

Transformation

In the previous exercise, you transformed the response variable, ran a model, and made predictions. But you're not done yet! In order to interpret and visualize your predictions, you'll need to do a back transformation.



+0 XP

The `prediction_data`, which you created in the previous exercise, is

Brilliant back transformation! Notice that your back-transformed predictions nicely follow the trend line and allow you to make more accurate predictions.

Please rate this exercise:



PRESS ENTER TO

Continue

Submission

What is the correct way to add a layer of prediction points to the plot?

Was this feedback helpful?

✓ Yes

✗ No

Become a power user!

SUBMIT ANSWER: CTRL + SHIFT + ENTER

[See all keyboard shortcuts](#)

script.py

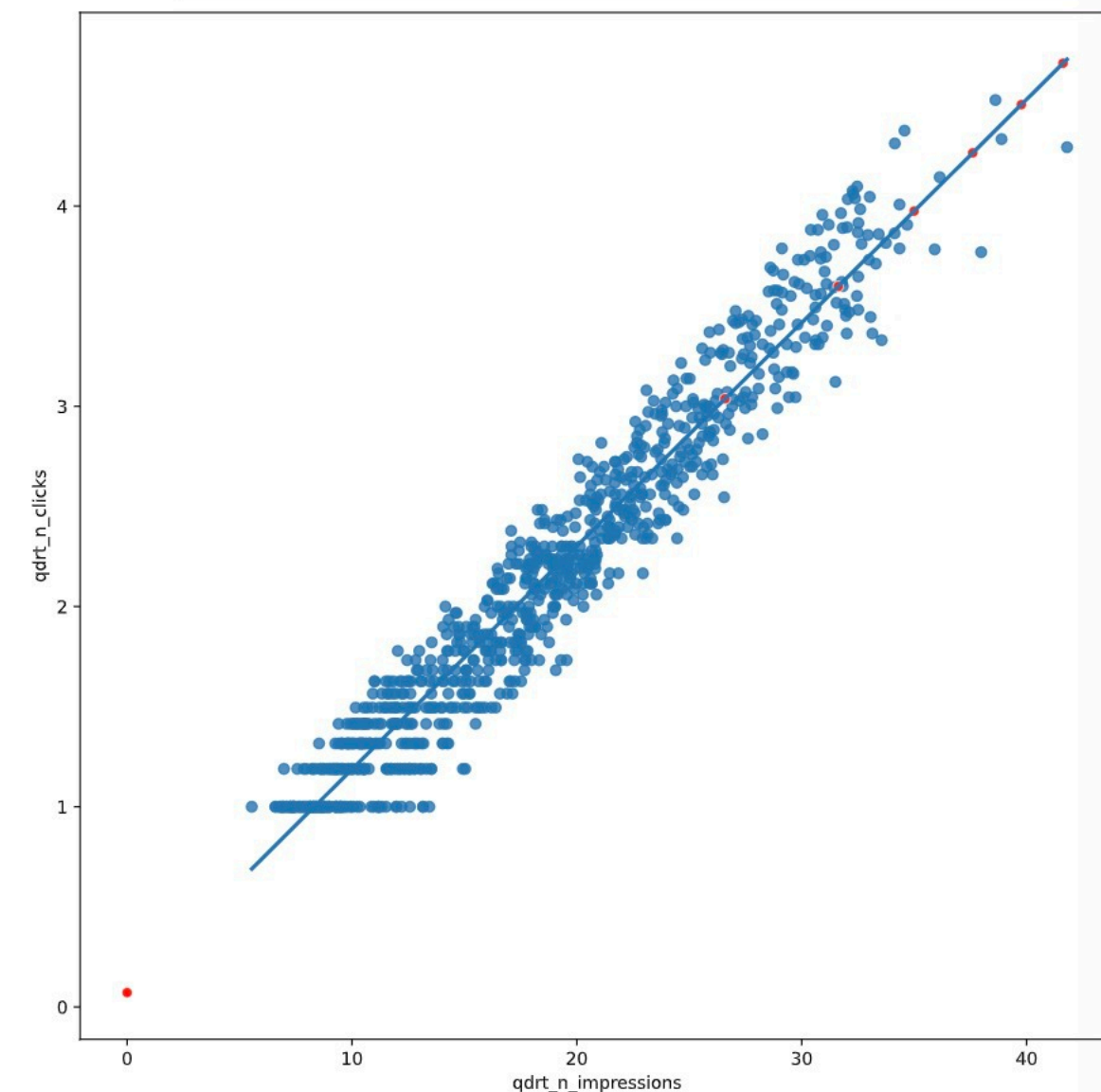
```
1 # Back transform qdrt_n_clicks
2 prediction_data["n_clicks"] =
  prediction_data
  ["qdrt_n_clicks"] ** 4
3
4 # Plot the transformed
  variables
5 fig = plt.figure()
6 sns.regplot
  (x="qdrt_n_impressions",
   y="qdrt_n_clicks",
   data=ad_conversion, ci=None)
7
8 # Add a layer of your
  prediction points
9 sns.scatterplot
  (x='qdrt_n_impressions',
   y='qdrt_n_clicks',
   data=prediction_data,
   color='red')
10 plt.show()
```



Run Code

Submit Answer

Plots



← Previous Plot

5/5

→ Next Plot

IPython Shell

Slides

Notes

<script.py> output:

	qdrt_n_impressions	n_impressions	qdrt_n_clicks	n_clicks
0	0.000	0.000e+00	0.072	2.650e-05
1	26.591	5.000e+05	3.038	8.514e+01
2	31.623	1.000e+06	3.599	1.677e+02
3	34.996	1.500e+06	3.975	2.497e+02
4	37.606	2.000e+06	4.266	3.312e+02