[Video Notes](https://zoom.us/rec/play/dgAIzbie4-fz4DJoDIcz3Lxsy1Zq9FUAmmmuR5CABRwswozC8VDNe1pXUS-Sa16vW5bMw9l5X5Y3YycX.mb3rfj5qZ4Bl7cN4)

<https://docs.google.com/document/d/1CBeK1W3l-jCnOISkVbbOmQPIU-kjp8OQTx_oZk2blMs/edit>

[00:01:07]

* Read the covid data file in RStudio using the R read.csv() function.

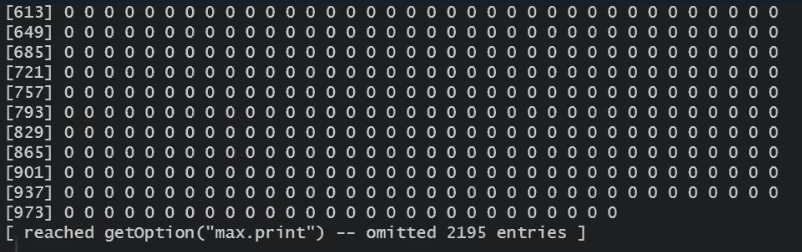


[00:02:25]

* Here we will learn how to manipulate the data before we can analyze it.

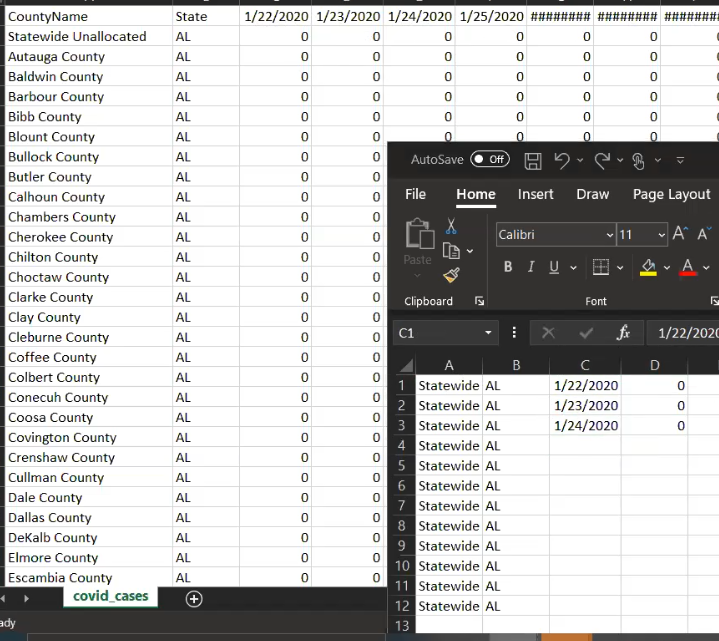
What were the cases on 01/22/2020:





[00:03:32]

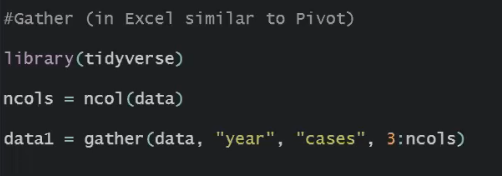
* We need to modify the file in a way where we can sum all the cases for a particular date.



* To do that, we will use the R gather() function.
* Gather takes multiple columns and collapses into key-value pairs, duplicating all other columns as needed.
* You use gather() when you notice that you have columns that are not variables.

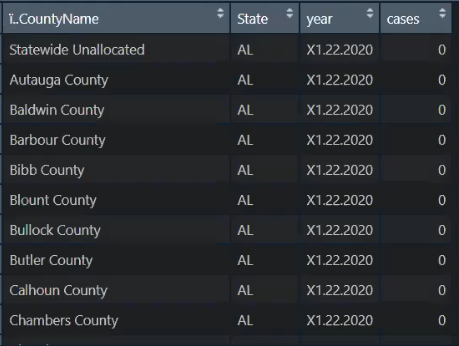
<https://tidyr.tidyverse.org/reference/gather.html>

[00:07:18]



*The ncol() function will return the number of columns present in the dataset.*

data1:



*As we can see, data1 shifted the “dates” from column names to rows.*

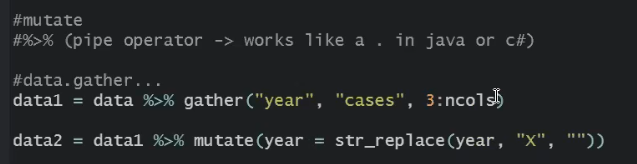
[00:11:07]

**Activity**:

* Use the gather() function on the new file.

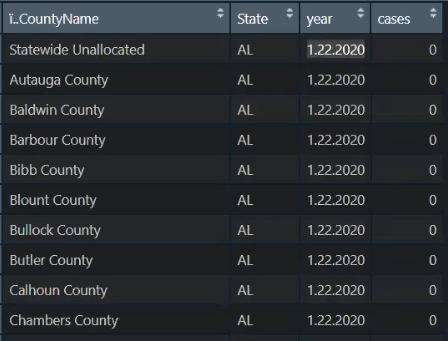
[00:23:53]

* Now each row of the year column has “X” in it and the remaining are date but not of type date.
* To solve that, we will first remove the “X”.
* For that, we will use R mutate() function.
* In R programming, the mutate function is used to create a new variable from a data set.



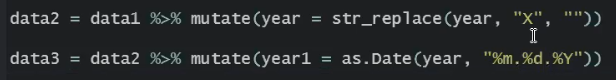
* Here we have used the str\_replace() function to replace the “X” present in the year column with nothing i.e.””.

data2:



[00:28:18]

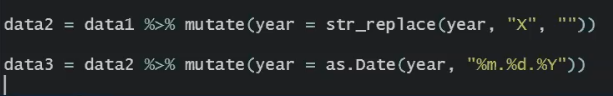
* Now we will convert the values present in the year column to Date format.
* For that, we will use R as.Date() function.
* Use the as. Date( ) function to convert character data to dates.
* The format is as. Date(x, "format"), where x is the character data and format gives the appropriate format.

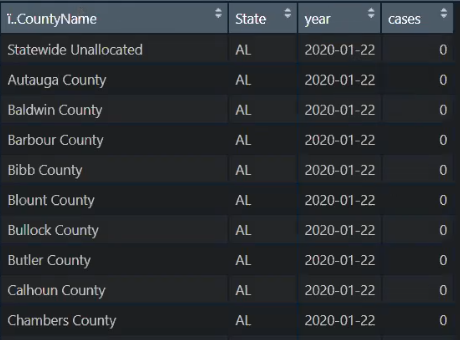


data3:



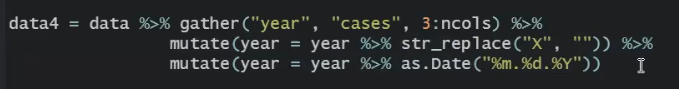
* We will mutate in the same year column instead of creating a new column.





[00:31:22]

* Using pipe operator to pipe gather(), str\_replace() and as.Date() functions into one.



[00:33:51]

* Use the R filter() function to filter the output based on given conditions.
* For eg, here we want the data of only Washington state.

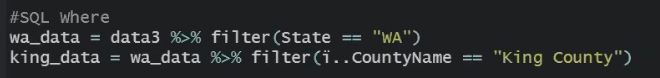


wa\_data:

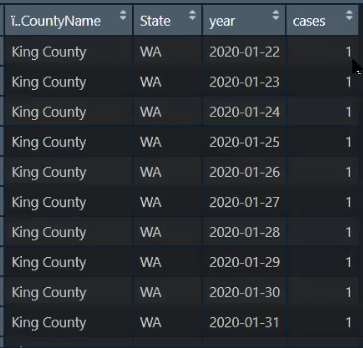


[00:35:02]

* Filtering by CountyName:

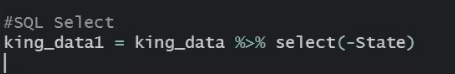


king\_data:

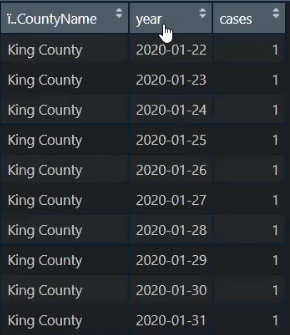


[00:36:20]

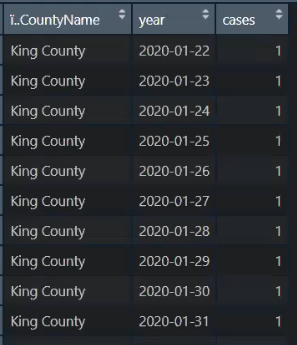
* Here we are selecting every column except State column.



king\_data1:

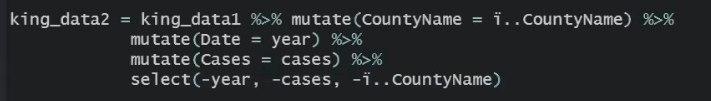




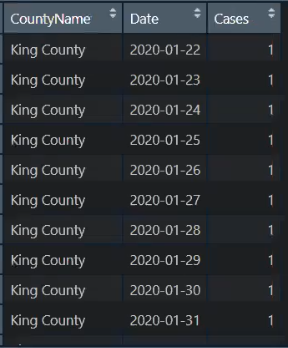


[00:37:58]

* We can use mutate() function to fix the name of the columns.



king\_data2:

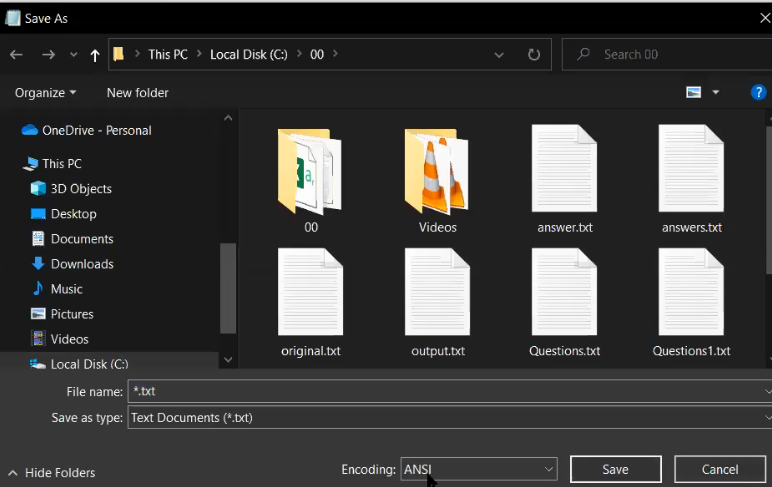


[00:41:34]

* mutate creates columns (potentially overwriting an existing column)
* select picks or removes columns
* filter filters rows by a condition

[00:42:38]

* An Excel file is saved in an encoding format that is not likable by R.
* To solve that, one way is to take that document and filter it through a notepad.
* In Notepad, go to File -> Save As -> change Encoding to ANSI:



* Another way is to use the $ symbol:



[00:43:57]

**Activity**:

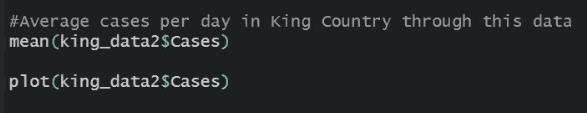
1- Take covid data

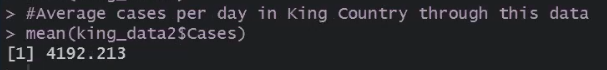
2- Gather it

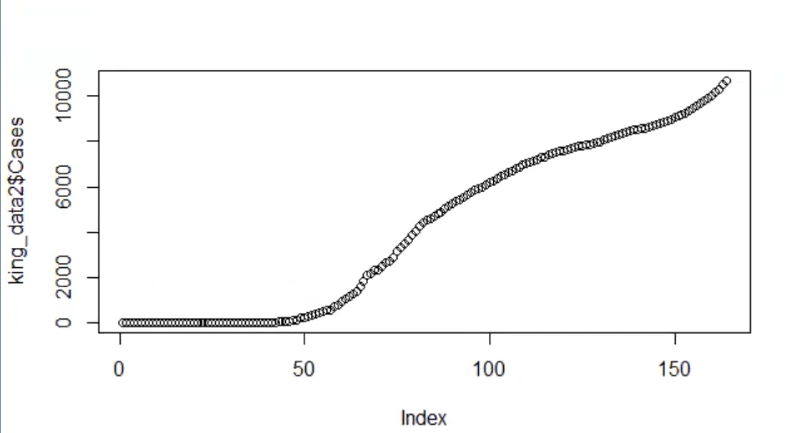
3- Keep only Florida

4- Keep cases from the first available date till June 1st, 2020  
5- Plot the trend of cases over time

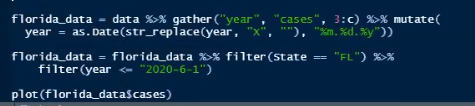
[00:45:38]

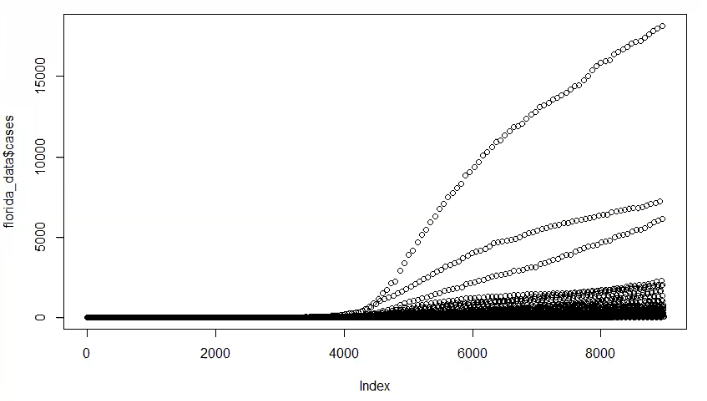






[01:07:56]





[01:09:37]

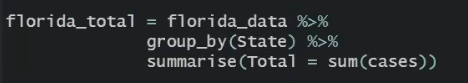
* The next operation we will perform is a group by.

group\_by():

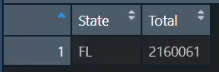
* group\_by() takes an existing tbl and converts it into a grouped tbl where operations are performed "by group".
* ungroup() removes grouping.

summarise():

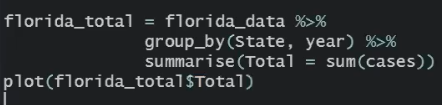
* summarise() creates a new data frame.
* It will have one (or more) rows for each combination of grouping variables; if there are no grouping variables, the output will have a single row summarising all observations in the input.



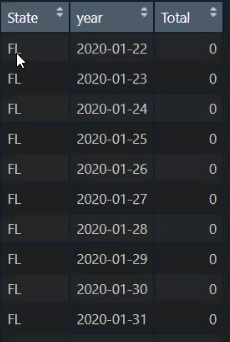
florida\_total:



[01:11:40]

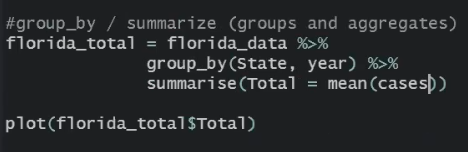


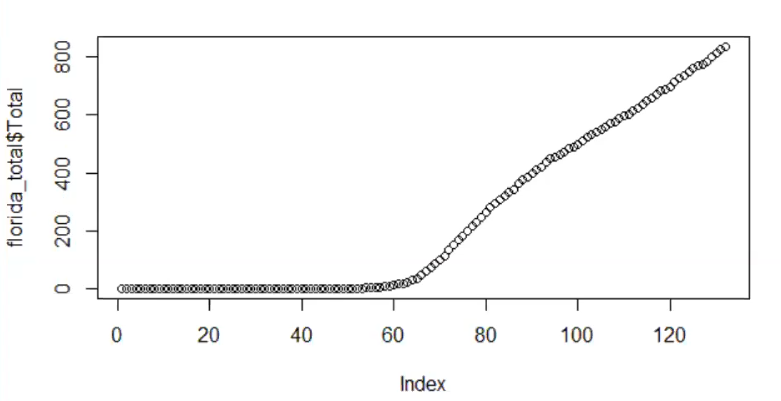
florida\_total:



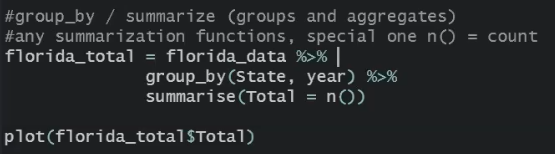


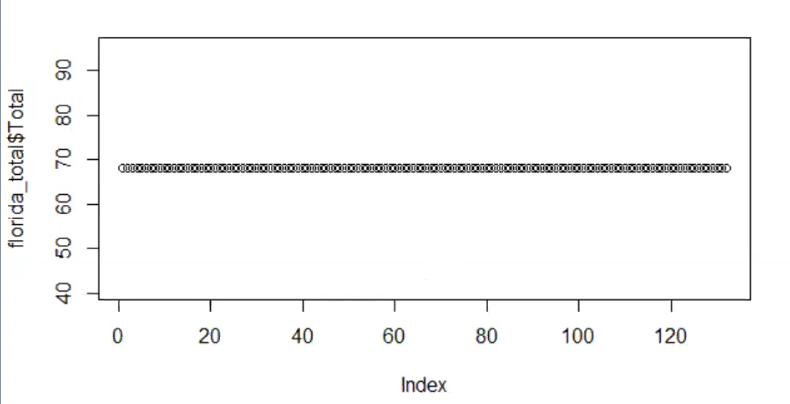
[01:13:17]



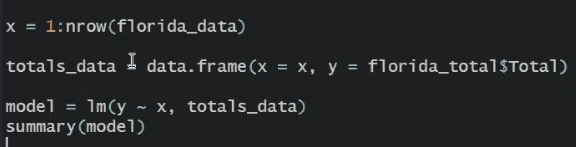


* Here n() = count:



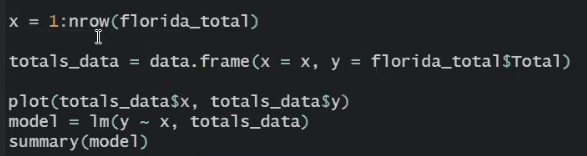


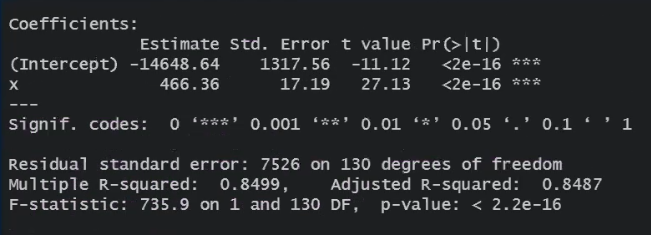
[01:14:15]

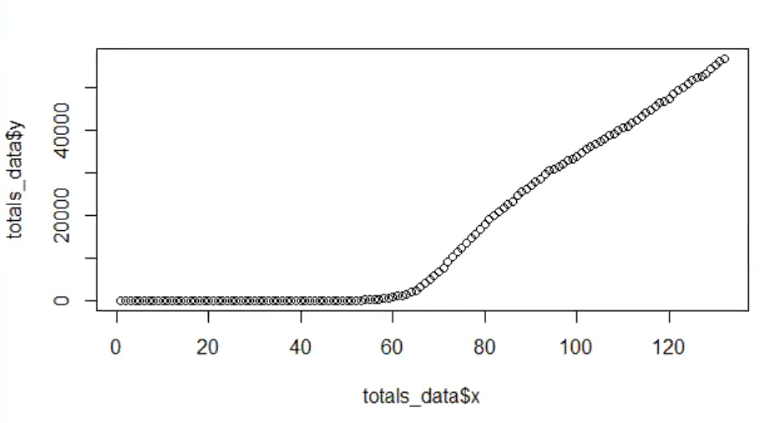




[01:18:40]



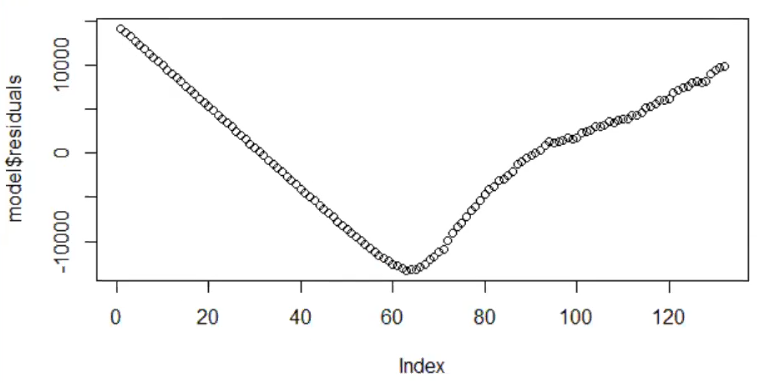




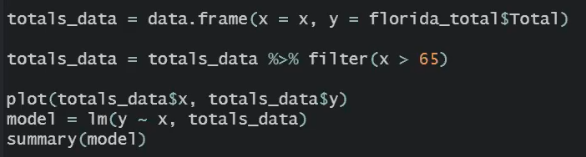
[01:19:28]

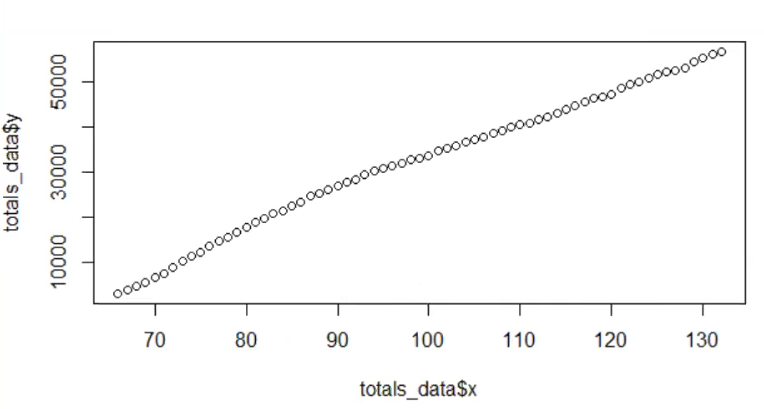
* Plotting the residuals:

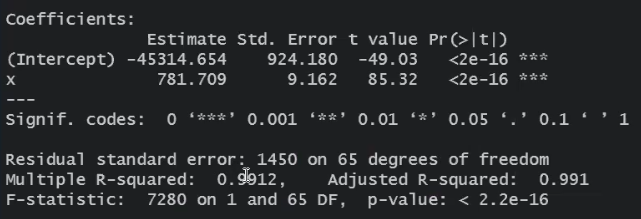




* Data after point 65:







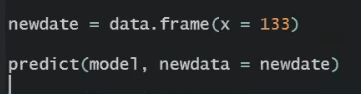
[01:20:51]

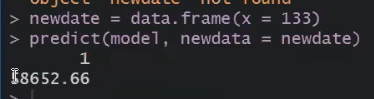
florida\_total:



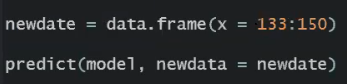
The florida data goes till 2020-06-01 i.e. 132.

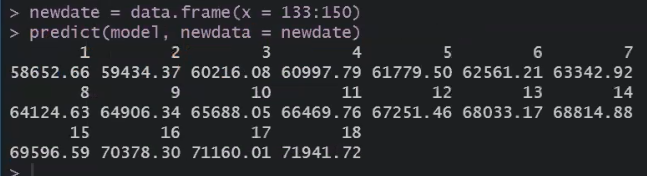
* We can use the predict() function to predict the next day:



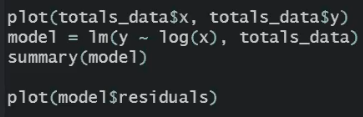


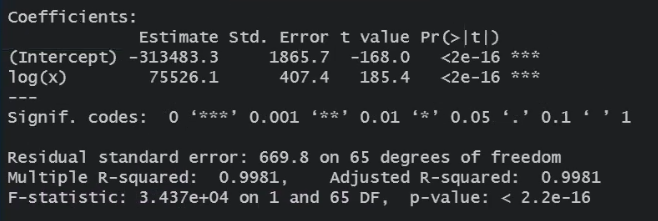
* To predict more:

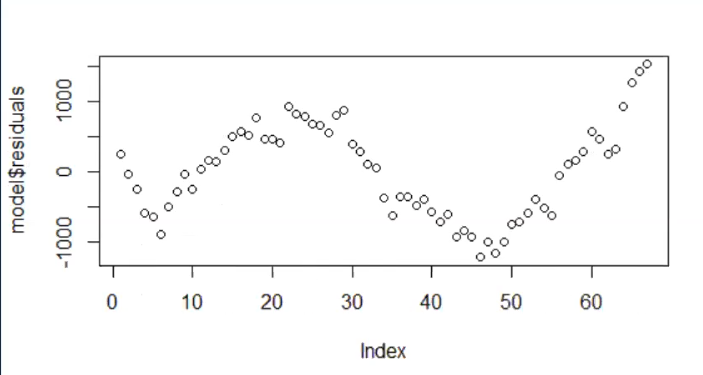




[01:37:37]



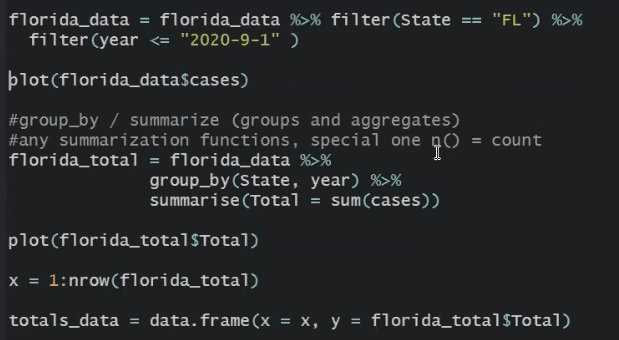


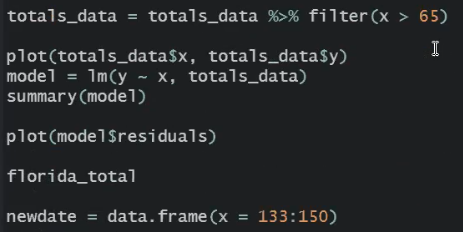


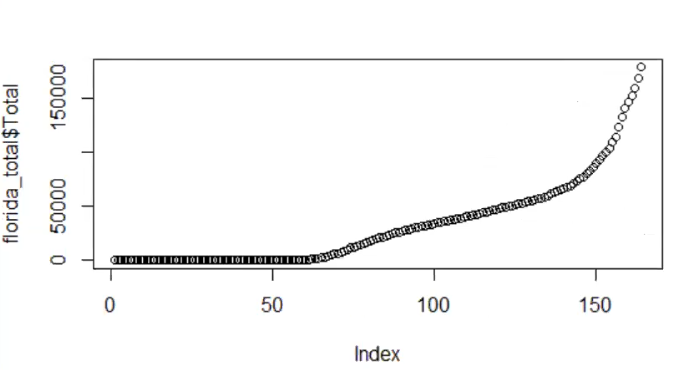
* Here after using log(x), the residuals look a little better.

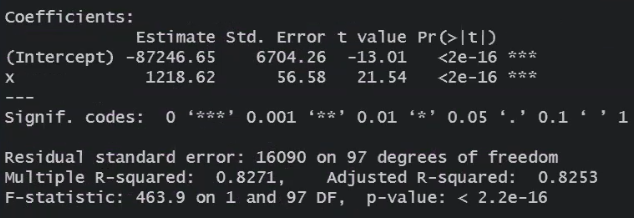
[01:39:38]

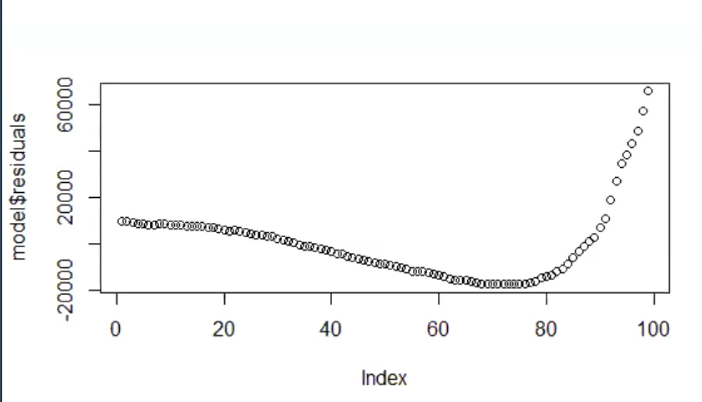
* Here we will extend the date to 2020-09-01





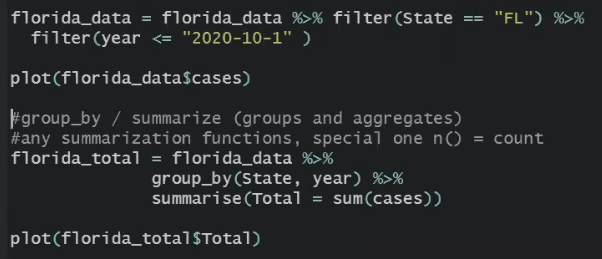


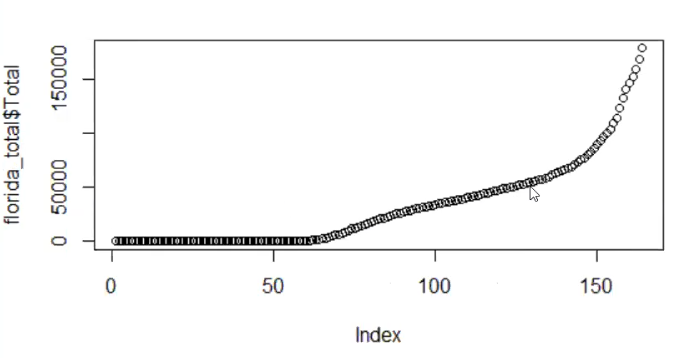


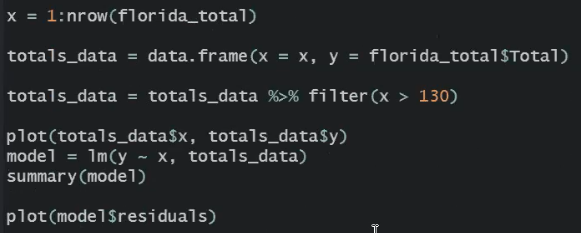


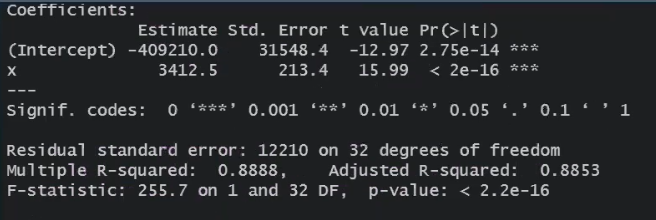
[01:41:43]

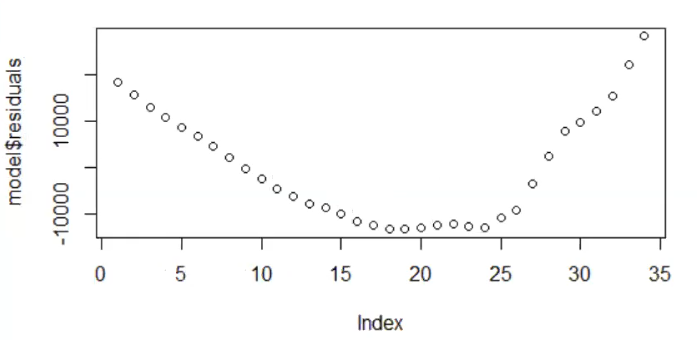
* Changing the date to 2020-10-01





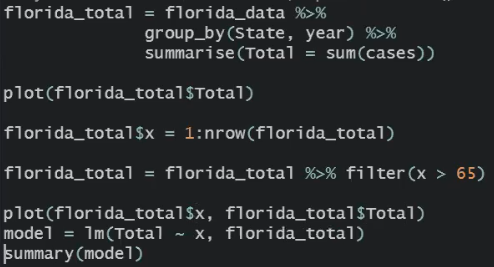


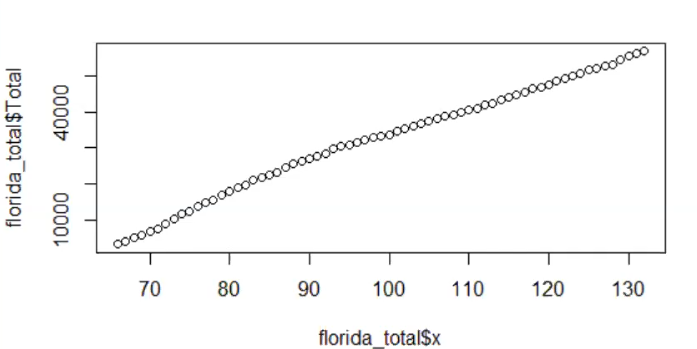


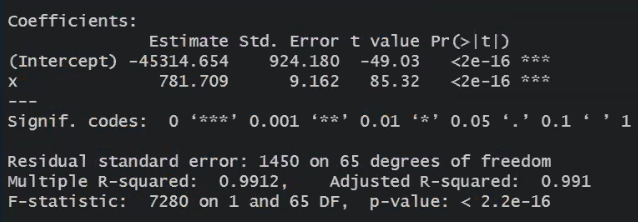


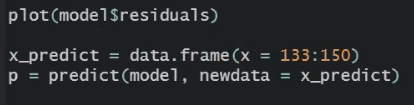
* The model is usually good only for short range of values and it becomes unpredictable after that.

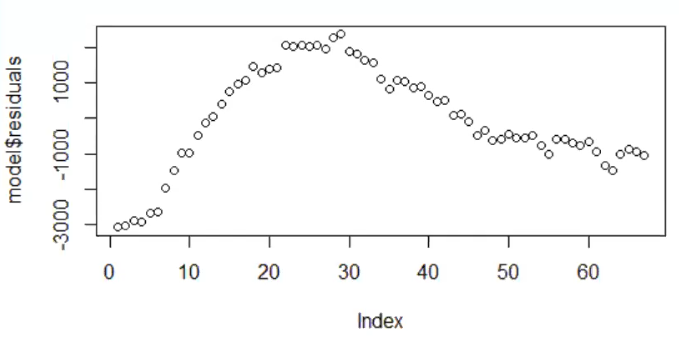
[01:43:58]











[01:47:25]

* Here we will create a new data frame named predicted.



predicted:



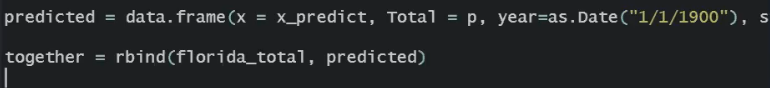
* We will add year field and Florida state.



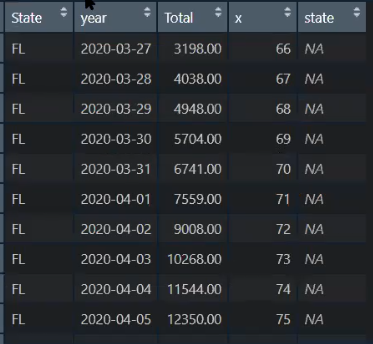


[01:48:25]

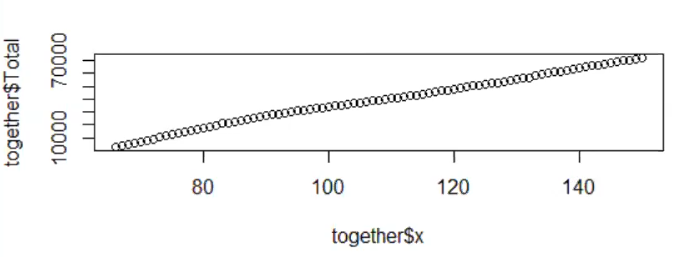
* The fields of predicted and florida\_total are now the same, so we can merge them.
* To do that, we will use the R rbind() function.
* The rbind function can be used to combine several vectors, matrices and/or data frames by rows.



together:

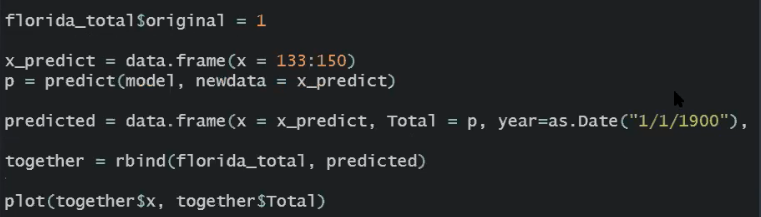


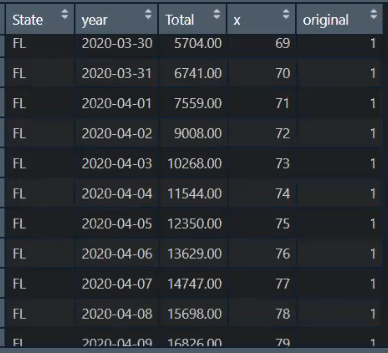




[01:50:00]

* Here we want to separate the points that come from prediction vs the points that come from the numbers that we currently have.
* To do that, we will add a new column named original.
* All the predicted values will be marked by original = 0.





*florida\_total$original = 1*

*x\_predict = data.frame(x = 133:150)*

*p = predict(model, newdata = x\_predict)*

*predicted = data.frame(x = x\_predict, Total = p, year=as.Date("1/1/1900"), State="FL", original=0)*

*together = rbind(florida\_total, predicted)*

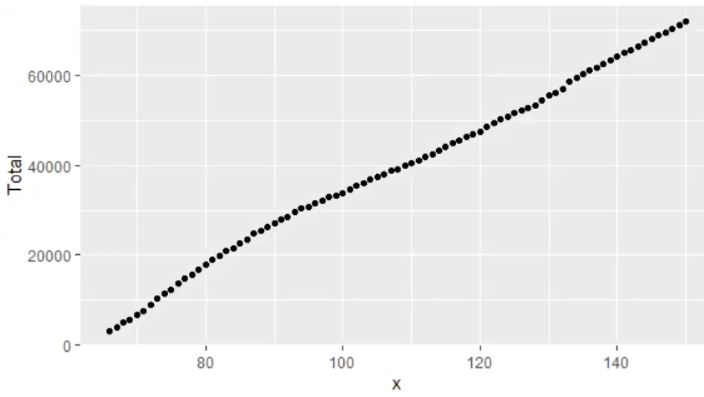
[01:51:17]

* ggplot2 is a plotting package that makes it simple to create complex plots from data in a data frame.
* It provides a more programmatic interface for specifying what variables to plot, how they are displayed, and general visual properties.

aes:

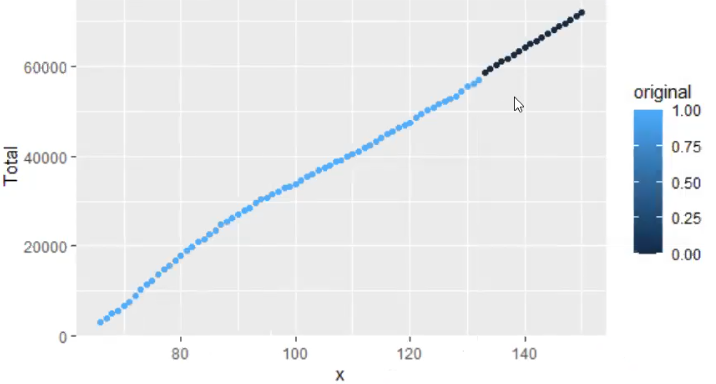
* map variables in the data to visual properties of ggplot geoms (points, bars, box plot, etc) using aesthetics (or aes) and include: color and fill, points, shape.





[01:54:04]



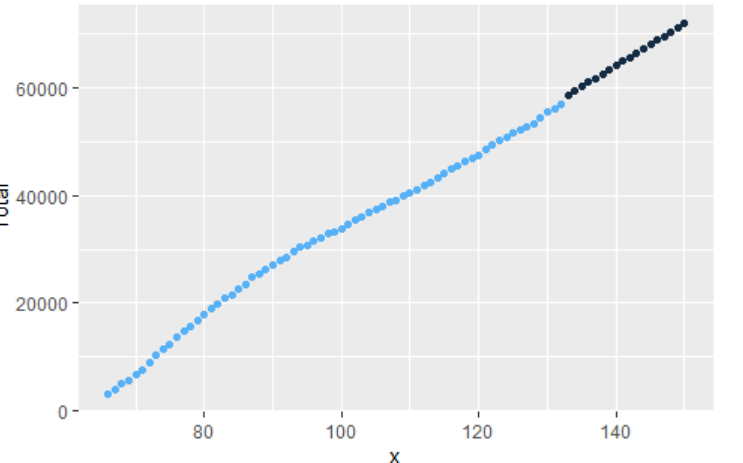


*Here the points before the prediction are in blue and those after the prediction became dark blue.*

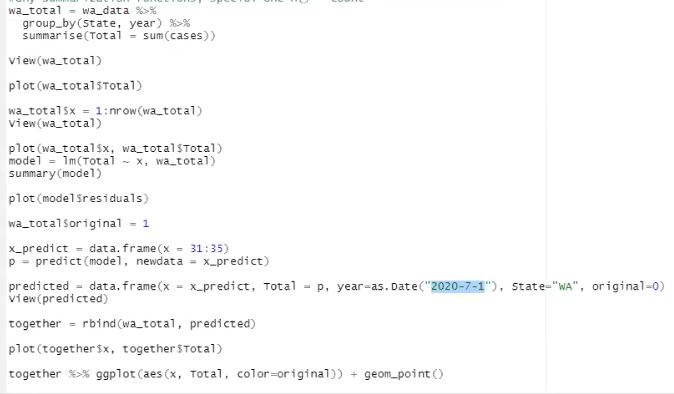
[01:55:38]

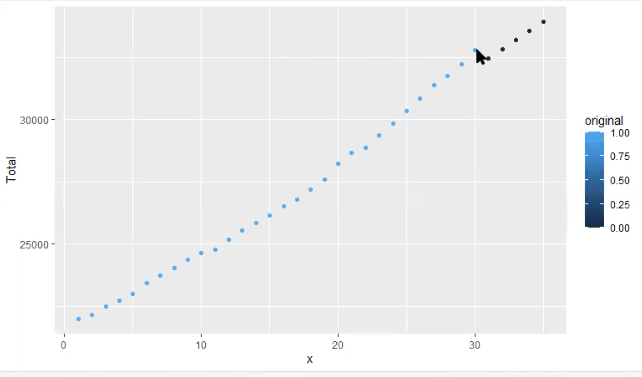
**Activity**:

Use the data from Washington in June to predict the cases in the first five days of July using a linear model. Produce a chart like this:

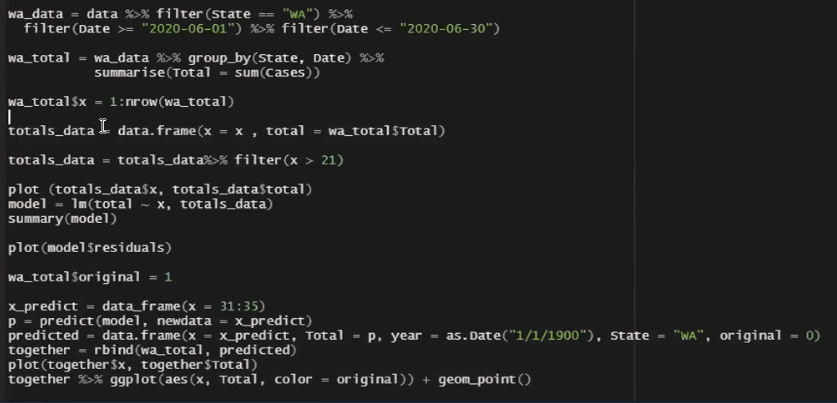


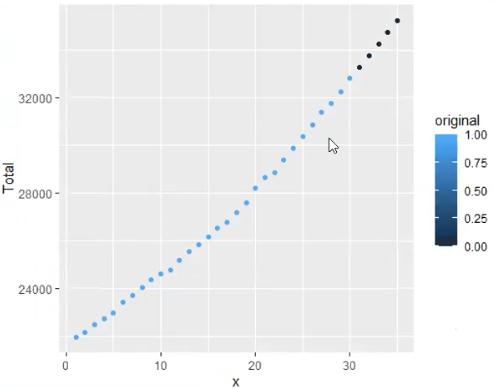
[02:22:30]





[02:24:50]





[02:26:41]

* Now changing the original to TRUE and the original value for predicted values to FALSE.





