Intro to Data Science
Linear Regression
Intermediate

1 Log transformation feature engineering for non-linear regression

File to use: `Non Linear Regression`

The `initial` worksheet shows the initial dataset from a marketing dept. Ad spend is the independent variable and revenue is the dependent variable. The plotted graph implies a non-linear relationship between these two variables. This is a standard logarithmic curve that occurs many times in the real world, especially in the business intelligence space. This is a classic illustration of diminishing returns where early ad dollars are spent have a big impact on revenue, and then this impact starts to slow as more and more money is spend on ad.

Create the first linear regression model using the original dataset (independent variable X - `ad_spend` and dependent variable Y - `revenue`), and place the regression table below the Linear Regression Output highlighted row. Use the two coefficients (Intercept and ad_spend) to create the linear regression equation,

$$\hat{y} = b_0 + b_1 X_1$$

Dependent variable

y-intercept (constant)

Slope coefficient

Independent variable

which you can then place in the first cell of the forecast Linear column, for e.g.

```
=$J$42+($J$43*C4)
```

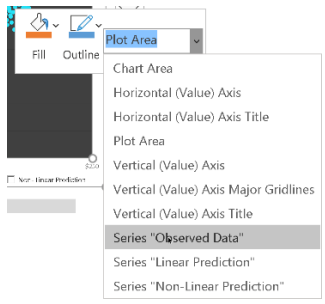
and then subsequently populate the rest of the column with the predicted values

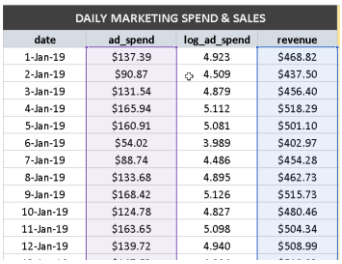
Now tick the Linear Prediction checkbox to plot a linear graph using these newly predicted values - this is essentially the linear regression model. Notice that while it is not a bad fit, the starting and ending portion of the graph deviates quite a bit from the original data points in that area.

To account for the logarithmic curve, we can perform feature engineering. More specifically, we can use a natural log transformation on the independent variable and use this transformed variable value to generate our regression model instead.

Click on the + at the top to unhide the hidden D column, which contains the transformed variable using the Excel LN function to perform a natural log transformation.

Select the portion of the graph with the multiple blue dots (the original datapoints), or click anywhere the graph area and select Series "Observed Data", which should highlight the two columns used to plot the multiple blue dots (`ad_spend` and `revenue`).

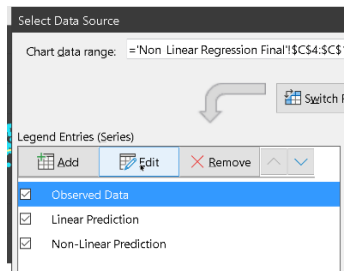


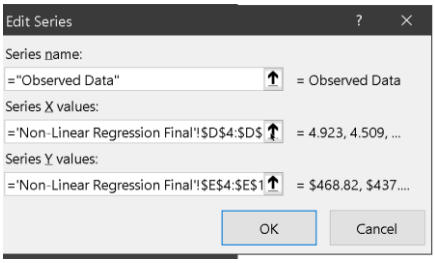


DAILY MARKETING SPEND & SALES

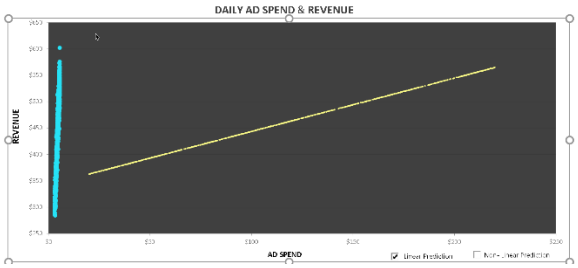
date	ad_spend	log_ad_spend	revenue
1-Jan-19	$137.39	4.923	$468.82
2-Jan-19	$90.87	4.509	$437.50
3-Jan-19	$131.54	4.879	$456.40
4-Jan-19	$165.94	5.112	$518.29
5-Jan-19	$160.91	5.081	$501.10
6-Jan-19	$54.02	3.989	$402.97
7-Jan-19	$88.74	4.486	$454.28
8-Jan-19	$133.68	4.895	$462.73
9-Jan-19	$168.42	5.126	$515.73
10-Jan-19	$124.78	4.827	$480.46
11-Jan-19	$163.65	5.098	$504.34
12-Jan-19	$139.72	4.940	$508.99

Right click on and select Select Data option, and in the Select Data Source dialog box, highlight Observed Data and click Edit. In the Edit Series dialog box, select Series X values and base that on the new column D (`log_ad_spend`), which is the natural log of all the values in the original independent variable, column C (`ad_spend`). Click Ok to accept everything.

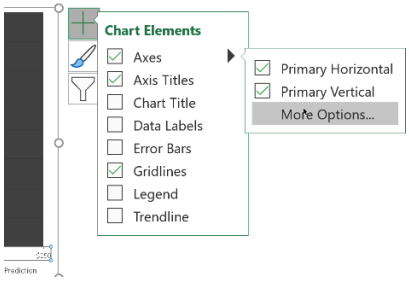


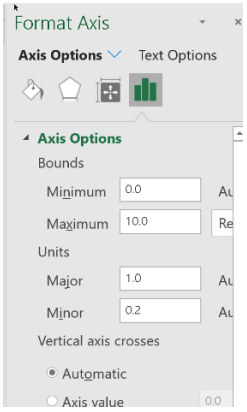


You should now see that plotting the natural log of the independent variable against the dependent variable gives us a linear graph, which fits very well with a regression model.

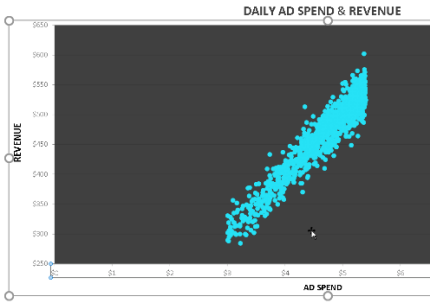


You can select the graph -> Axes -> More Options. In the Format Axis -> Axis Options dialog box, change the Maximum value from 250 to 10.





This show us a very clear linear relationship between the transformed independent variable (`log_ad_spend`) and the revenue.



Now, you can switch back to the Axis options of a maximum value of 250, and then change the Series X values back to the original column C (`ad_spend`) to restore the graph shape.

Now, we create the 2nd linear regression model using the transformed dataset (independent variable X - `log_ad_spend` and dependent variable Y - `revenue`) and place the regression table below the Non-Linear Regression Output highlighted row.

Notice the R Square for this regression model is higher than that for the first model, showing clearly how the transformation of the single independent variable (Feature Engineering) improves model predictive accuracy.

Once again, use the two coefficients (`Intercept` and `log_ad_spend`) to create the linear regression equation, and populate the Forecast (Non-Linear) column with values from this equation in the same way as we had done previously for the Forecast (Linear) column.

Now tick the Non-Linear Prediction checkbox to plot a non-linear graph using these newly predicted values - this is essentially the non-linear regression model. Notice that it is now visually a much better fit that the original linear regression model.

2 Dummy variable feature engineering for categorical variables

File to use: `Deliveries-pt1`

Consider a sample case study that is common in supply chain management / planning.

The `initial` worksheet shows the initial dataset. Here, the variable we are interested to study / predict is the total delivery time (in minutes) for a package involving a truck. One of the independent variables is the Region, which is categorical. The categories A and B can represent any 2 real destinations, for e.g. London and New York, or Selangor and Penang.

First, try to generate a regression model using Minutes as the Y variable and Region, Parcels and TruckAge as the X variables. Notice that Excel will display an error indicating a regression model cannot be built from non-numerical data (the Region column).

Make a copy of the `initial` worksheet and create a new column called REGA which is the dummy variable column. We will use this dummy variable column in our linear regression equation (as shown in `final-dummy-variable-v1`. When REGA is 0, this indicates the original category of B, and when REGA is 1 this indicates the original category of A.

Key point: For a categorical column with X possible categories, we need X - 1 dummy variable columns to represent all these categories. Therefore, for Region column that only has 2 categories (A, B), we only need to have 1 dummy variable column.
We use the IF statement to generate the values for the dummy variable column REGA from the original REGION column. We then generate the regression model using this dummy variable column REGA as one of the independent variables (the other 2 being Parcels and TruckAge).

The final result should be identical to the worksheet `final-dummy-variable-v1`.

For a dummy variable, the coefficients of dummy variables are to be interpreted with a reference to the base category, which is the category that was initially encoded as 0 (category B). The interpretation is again with respect to the increase in the value of the dependent variable (the total delivery time in minutes). Here we interpret that **deliveries made to region A takes beta1 (106.84) minutes longer than it takes to region B** (with parcels and truck age staying the same). This would generally imply that region A is further away from region B from the origin of travel

Note that we would still get exactly the same interpretation if we used a different encoding from the one here: for e.g. the dummy variable column is called REGB instead, and when REGB is 0, this indicates the original category of A and when REGB is 1 this indicates the original category of B (you can create another copy of `initial` to do this). Now when we generate the regression model (which should look like `final-dummy-variable-v2`, the value of beta1 of -106.84 indicates that the deliveries made to region B are 106.84 minutes SHORTER (because of negative value) than it takes

to region A (the base category in this instance). This is the identical interpretation of the first encoding approach.

In other words, it doesn't matter which category you represent as 0 and which one you represent as 1: as long as we keep to the key point: **For a categorical column with X possible categories, we need X - 1 dummy variable columns to represent all these categories**.

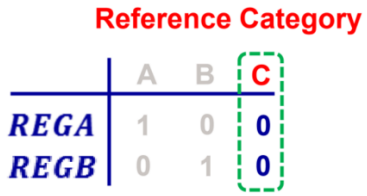
3 Dummy variables for multiple categories

File to use: `Deliveries-pt2`

This is similar to the previous case study, except now the `Initial` worksheet contains a categorical column Region that has 3 distinct categories (instead of 2). Again, we follow the key rule:

For a categorical column with X possible categories, we need X - 1 dummy variable columns to represent all these categories. Therefore, for Region column that has 3 categories (A, B, C), we only need to have 2 dummy variable columns.

Make a copy of the `initial` worksheet and create 2 new columns called REGA and REGB, which are now the dummy variable columns that we will use in our linear regression equation (as shown in `final-dummy-variable-v1.)`
The values of the REGA and REGB and how they correspond to the 3 categories in the original Regions column is as follows:



Notice that both the columns will have the value 0 for the reference category of C.

Similar to the previous case of a single dummy variable, we can use the IF statement to generate the values for the 2 dummy variable columns REGA and REGB with REGC as our reference category.

From the central warehouse, trucks leave to drive to region A, region B or region C. These regions are at varying distances from the warehouse and the time it takes for the truck to reach a particular region may vary. This duration is termed as fixed time to make parcel deliveries because the trucks have yet not delivered any parcel. They simply have reached the region from the warehouse carrying parcels for delivery in that region. The differences in these fixed times to make parcel deliveries across the three regions is captured by the beta 1 and beta 2 coefficients.

Beta 1 gives us the difference in the fixed time to make deliveries across Region A, as compared to Region C. So, we can conclude that delivery time to Region A is 107 minutes longer than Region C, starting from the same origin (central warehouse).
Beta 2 gives us the difference in the fixed time to make parcel deliveries across Region B as compared to Region C. So, we can conclude that delivery time to Region B is 1.2 minutes longer than Region C (in other words, delivery time to both regions is nearly identical).

Once a truck has reached a particular region, Region A, Region B, or Region C, it then makes those partial deliveries across various customers in that region. The time it takes to deliver each extra parcel within the region can be thought of as the marginal time to deliver the parcel, and is captured by the beta 3 coefficient (9.92).

We can use these coefficients in the linear regression equation to make a prediction for the delivery time for a fixed number of parcels (25) and using a truck that is 10 years old. The results are shown in `final-dummy-variable-v1`

The way we interpret p-values of dummy variable columns is as follows:
If the P-value for a dummy variable column coefficient is lower than significance level α, this means that the dependent variable value for the category represented by that column is significantly different from the base / reference category. Otherwise, it is not.

For e.g. the p-value for REGA (1.09388E-16) is much smaller than the significance level α (0.05) for the default CL of 95%. This means that delivery time to REGA is significantly different from the base category of REGC (which is also indicated by the high value of the coefficient of REGA in the interpretation that we described just now).

However, the p-value of REGB (0.156255218) is much higher than the significance level α (0.05). This means there is no significant difference between the delivery time to REGB compared to the base category of REGC (which is also indicated by the low value of the coefficient of REGB in the interpretation that we described just now).

Note that we could also have employed a completely different encoding for our dummy variable columns then what we have used (for e.g. having REGB and REGC columns instead of REGA and REGB columns) and still obtain the identical interpretation and result.

	A	B	C
REGB	0	1	0
REGC	0	0	1

You can create another copy of `initial` to do this. Now when we generate the regression model (which should look like `final-dummy-variable-v2)`, the value of beta2 of -107.84 indicates that the deliveries made to region C are 107 minutes SHORTER (because of negative value) than it takes to region A (the base category in this instance). This is the identical interpretation of beta1 for REGA in the the first encoding approach (`final-dummy-variable-v1)`

The value of beta1 of -106.49 indicates that the deliveries made to region B are 106 minutes SHORTER (because of negative value) than it takes to region A (the base category in this instance). This value is not available in the first encoding approach.

However, by comparing the values of beta1 and beta2 here, we can deduce the value of beta2 from the first encoding approach (`final-dummy-variable-v1`):

We can express the meaning of beta1 in this model as:

a) REGA - REGB = 106

We can express the meaning of beta2 in this model as:

b) REGA - REGC = 107

We can rewrite a) as:
REGA = 106 + REGB
Substitute this into b)
106 + REGB - REGC = 107
REGB - REGC = 1
This simply means that it takes about 1 minute longer to deliver to Region B compared to Region C. This is the exact interpretation of beta2 for REGB in the first encoding approach (`final-dummy-variable-v1`):

Also, the predicted delivery times for 3 different regions: A, B, C for a fixed number of parcels (25) and truck age (10 years old) is the same as in the first encoding approach.

4 Multicollinearity and its impact

File to use: `Cars-pt1`

Construct regression models for the 3 separate linear regression equations shown in `Initial`.
You can create a copy for this purpose, the final results are shown in `Final1-Complete Regression` and `Final2-Partial Regression`.

Notice that when the two independent variables Displacement and Cylinders are included in the regression model, their respective p-values are higher than the significance level α (by default 0.05 for confidence level of 95% for normal regression model tables), which indicate that they are not relevant / important in the model.
However, when these 2 variables are used independently in a simple linear regression model with the dependent variable of MPG, they both have very low p-values respectively, indicating that they are very relevant / important for the model.

We perform a correlation check on these two variables using the Correlation functionality from the Data Analysis ToolPack and determine that there is a high level of correlation between them

This signals the problem of multicollinearity in a regression model. This occurs when independent variables in a regression model have high degree of correlation with each other (such as Displacement and Cylinders in this example).
This problem complicates interpretation of regression coefficients. In the original interpretation, the coefficient represents the change in the dependent variable for each 1 unit change in an independent variable when you hold all of the other independent variables constant. However, strong correlation causes multiple independent variables to change simultaneously

The simplest approach to fix this is to just drop one of the correlated independent variables when creating the regression model.