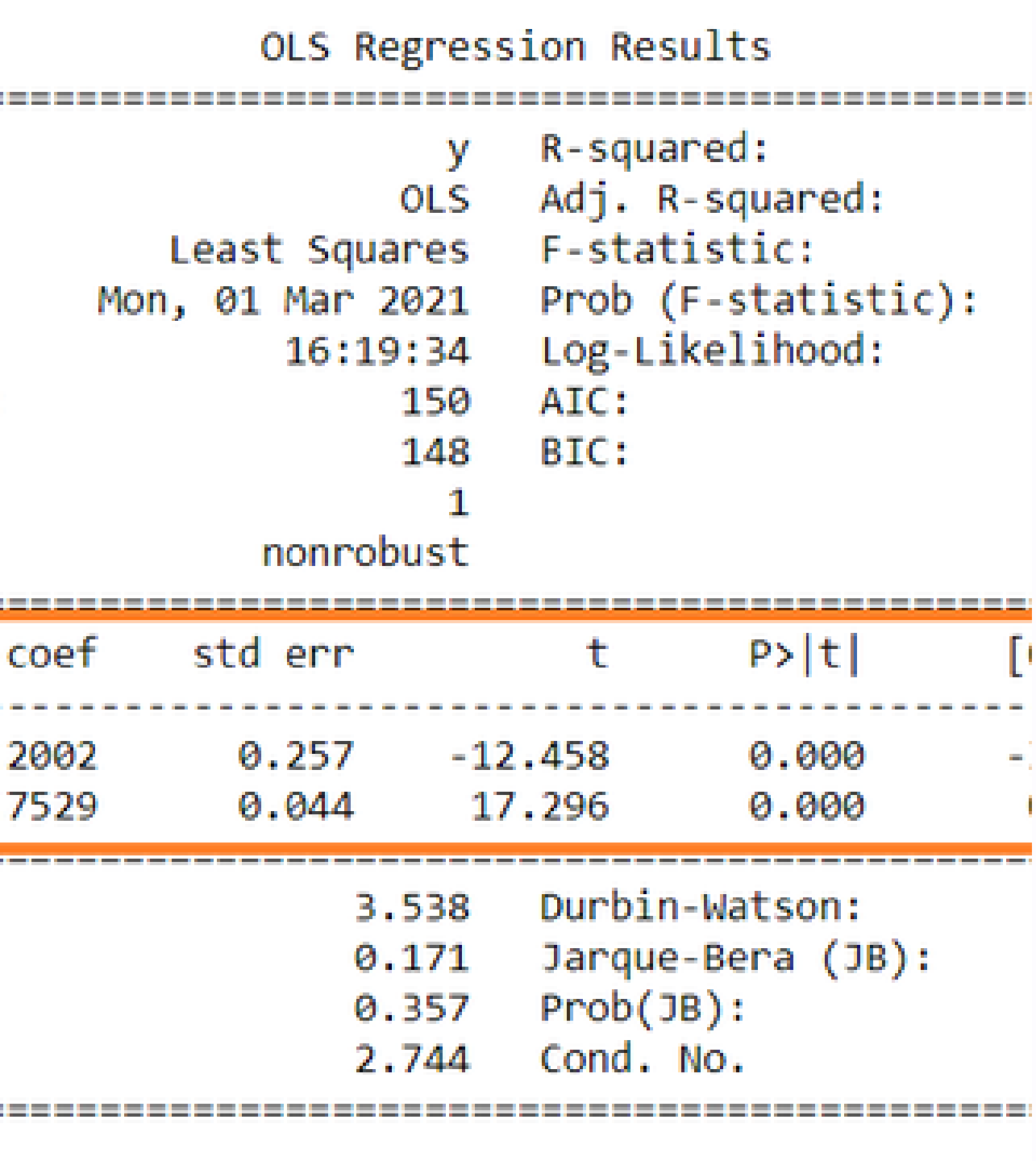


OLS Regression Results

	y		R-squared:
	OLS		Adj. R-squared:
	Least Squares		F-statistic:
	Mon, 01 Mar 2021		Prob (F-statistic):
	16:19:34		Log-Likelihood:
	150		AIC:
	148		BIC:
	1		
	nonrobust		

coef	std err	t	P>|t|	[
2002	0.257	-12.458	0.000	-
7529	0.044	17.296	0.000	

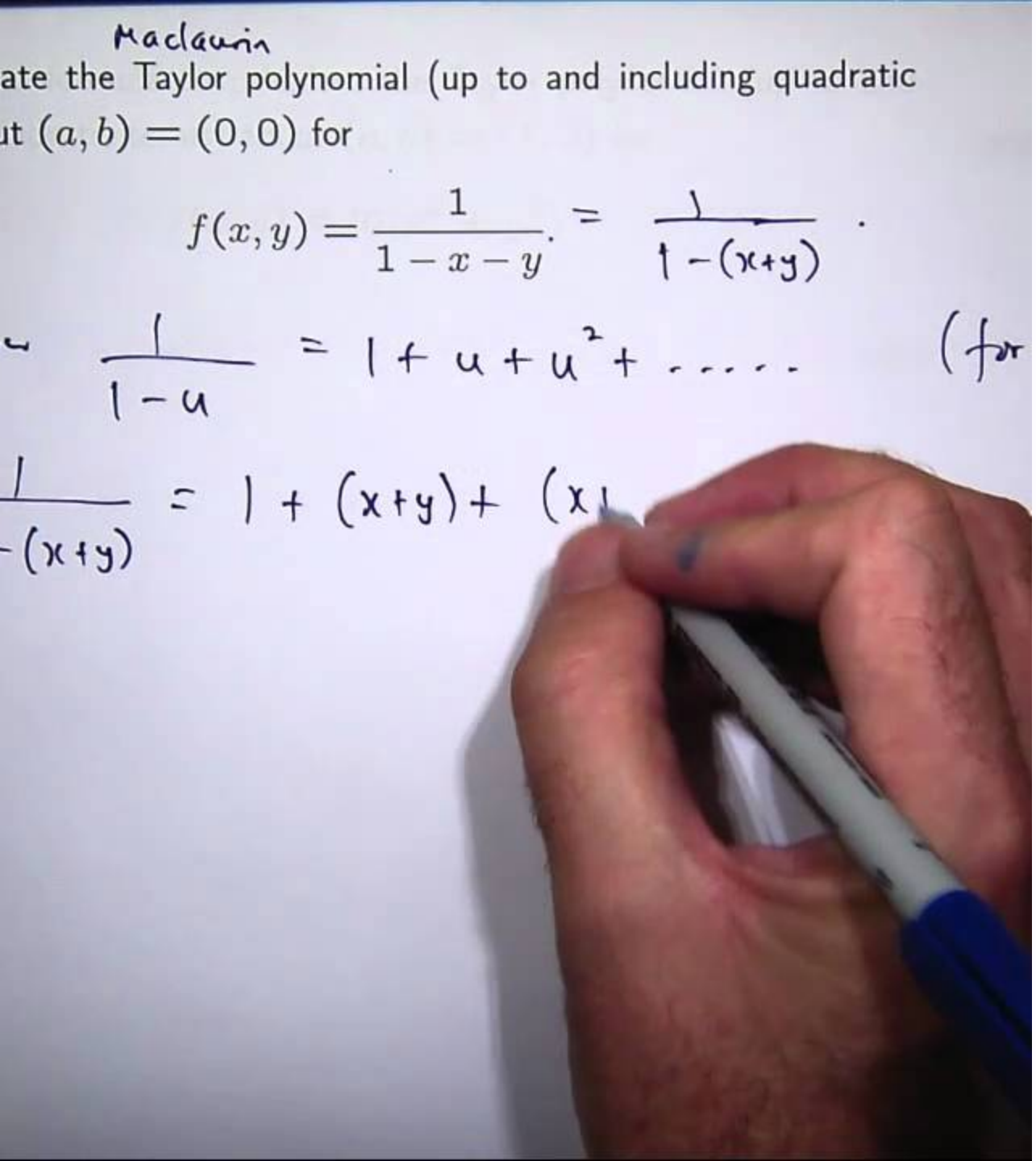
	3.538	Durbin-Watson:
	0.171	Jarque-Bera (JB):
	0.357	Prob(JB):
	2.744	Cond. No.

Regression

- **Linear regression** is a method used to model the relationship between a dependent variable and one or more independent variables.
- It assumes that there is a linear relationship between the variables

Simple Linear Regression

- Simple linear regression involves modeling the relationship between a dependent variable `y` and a single independent variable `x`.
- The line of best fit can be represented by the equation `y = b0 + b1*x`, where `b0` is the y-intercept and `b1` is the slope of the line.



Linear Regression

- Multiple linear regression involves modeling the relationship between a dependent variable `y` and multiple independent variables `x1, x2, ..., xn`.
- The line of best fit can be represented by the

in Python

- We can use the `LinearRegression` class from the `scikit-learn` library to perform linear regression in Python.
- Here is an example of how to fit a simple linear regression model:

```
from sklearn.linear_model import LinearRegression

# define the data
X = [[0], [1], [2]]
y = [0, 1, 2]

# create and fit the model
model = LinearRegression()
model.fit(X, y)
```

- We can use visualization libraries such as matplotlib or seaborn to create scatter plots and visualize the line of best fit.
- Here is an example of how to create a scatter plot with a fitted line:

```
import matplotlib.pyplot as plt

# define the data
X = [0, 1, 2]
y = [0, 1, 2]

# plot the data
plt.scatter(X, y)

# plot the fitted line
plt.plot(X, y)
```