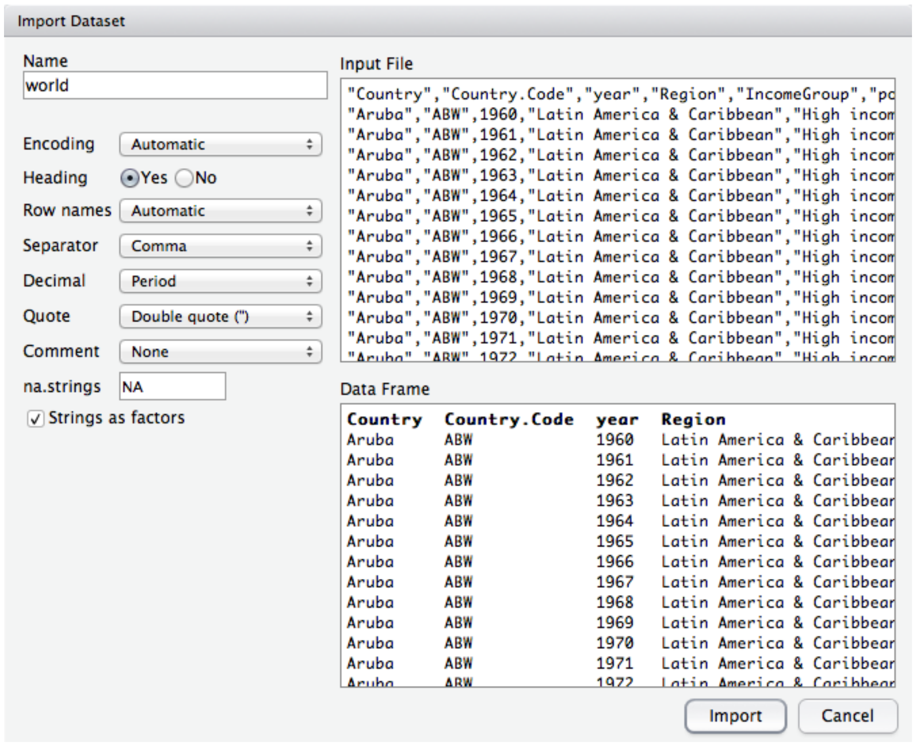
Foundations of Data Analysis

The University of Texas at Austin

R Tutorials: Week 6

Exponential and Logistic Growth Models

In this R tutoria, we're going to learn how to run both exponential and logistic models on numeric data. And for this video, we're going to be using the World Bank data set. So we need to import that first. And let's just call the data frame "world", to make our lives a little easier. And I do have the variable names in my first row. So I'm going to change my heading option to Yes.



Name: world

Encoding: Automatic

Heading: [✓ Yes] No

Row names: Automatic

Separator: Comma

Decimal: Period

Quote: Double quote (")

Comment: None

na.strings: NA

✓ Strings as factors

```
"Country","Country.Code","year","Region","IncomeGroup","pc
"Aruba","ABW",1960,"Latin America & Caribbean","High incom
"Aruba","ABW",1961,"Latin America & Caribbean","High incom
"Aruba","ABW",1962,"Latin America & Caribbean","High incom
"Aruba","ABW",1963,"Latin America & Caribbean","High incom
"Aruba","ABW",1964,"Latin America & Caribbean","High incom
"Aruba","ABW",1965,"Latin America & Caribbean","High incom
"Aruba","ABW",1966,"Latin America & Caribbean","High incom
"Aruba","ABW",1967,"Latin America & Caribbean","High incom
"Aruba","ABW",1968,"Latin America & Caribbean","High incom
"Aruba","ABW",1969,"Latin America & Caribbean","High incom
"Aruba","ABW",1970,"Latin America & Caribbean","High incom
"Aruba","ABW",1971,"Latin America & Caribbean","High incom
"Aruba","ABW",1972,"Latin America & Caribbean","High incom
```

```
Country  Country.Code  year  Region
Aruba    ABW           1960  Latin America & Caribbean
Aruba    ABW           1961  Latin America & Caribbean
Aruba    ABW           1962  Latin America & Caribbean
Aruba    ABW           1963  Latin America & Caribbean
Aruba    ABW           1964  Latin America & Caribbean
Aruba    ABW           1965  Latin America & Caribbean
Aruba    ABW           1966  Latin America & Caribbean
Aruba    ABW           1967  Latin America & Caribbean
Aruba    ABW           1968  Latin America & Caribbean
Aruba    ABW           1969  Latin America & Caribbean
Aruba    ABW           1970  Latin America & Caribbean
Aruba    ABW           1971  Latin America & Caribbean
Aruba    ABW           1972  Latin America & Caribbean
```

Figure 1: Importing Data in RStudio.

Alternatively, we can import the dataset with the "**read.csv()**" function:

```r
world <- read.csv("~/Desktop/SDSFoundations/WorldBankData.csv")
```

So this data set is a collection of various variables from countries per year from the years 1960 through 2012. And for this example, we're going to be looking at the records from the United Kingdom from years 2000 to 2009 only. So the first thing we're going to have to do is subset our data with some logical indexing statements. So let's start by first just pulling out all of the Great Britain or United Kingdom records. So I'm going to go into my World data frame, and only pull out the cases where my country code corresponds to the United Kingdom, which is just coded **GBR**. And I want to pull out every single column. So I'm just going to use the comma, and then nothing after it.

```
gbr <- world[world$Country.Code == "GBR", ]
```

In addition to subsetting just based on the country, I also only want to pull out the years 2000 through 2009. So I'm going to make another data frame – let's call it *gbr2000*. And this new data frame will be based on the *gbr* data frame, but only where *gbr* Year is greater than or equal to 2000, and also the year is less than 2010. Again, I want to pull out every single column. So with this logical statement, I will only be getting the records from Great Britain between the years 2000 and 2009.

```
gbr2000 <- gbr[(gbr$year >= 2000 & gbr$year < 2010), ]
```

So let's just double check that – click on 'gbr2000' or use:

```
View(gbr2000)
```

Here we've got United Kingdom, with years 2000 through 2009. And the variable we want to model here is this Motor Vehicles variable. So we want to see how the number of motor vehicles in the United Kingdom changes year to year.

So something to keep in mind when you're going to be running an exponential or logistic model is that you need to start your time variable at 0. So right now, our year variable goes from 2000 through 2009. So I want to make a new variable – I'm just going to call it "time" – that is equal to "gbr2000$year - 2000".

```
time <- gbr2000$year - 2000
```

So if we look at time, we're going to see a vector that just says the numbers 0 through 9. And this will represent our change for every one year, except we'll just be starting at 0, which will make the logistic model work.

```
time
```

```
##  [1] 0 1 2 3 4 5 6 7 8 9
```

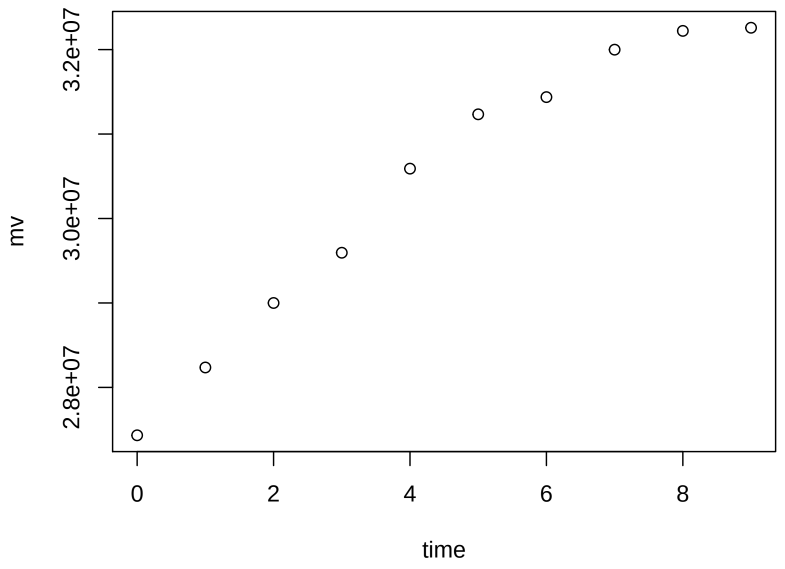
And finally, I need to make one more vector that contains the number of motor vehicles for each of those years. So I'm just going to pull out "gbr2000$motor.vehicles"" and put it in a vector just called "mv".

```
mv <- gbr2000$motor.vehicles
mv
```

```
##  [1] 27433000 28236000 28999705 29594461 30590349 31233663 31437297
##  [8] 31998958 32221383 32258677
```

So first, let's just make a scatter plot of these two variables to see the pattern of the motor vehicles variable as it changes year to year.

```
plot(time, mv)
```



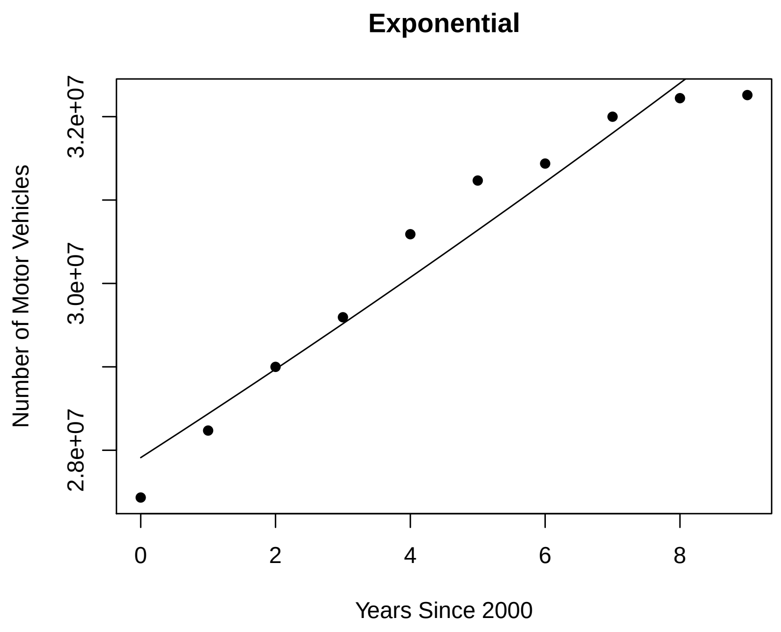
So just quickly, I can see that the number of motor vehicles is pretty steadily increasing. But it looks like it tapers off a little bit towards the end of this decade.

Now to go ahead and fit both the exponential and logistic models to this data set. We're going to use the functions "**expFit()**" for exponential, and also "**logisticFit()**" for the logistic growth model. These functions are also from the **SDSFoundations** package available on the course website. To get access to the functions, we'll need to call our SDSFoundations package with the **library()** function first.

```
library(SDSFoundations)
```

Both of the functions (expFit() and logisticFit()) take just two arguments – the independent variable first, and then the dependent variable second.

```
expFit(time, mv, xlab = "Years Since 2000", ylab = "Number of Motor Vehicles")
```

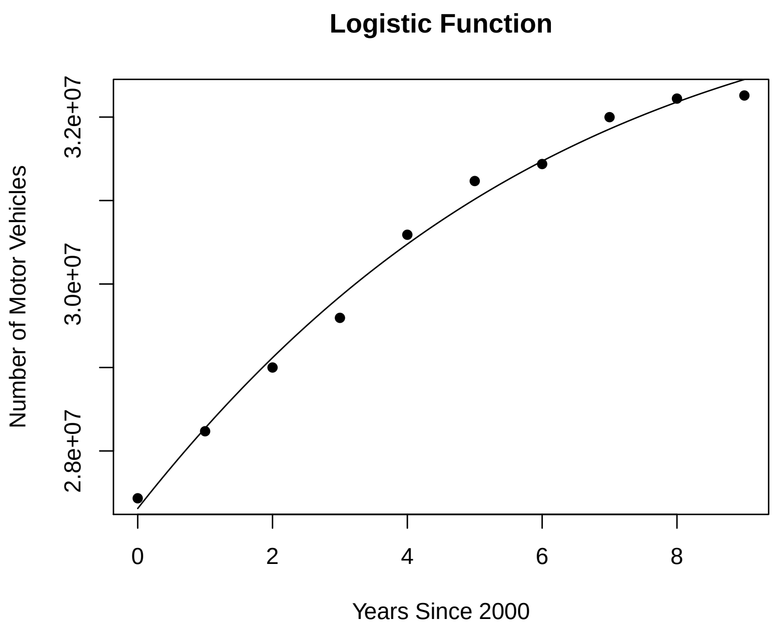


Now, if we run this command, we're going to see a couple of things. We'll see a plot – a scatter plot – of the data, and also the exponential model fit line through the data in our plot window. And over on the Console window, we'll actually see the parameters of the exponential model, as well as the R-squared value (R^2) outputted for us.

```
## Exponential Fit
##  a =  27911562
##  b =  1.01882
##  R-squared =  0.94502
```

So we can see that this model fits the data fairly well. Has an r squared of about 0.95. We can also see that this line doesn't really follow the pattern of the points very well. So let's see if we can do better with the logistic growth function. And that function in R is called "**logisticFit()**". And it takes the same arguments. And it outputs the same thing.

```
logisticFit(time, mv, xlab = "Years Since 2000", ylab = "Number of Motor Vehicles")
```



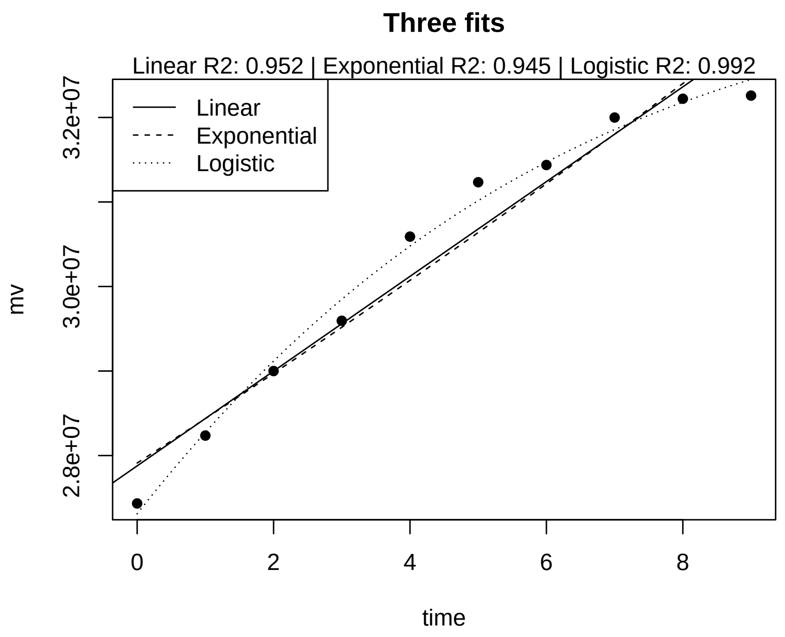
Except now, we see the logistic model fit in our plot. And then we get the parameters for our logistic equation along with the R-squared (R^2).

```
## Logistic Fit
##  C =  33759583
##  a =  0.23613
##  b =  1.21695
##  R-squared =  0.99211
```

And we can see that this model does fit better than the exponential model because it has an r squared of 0.99.

There's another function that you can use to quickly see which model might be fitting your data best. And we've talked about three models so far. The two in this video are exponential and logistic. And in the previous video, we talked about a linear model. The function "**tripleFit()**" will take the independent variable and dependent variable as arguments again. And it will produce a graph with all three of these models fit to the data, along with their R-squared (R^2) values.

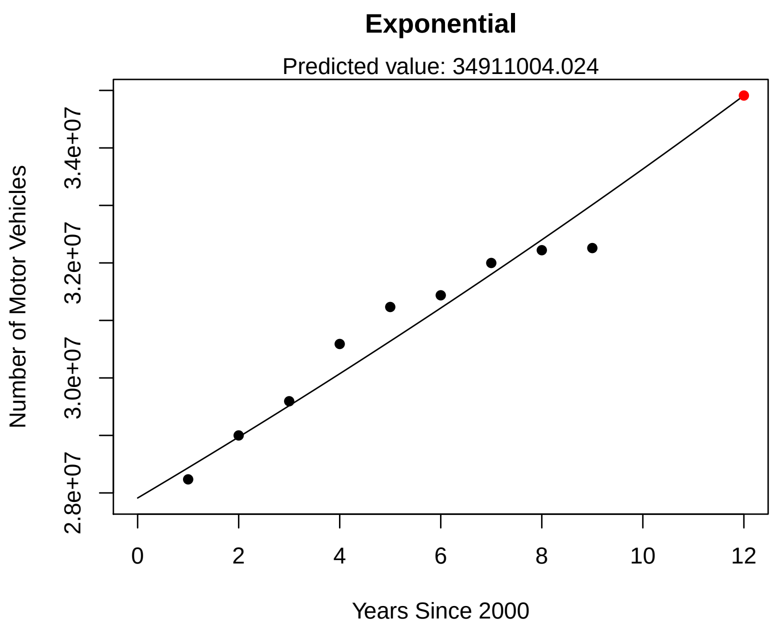
```r
tripleFit(time, mv)
```



So the linear model had an R-squared of 0.95. Exponential R-squared was very close. And we can see those two models are pretty close to each other, while the logistic is a little bit better than both of them at an R-squared of 0.99.

Finally, just to show you a couple more functions that might come in handy. You might sometimes want to predict a value of your dependent variable based on your value of your independent variable for one of these models. And to do this we can use two new functions. The first one I'll show you is called "**expFitPred()**", which is basically a prediction for your exponential model fit. It's going to take, again, the independent variable, then the dependent, but then the third argument is needed, which is the value of your independent variable at which you want to predict your dependent variable from. So say we want to predict, using the exponential model, the number of motor vehicles in Great Britain in 2012. So, remember, our time variable here is basically years since 2000. So the x value we want to give it – or the independent value we want to give it – would be 12. That would correspond to the year 2012.

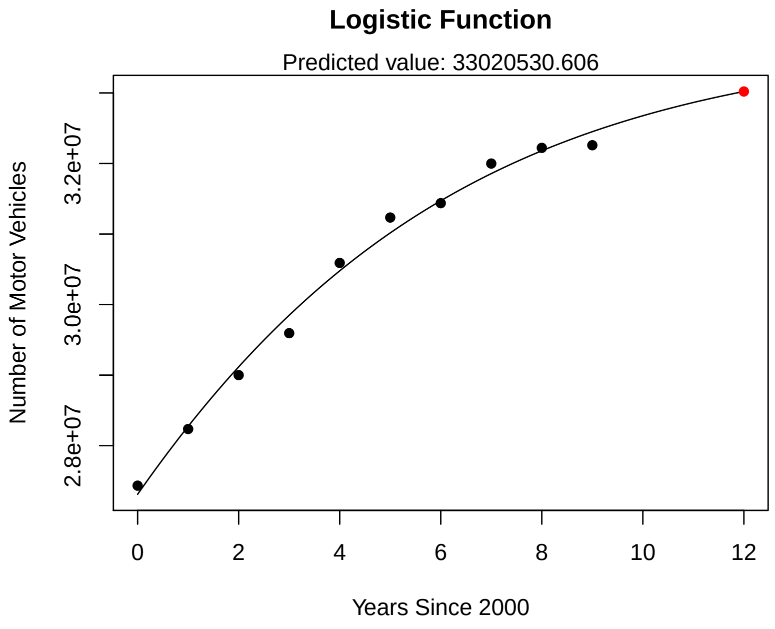
```r
expFitPred(time, mv, xval = 12, xlab = "Years Since 2000",
    ylab = "Number of Motor Vehicles")
```



If we run this, we're going to see another graph pop up. And it's going to show our current data set, our exponential model. And then it's going to extend out to the x value, or year, that we asked for in that third argument. It also gives us the actual value of our dependent variable predicted by our model at that point. So we can see that based on the exponential model, we would predict about 35 million motor vehicles in Great Britain in 2012.

Let's compare that to our logistic function prediction. And we can run that by the function "**logisticFitPred()**". It's set up exactly the same as the other function. We're going to give it the independent, then the dependent, and then our value of our independent we want to predict for.

```
logisticFitPred(time, mv, xval = 12, xlab = "Years Since 2000",
    ylab = "Number of Motor Vehicles")
```



This, again, shows our graph with our data points and the model, but then also that prediction at 2012. So based on the logistic model, we would only predict 33 million motor vehicles for Great Britain in the year 2012.