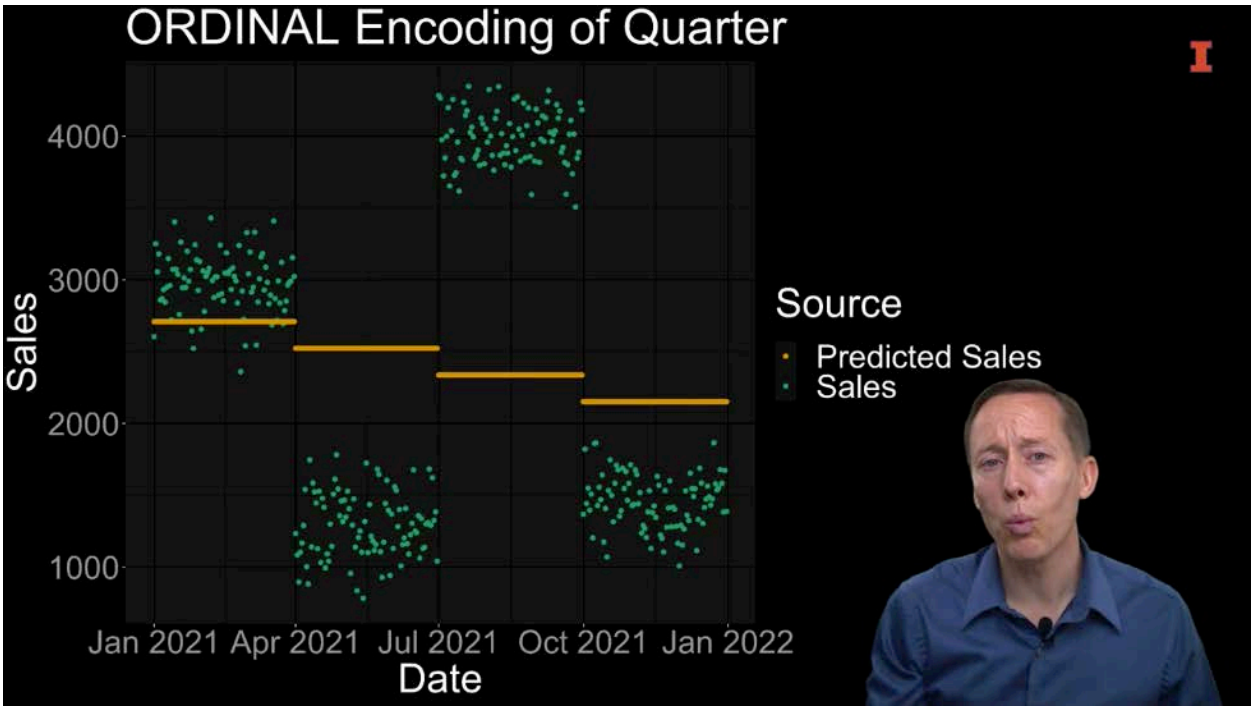
This type of ordinal encoding is not possible for other variables known as nominal variables, such as country or gender.



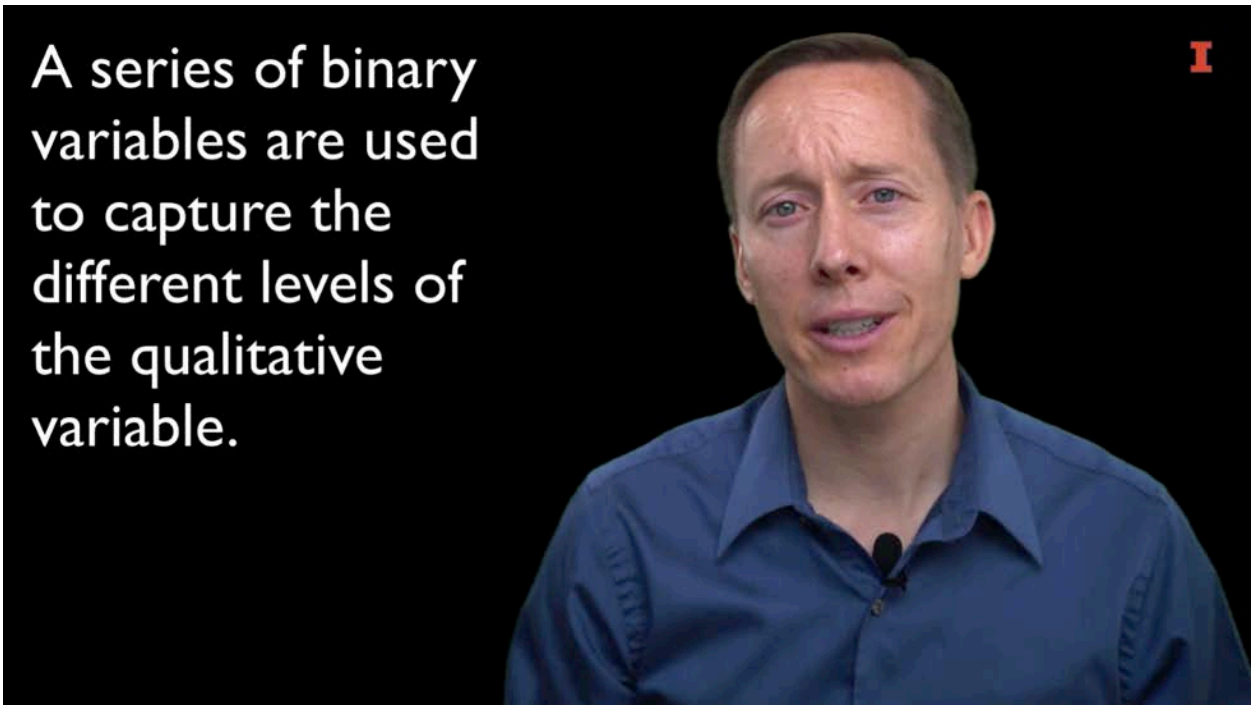
Sometimes, even if it is possible to convert ordinal variables to numeric values, it doesn't make sense to do so.



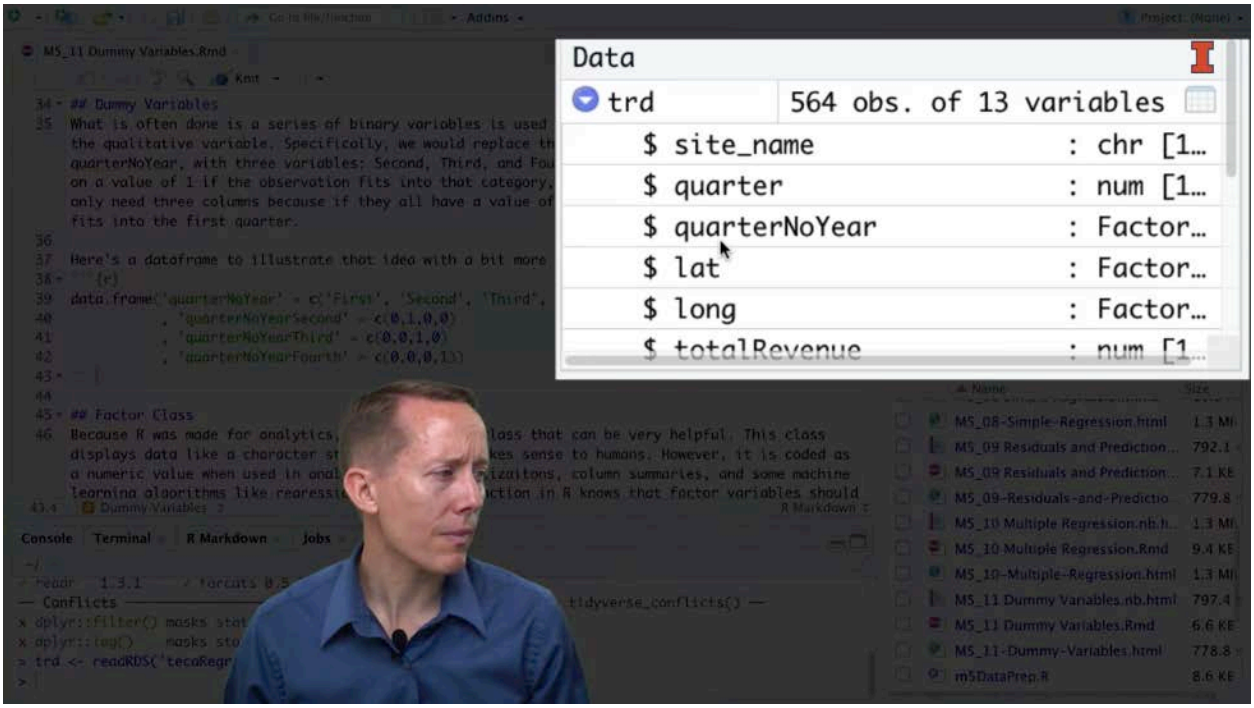
Because in a linear model, it would assume that the change between each value is constant. This is especially true for a quarter of the year for instance.



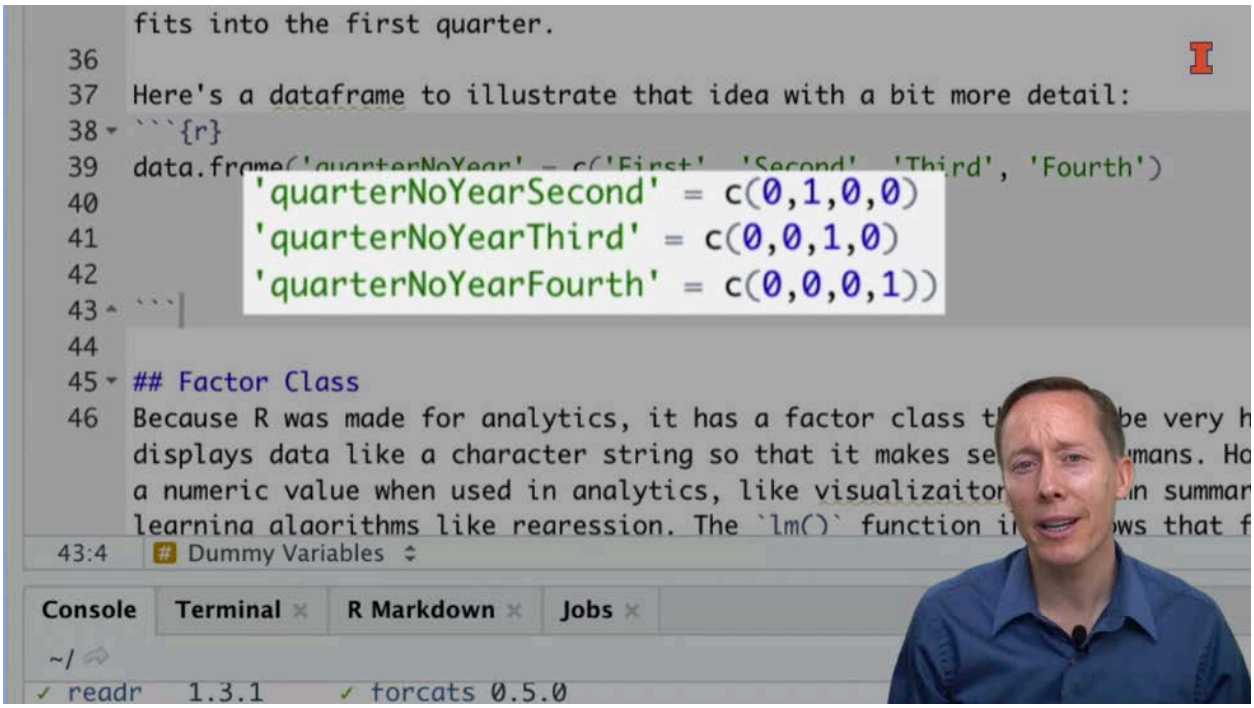
If quarters 1 and 3 are the busiest seasons of the year for some industries, and then lower in quarters 2 and 4, then it wouldn't make sense to force a constant positive or a constant negative coefficient on the variable that represents quarter of the year.



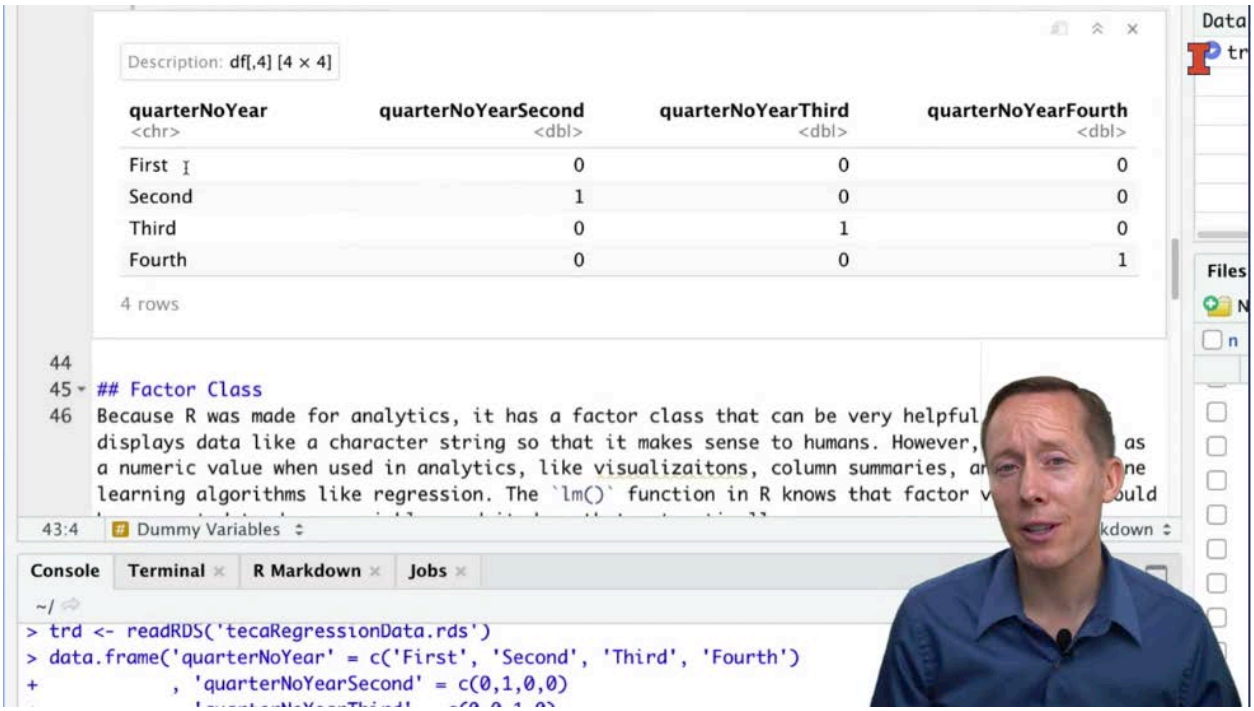
What is often done is a series of binary variables are used to capture the different levels of the qualitative variable.



Specifically for the quarter of the year variable, quarterNoYear, we would replace that with three other variables, 2nd, 3rd, 4th.

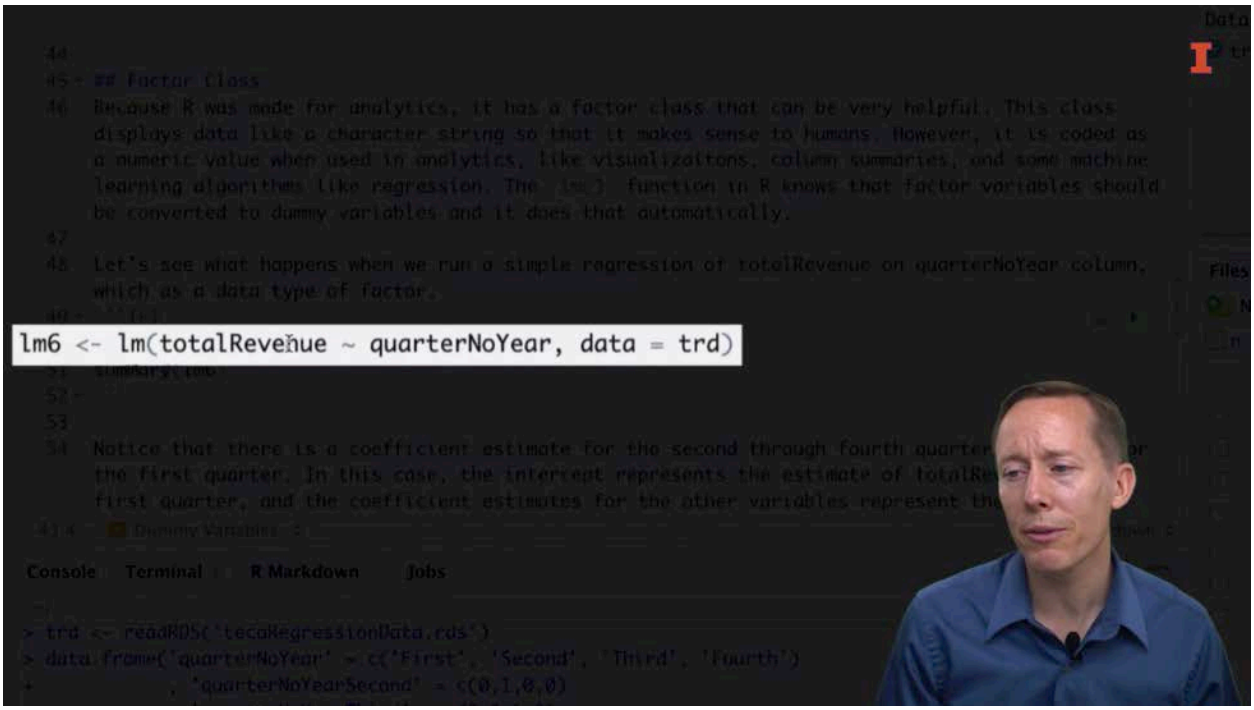


The values in these new columns take on a value of one if the observation fits into that category, and a value of zero otherwise. We only need three columns because if they all have a value of zero, then that means the observation fits into the first quarter. Here's a data frame to illustrate that idea with more detail.

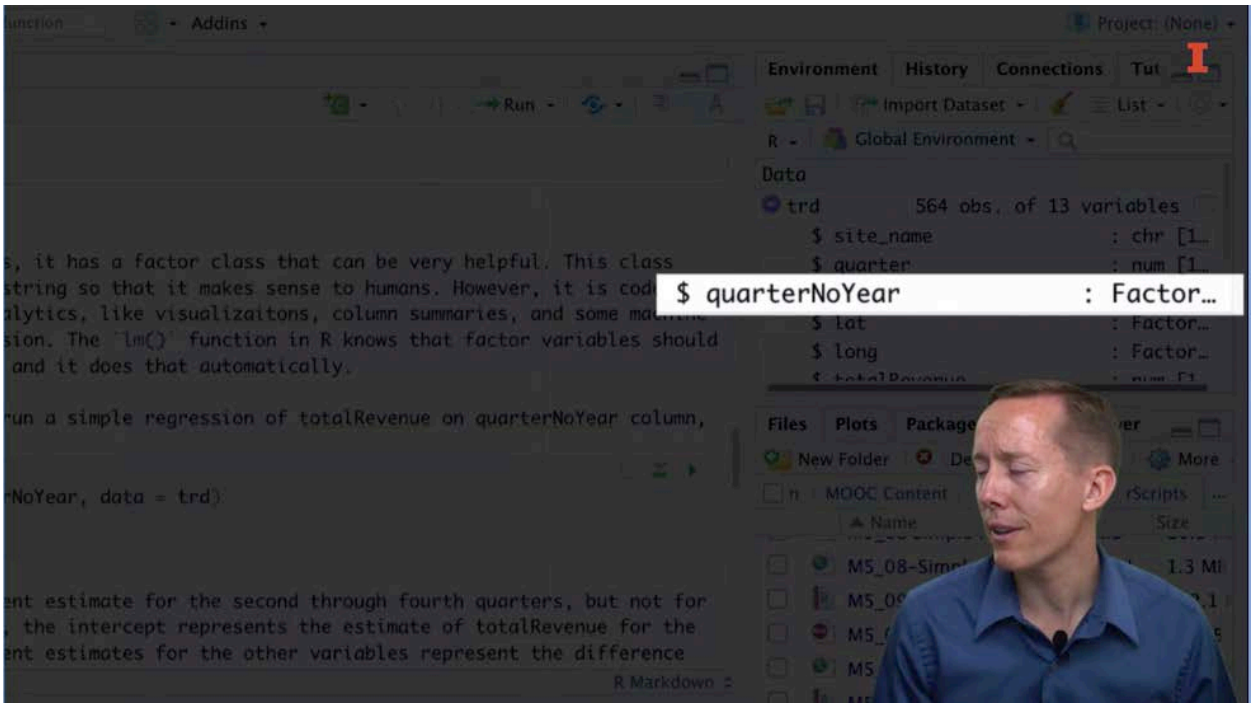


This first column represents the quarter of the year that the observation falls into, and we would replace this one column with these three columns, quarterNoYear 2nd, 3rd,

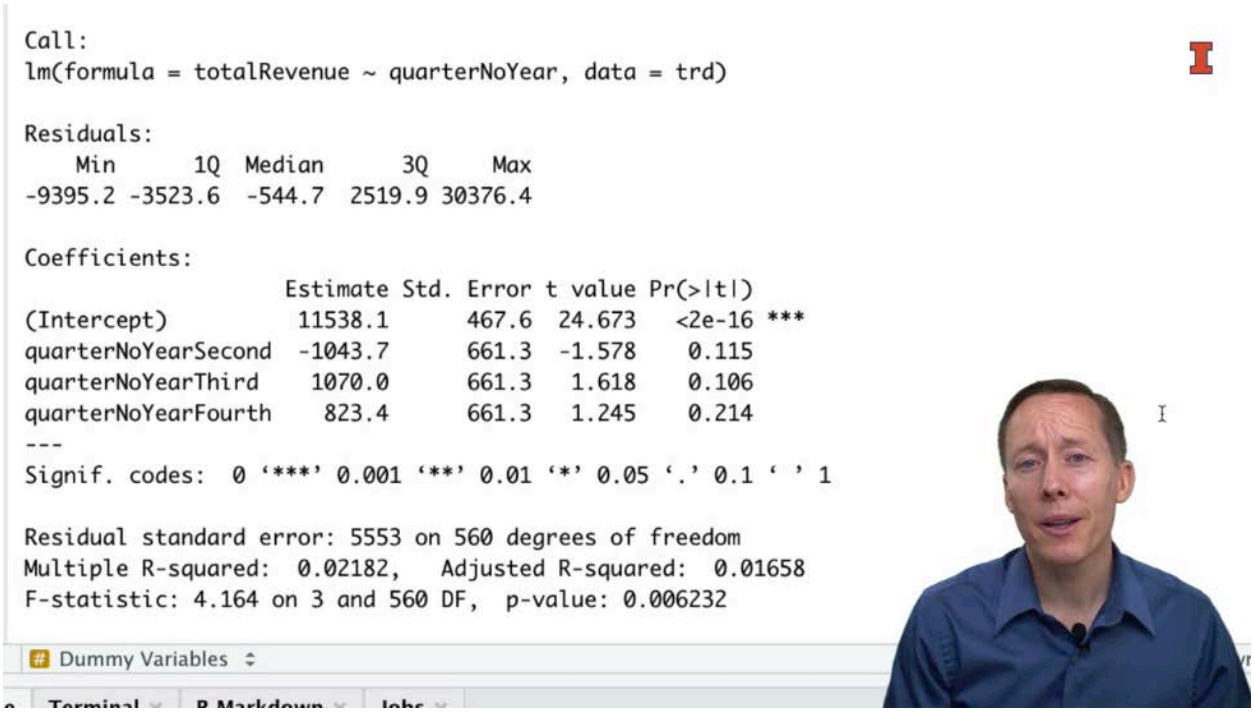
and 4th, and for observations that fit into the first quarter, they would have zeros in all three of those columns. For observations that fit into the second quarter, they would have a one in the quarterNoYear second and zero in the other two, and so on. Now, because R was made for analytics, it has a factor class that can be very helpful for dealing with categorical variables. This factor class displays data like a character string so that it makes sense for us to read as humans. However, it is coded as a numeric value when used in analytics like visualizations, column summaries, and some machine learning algorithms including regression. The lm function in R knows that factor variables should be converted to dummy variables and it does that automatically.



Let's test this out by running a simple regression of total revenue on the quarter No year factor variable.

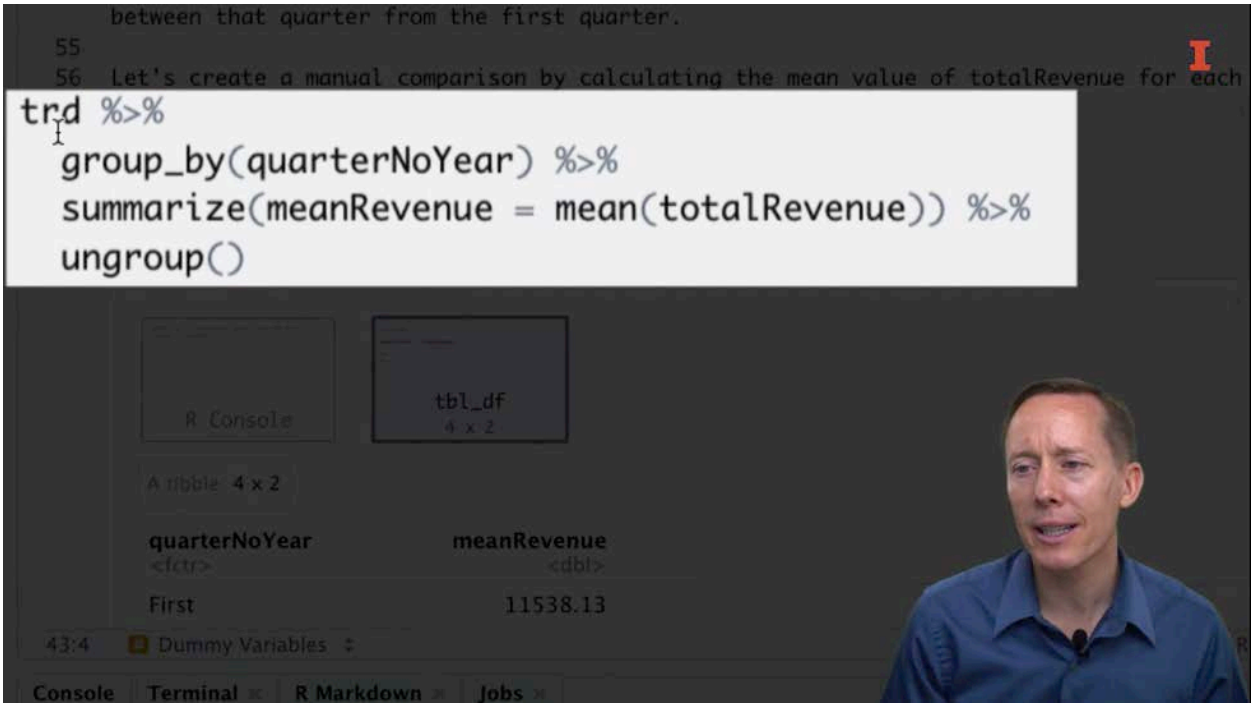


We can see that this is a factor in the environment pane here. Let's look at the summary.

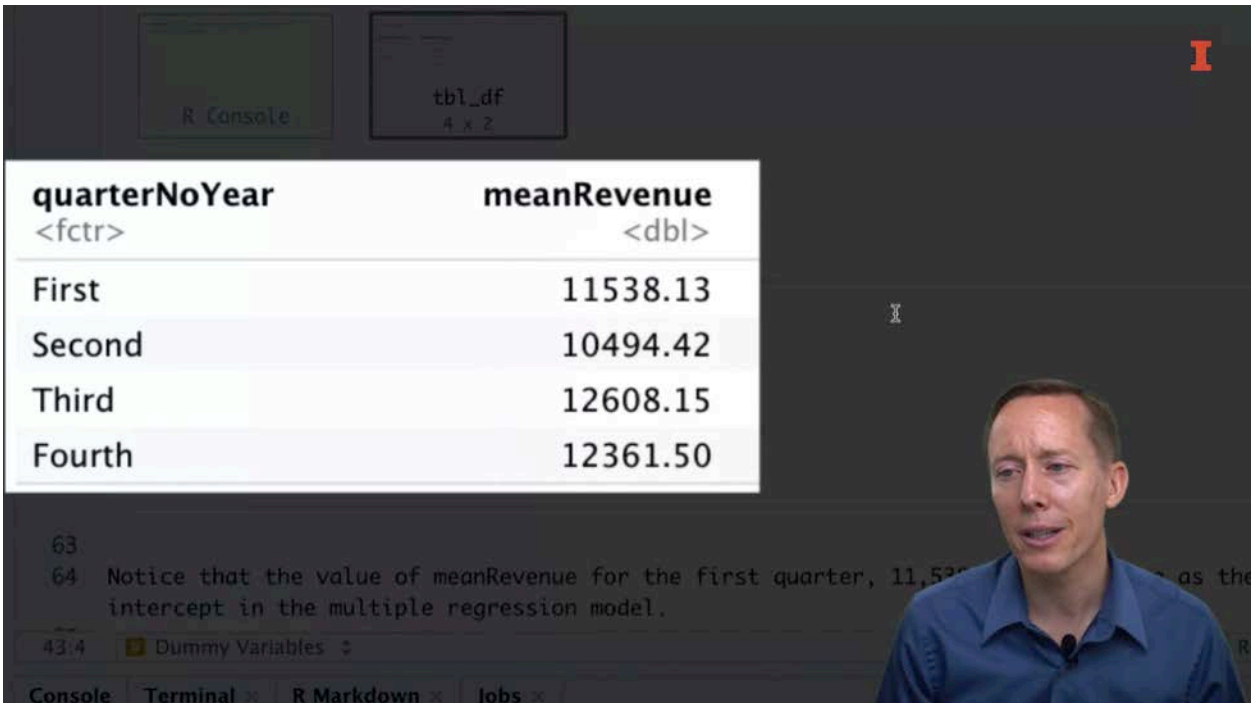


Notice that there is a coefficient estimate for the second through fourth quarters, but not for the first quarter. In this case, the intercept represents the estimate of total revenue for the first quarter and the coefficient estimates for the other variables represent the difference between that quarter from the first quarter. Let's create a manual comparison

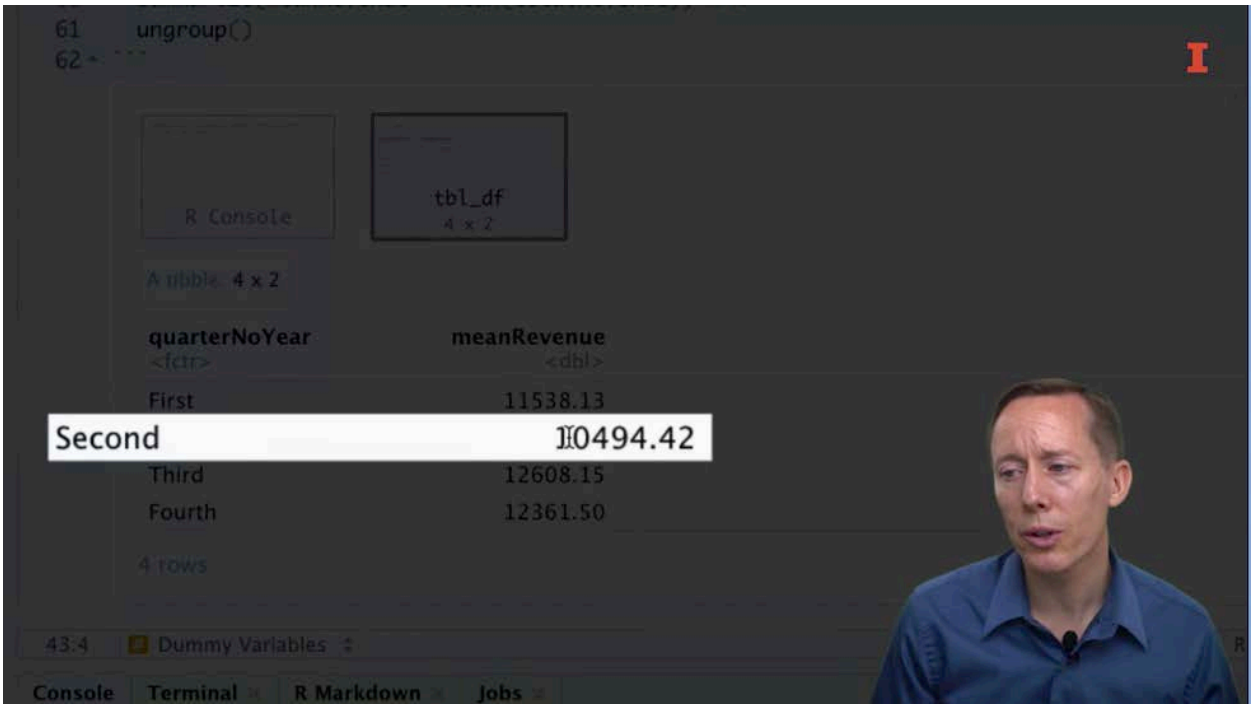
by calculating the mean value of total revenue for each quarter to verify these co-efficients and what they mean.



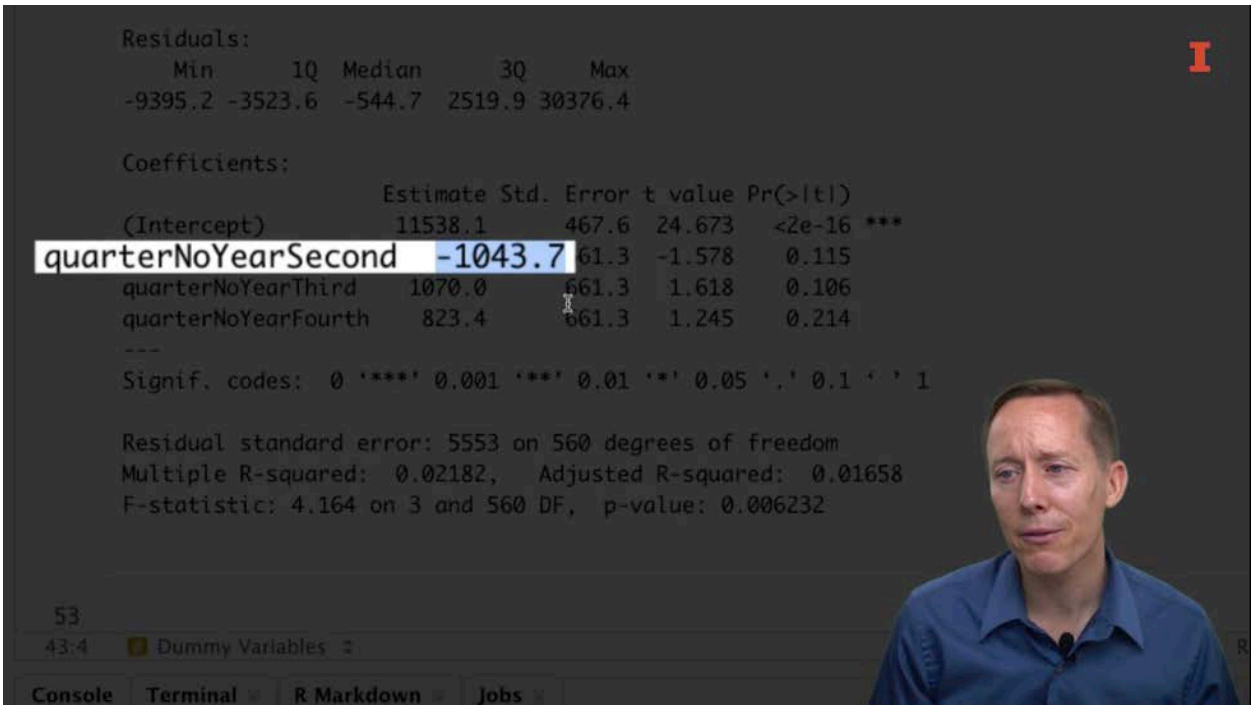
So I will go ahead and take that TRD data frame, group_by(quarterNoYear) and then summarize the mean value of total revenue for each of those quarters.



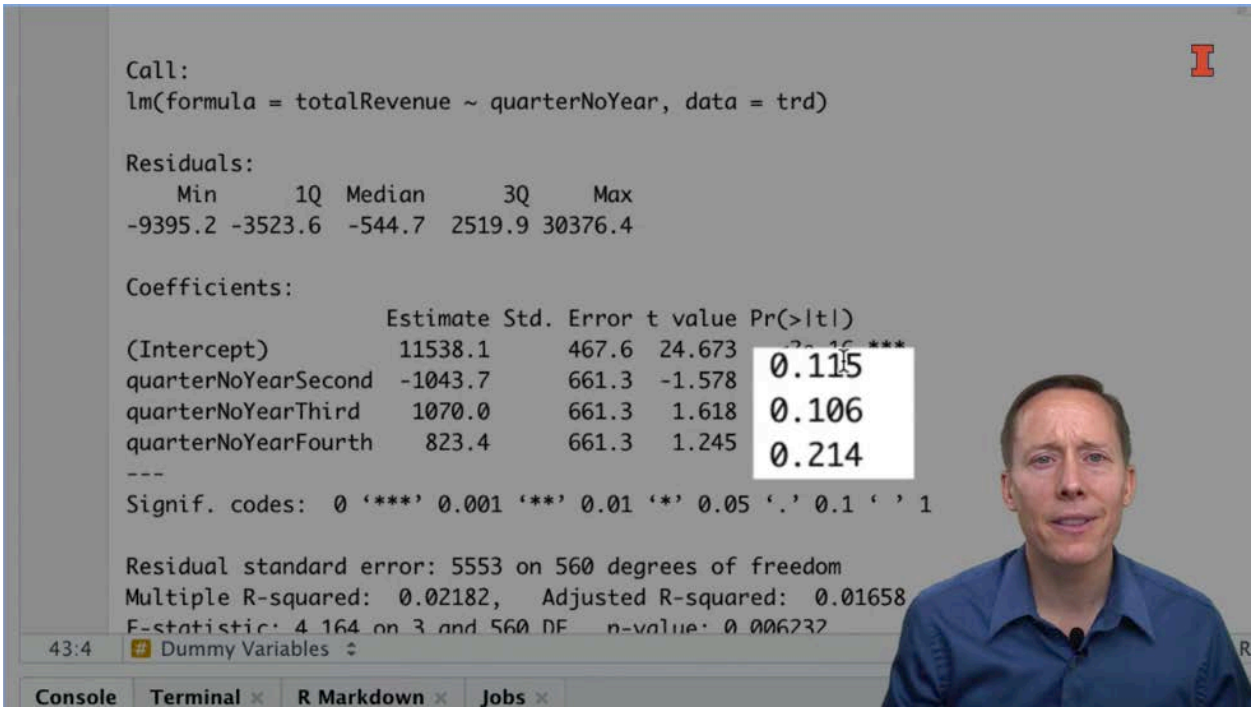
That will give us this table here. We can see that the first-quarter has a mean revenue of 11,538.13, and comparing that to our regression output, that's the estimate for the intercept.



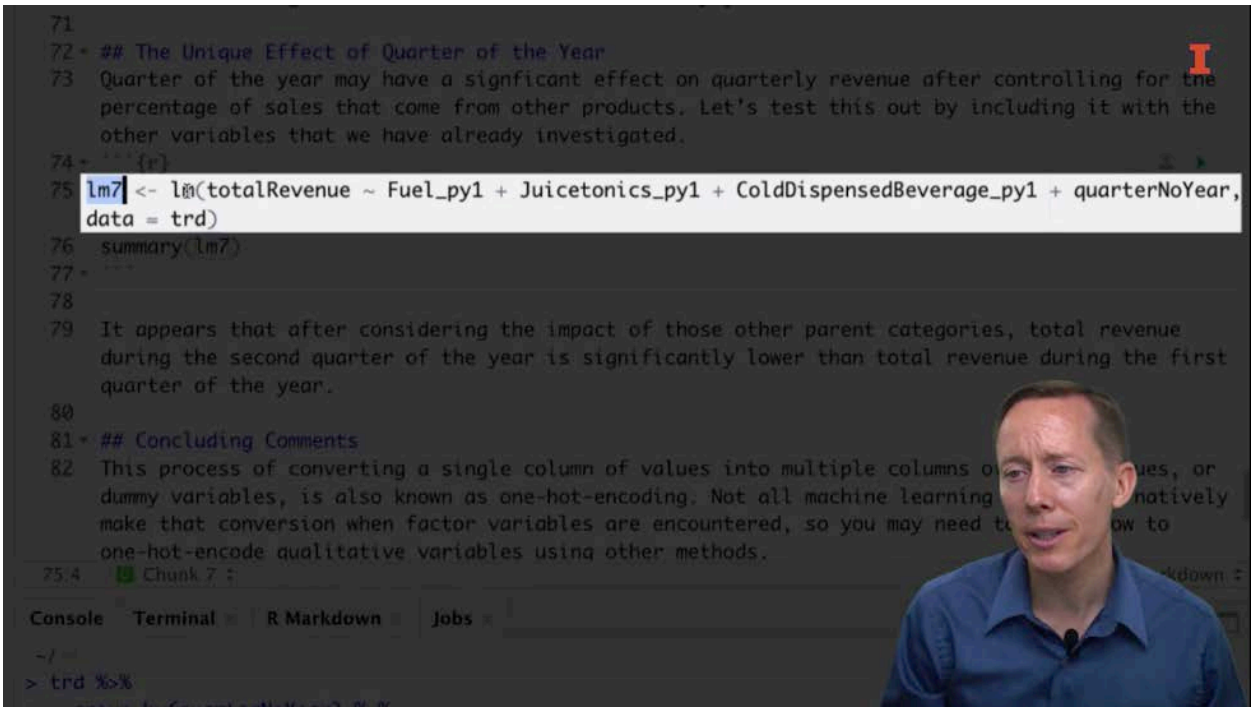
Now, the mean total revenue for the second quarter is less than the first-quarter.



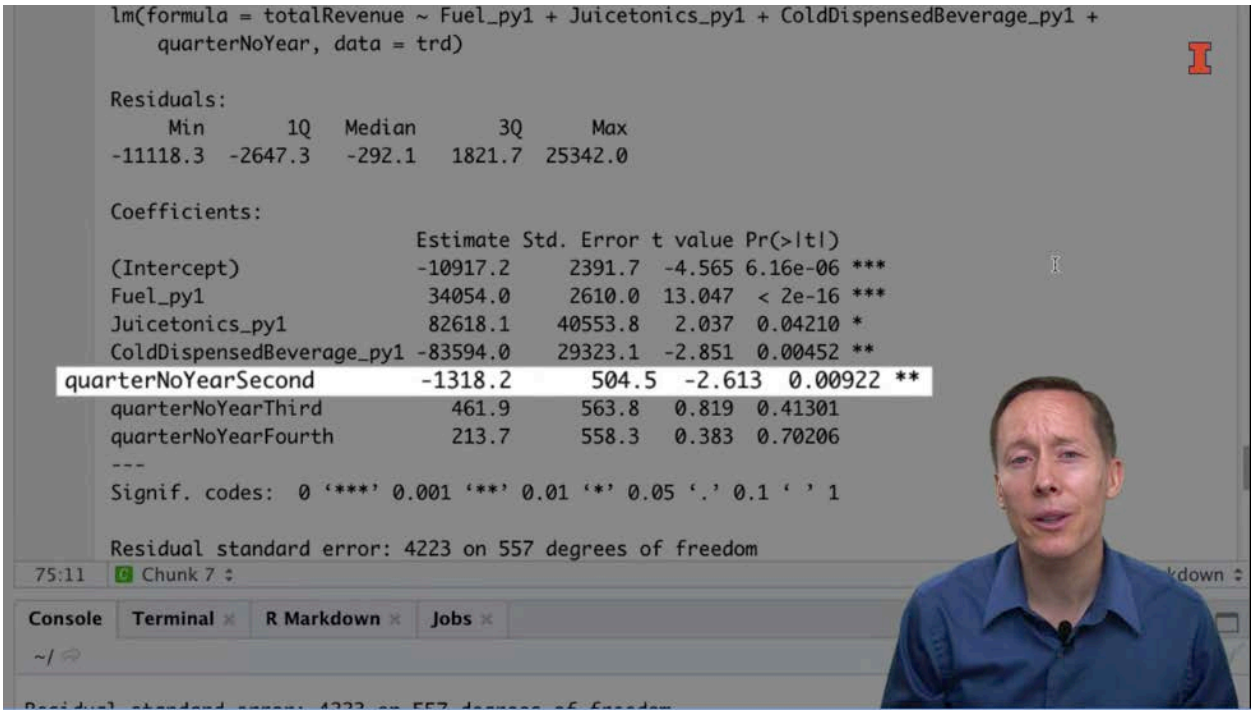
It's 10,494, almost 1,000 less or a little more than 1,000 less, so if we look at the estimate for quarterNoYear second, it's that difference between 11,538.1 and 10,494.42, and similarly with the third and fourth quarters as well.



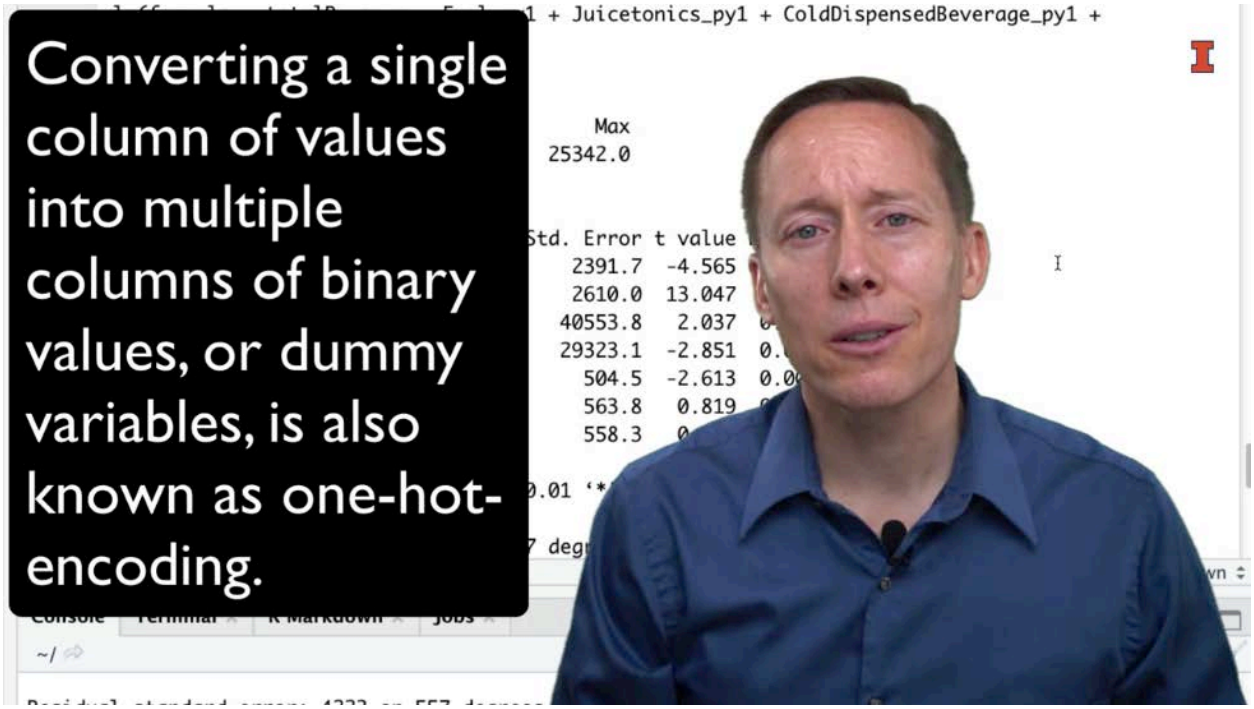
Notice that none of these coefficient estimates are statistically significant at the 0.05 level. Meaning that these differences could really just be a result of random fluctuations.



It may be the case that quarter of the year has a significant effect on quarterly revenue after controlling for the percentage of sales that come from other products. Let's test this out by including it with the other variables that we have already investigated. So I'm creating in this code chunk, a new object, lm7, in which I'm regressing total revenue on fuel_py1, juice tonics_py1, cold dispensedbeverages_py1 and quarterNoYear, and then we'll look at the summary of that model.

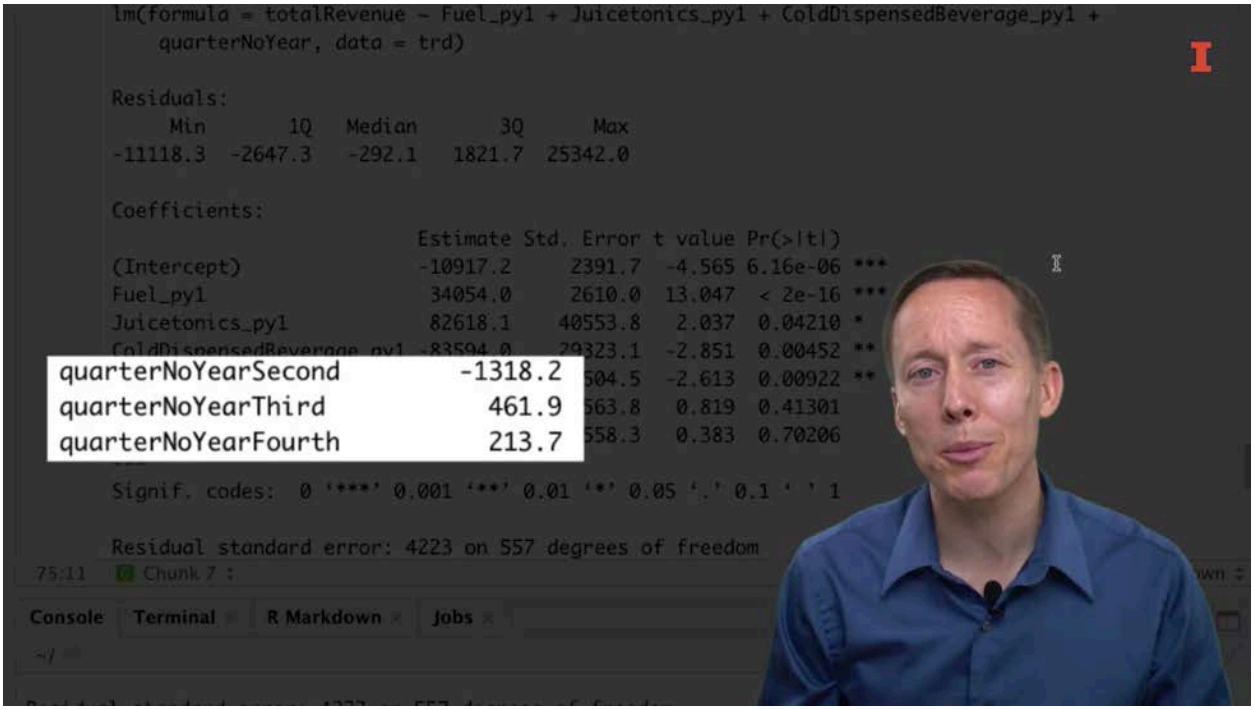


It appears that after considering the impact of those other parent categories, total revenue during the second quarter of the year is significantly lower than total revenue during the first quarter of the year.



This process of converting a single column of values into multiple columns of binary values or dummy variables is also known as one-hot encoding. Not all machine learning algorithms natively make that conversion when factor variables are encountered. So

you may need to learn how to one-hot encode qualitative variables using other methods. Creating dummy variables or one-hot encoding is a powerful way of capturing the effect of qualitative variables in machine learning models. Just remember that the interpretation is different than co-efficient estimates for quantitative variables.



Remember that the coefficients represent the difference in intercepts and not the difference in slopes.

Lesson 1-5: Review

Lesson 1-5.1 Module 1 Conclusion

At the conclusion of these lessons, I hope you have identified at least one way in which you can apply regression to a business problem that you're facing. Regression is like an exploratory data analysis robot because it can explore the independent effect of many variables, and then report back the direction and magnitude of their effects and the confidence that you can have in the effects. It can also return a model that you can use to predict future outcomes. So whether you're trying to get actionable insight by explaining relationships, making inferences, or predicting future outcomes, regression can help with all of those things. I hope you can start to appreciate the numerous ways in which regression can be applied to business analytics.

We really just introduced you to regression. It can be extended in many ways. One way that it can be extended is by adapting it to model non-linear relationships and complex interactions. Regression can also be extended for classification purposes, such as to buy or sell, or grant or deny a loan application. One of the most important benefits of gaining some experience with regression is that you now have a foundation upon which you can really start to appreciate other machine learning algorithms.