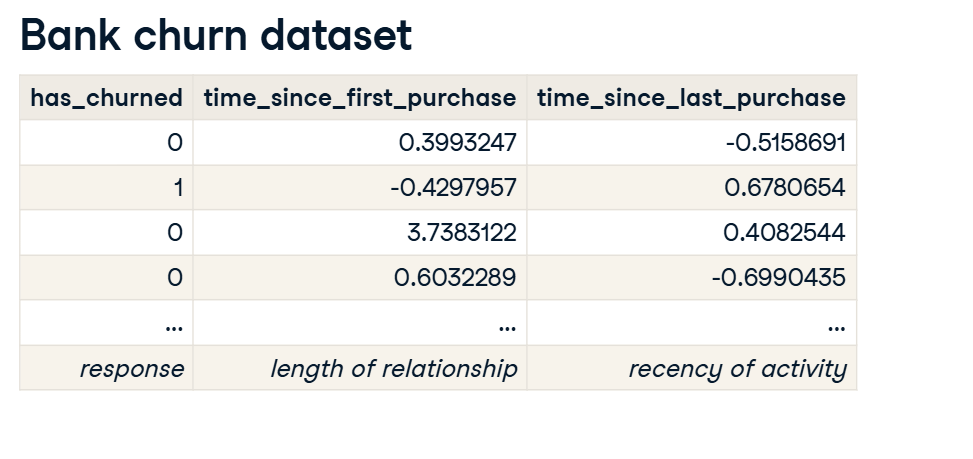
**Why you need logistic regression**

The datasets you've seen so far all had a numeric response variable. Now we'll explore the case of a binary response variable.

**2. Bank churn dataset**

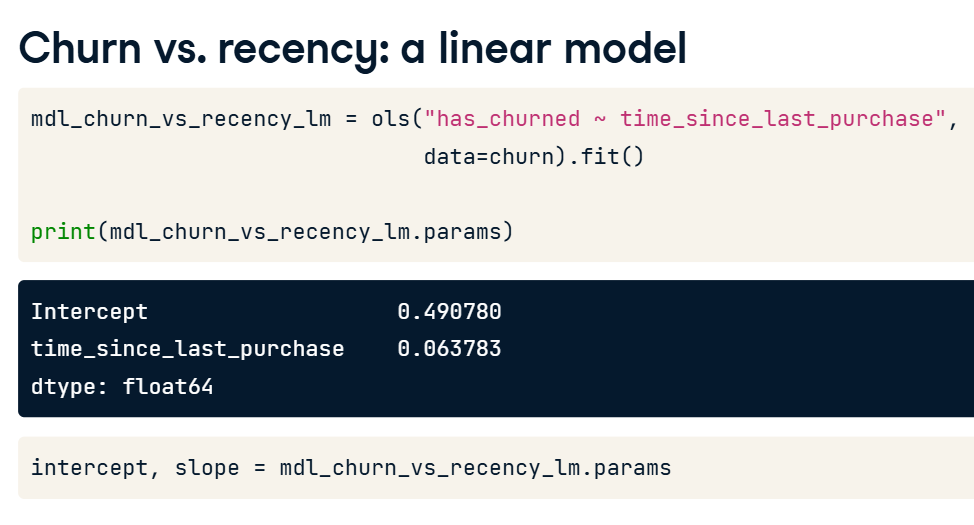
Consider this dataset on churn at a European financial services company in 2006. There are 400 rows, each representing a customer. If the customer closed all accounts during the time period, they were considered to have churned, and that column is marked with a one. If they still had an open account at the end of the time period, has\_churned is marked with a zero. Using one and zero for the response instead of a logical variable makes the plotting code easier. The two explanatory variables are the time since the customer first bought a service and the time since they last bought a service. Respectively, they measure the length of the relationship with the customer and the recency of the customer's activity. The time columns contain negative values because they have been standardized for confidentiality reasons.

1. 1 <https://www.rdocumentation.org/packages/bayesQR/topics/Churn>



**3. Churn vs. recency: a linear model**

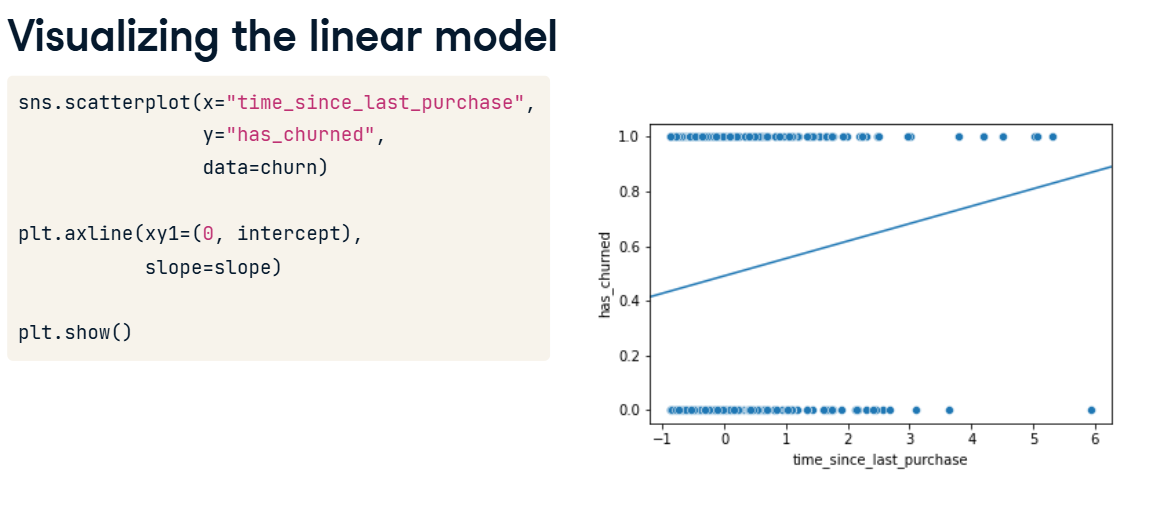
Let's run a linear model of churn versus recency and see what happens. We can use the params attribute to pull out the intercept and slope. The intercept is about point-five and the slope is slightly positive at zero-point-zero-six.



**4. Visualizing the linear model**

01:22 - 01:45

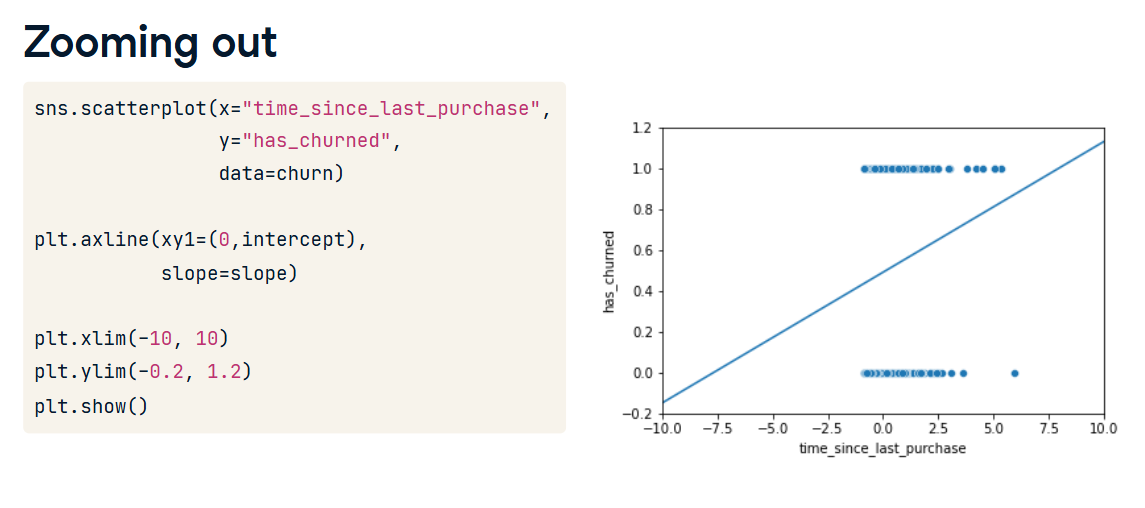
Here's a plot of the data points with the linear trend. I used plt dot axline rather than sns dot regplot so the line isn't limited to the extent of the data. All the churn values are zero or one, but the model predictions are fractional. You can think of the predictions as being probabilities that the customer will churn.



**5. Zooming out**

01:45 - 02:04

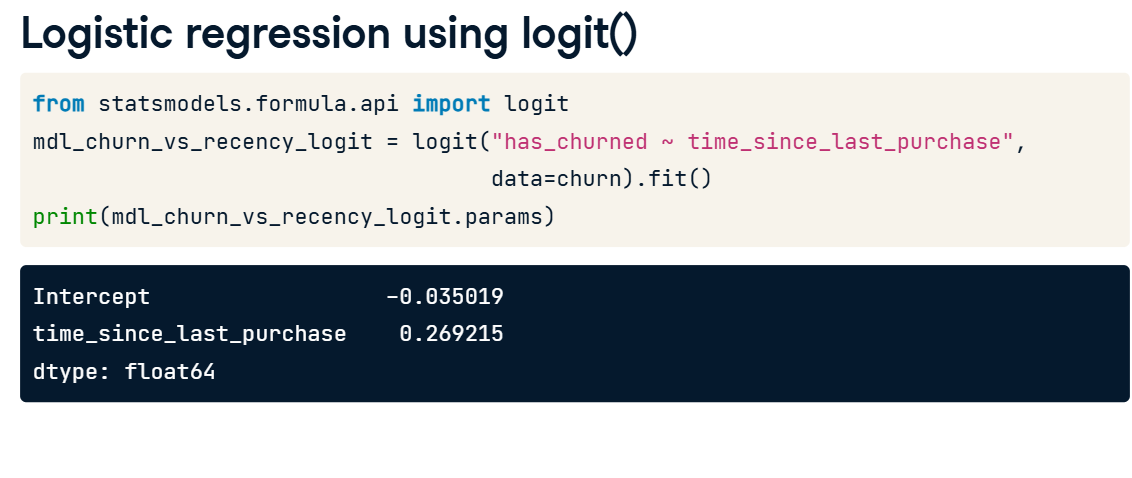
Zooming out by setting axis limits with xlim and ylim shows the problem with using a linear model. In the bottom-left of the plot, the model predicts negative probabilities. In the top-right, the model predicts probabilities greater than one. Both situations are impossible.



**6. What is logistic regression?**

02:04 - 02:25

The solution is to use logistic regression models, which are a type of generalized linear model, used when the response variable is logical. Whereas linear models result in predictions that follow a straight line, logistic models result in predictions that follow a logistic curve, which is S-shaped.



**7. Logistic regression using logit()**

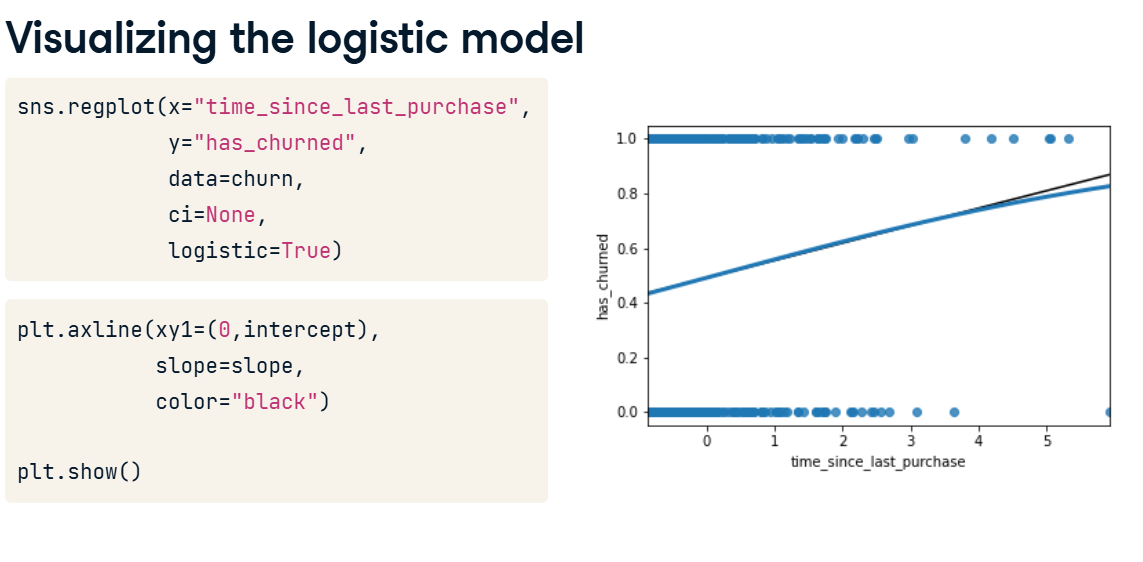
02:25 - 03:06

To run a logistic regression, you need a new function from statsmodels. From the same statsmodels dot formula dot api package, import the logit function. This function begins the process of fitting a logistic regression model to your data. The function name is the only difference between fitting a linear regression and a logistic regression: the formula and data argument remain the same, and you use the dot fit method to fit the model. As before, you get two coefficients, one for the intercept and one for the numerical explanatory variable. The interpretation is a little different; we'll come to that later.

**8. Visualizing the logistic model**

03:06 - 03:31

Let's add the logistic regression predictions to the plot. regplot will draw a logistic regression trend line when you set the logistic argument to True. Notice that the logistic regression line, shown in blue, is slightly curved. Especially when there's a longer time since the last purchase values, the blue trend line no longer follows the black, linear trend line anymore.



**9. Zooming out**

Now zooming out shows that the logistic regression curve never goes below zero or above one. To interpret this curve, when the standardized time since last purchase is very small, the probability of churning is close to zero. When the time since last purchase is very high, the probability is close to one. That is, customers who recently bought things are less likely to churn.

