BMY

Gerson Jimenez

03/11/2023

```
import os
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import statsmodels.formula.api as smf
import seaborn as sns
import statsmodels.api as sm
```

Q1

Data Preparation

```

mydata = pd.read_excel("/Users/gerson/Documents/School Terms/Spring 2023/ECO 4051 Financial Ecometrics/

data = mydata[(mydata.Ticker == "BMY")]
data = data.rename(columns = {'Revenue - Total': 'revtq', 'Fiscal Quarter': 'quarter'})
data = data[["Date", "quarter", "revtq"]]
data = pd.DataFrame(data)
data.head()
```

```
##            Date  quarter   revtq
## 660 1990-03-31        1  2458.0
## 661 1990-06-30        2  2484.0
## 662 1990-09-30        3  2622.0
## 663 1990-12-31        4  2736.0
## 664 1991-03-31        1  2735.0
```

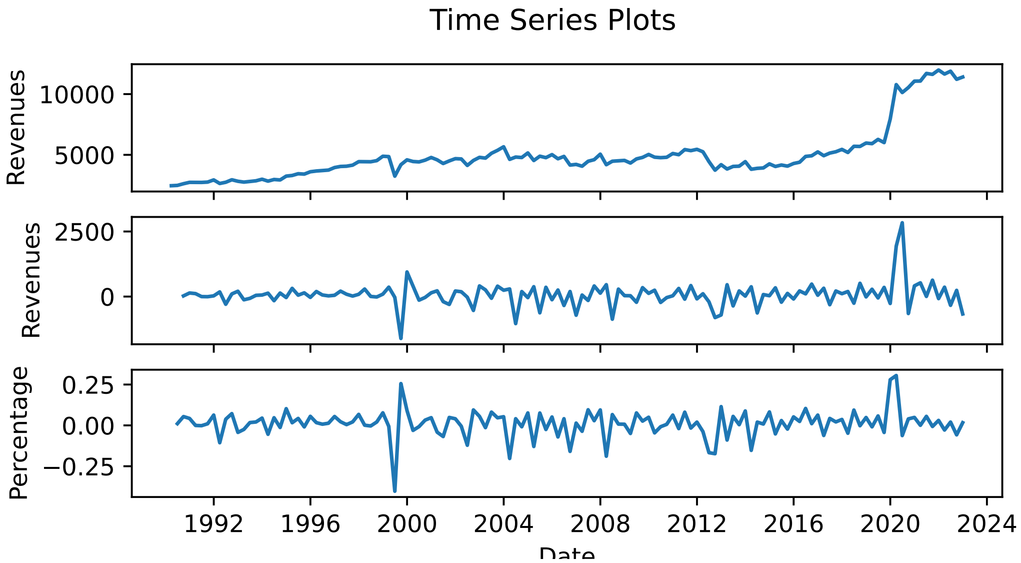
Q2

Plotting

```

fig, axes = plt.subplots(3,1, sharex= True)
fig.suptitle("Time Series Plots")

data['revtq1'] = data['revtq'].shift(1).diff().dropna()
data['revtq2'] = np.log(data['revtq']).diff().dropna()

axes[0].set_ylabel("Revenues")
sns.lineplot(ax =axes[0], x = "Date", y = "revtq", data = data)
sns.lineplot(ax= axes[1], x='Date', y='revtq1', data = data)
```

```
axes[1].set_ylabel("Revenues")
sns.lineplot(ax = axes[2], x= "Date", y = "revtq2", data = data)
axes[2].set_ylabel("Percentage")

# Top Graph is Quarterly Revenues
# Middle Graph is First Difference of the Quarterly Revenues
# Bottom Graph is the Log Difference of the Quarterly Revenues


plt.show()
```

Time Series Plots



Top Graph: Taking a look at the top graph which is the raw data of quarter revenue throughout the years, we are not able to clearly tell if there is no seasonality or trends, we do see some spikes and drops through a some years but seem to small to initiate any conclusion

Middle Graph: This graph represents the difference in quarters with one quarter of lag. We have a more clearer picture of volatility and the persistence of the data. The graph shows low persistence as it rarely deviates from mean 0. one interesting thing from this graph is that at the end of 2021 we see the revenues are decreasing.

Bottom Graph: The bottom graph is the log difference of revenues with no lag, We still see the low persistence but now the we see negative difference being more visible compared to the middle graph which seems to be favoring the positive differences in revenues. It is still quite unclear to see evidence of seasonality, it can be argued but we won't be seeing it until we create a regression model.

Q3

Trend-Stationary Model

```
# Creating new variables

trend1 = pd.Series(np.arange(1,len(data)+ 1))
trend = list(trend1)
trend2 = trend1 ** 2
```

```
trendsq = list(trend2)
trend3 = trend1 ** 3
trendcb = list(trend3)

data['trend'] = trend
data['trendsq'] = trendsq
data['trendcb'] = trendcb
```

```
# regression model on revenue on trend
model1 = smf.ols('revtq ~ trend', data = data).fit()
model1.summary()
```

```
## <class 'statsmodels.iolib.summary.Summary'>
## """
##                             OLS Regression Results                            
## ==============================================================================
## Dep. Variable:                  revtq   R-squared:                       0.512
## Model:                            OLS   Adj. R-squared:                  0.508
## Method:                 Least Squares   F-statistic:                     136.5
## Date:                Mon, 13 Mar 2023   Prob (F-statistic):           5.34e-22
## Time:                        12:04:27   Log-Likelihood:                -1155.1
## No. Observations:                 132   AIC:                             2314.
## Df Residuals:                     130   BIC:                             2320.
## Df Model:                           1                                         
## Covariance Type:            nonrobust                                         
## ==============================================================================
##                  coef    std err          t      P>|t|      [0.025      0.975]
## ------------------------------------------------------------------------------
## Intercept   2225.1837    269.559      8.255      0.000    1691.894    2758.473
## trend         41.0838      3.517     11.681      0.000      34.126      48.042
## ==============================================================================
## Omnibus:                       26.736   Durbin-Watson:                   0.087
## Prob(Omnibus):                  0.000   Jarque-Bera (JB):               37.492
## Skew:                           1.068   Prob(JB):                     7.22e-09
## Kurtosis:                       4.502   Cond. No.                         154.
## ==============================================================================
## 
## Notes:
## [1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
## """
```

Above is the regression results for revenue and the linear trend variable which we created.

One of major flaw to this linear model is the R-squared, after adjustment this model only represents 50% of the data, which is frankly too low to be useful in forecasting, but it can still be useful to figure out if there's a positive or negative trend

We do see the intercept for the trend starts at 41.08, which is roughly $41 million dollars per quarter. which means that as 1 unit of trend increases the revenues will increase by 41.08 million. This tells us we have a positive trend.

```
# regression model on revenue on trend + trendsq
model2 = smf.ols('revtq ~ trend + trendsq', data = data).fit()
model2.summary()
```

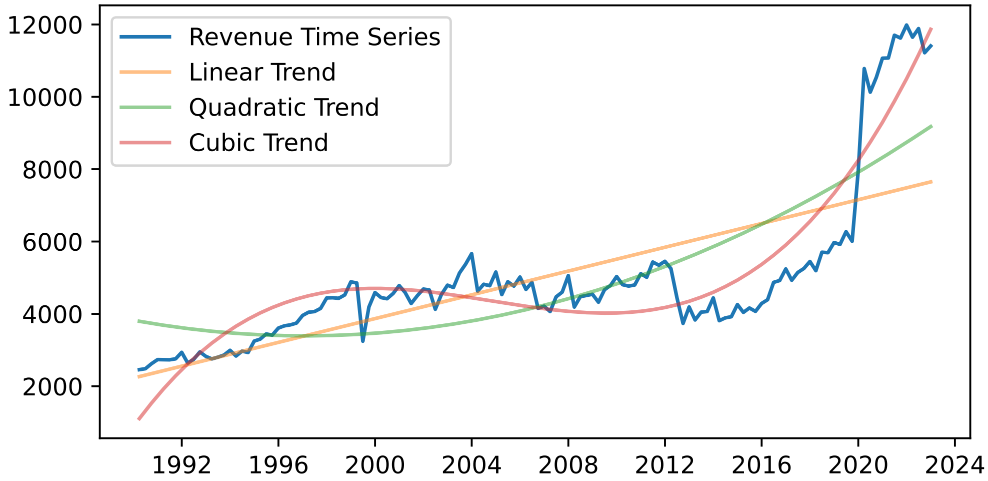
```
## <class 'statsmodels.iolib.summary.Summary'>
## """
##                             OLS Regression Results                            
## ==============================================================================
## Dep. Variable:                  revtq   R-squared:                       0.614
## Model:                            OLS   Adj. R-squared:                  0.608
## Method:                 Least Squares   F-statistic:                     102.7
## Date:                Mon, 13 Mar 2023   Prob (F-statistic):           2.05e-27
## Time:                        12:04:28   Log-Likelihood:                -1139.6
## No. Observations:                 132   AIC:                             2285.
## Df Residuals:                     129   BIC:                             2294.
## Df Model:                           2                                         
## Covariance Type:            nonrobust                                         
## ==============================================================================
##                  coef    std err          t      P>|t|      [0.025      0.975]
## ------------------------------------------------------------------------------
## Intercept   3825.0056    364.346     10.498      0.000    3104.139    4545.872
## trend        -30.5500     12.647     -2.416      0.017     -55.572      -5.528
## trendsq        0.5386      0.092      5.847      0.000       0.356       0.721
## ==============================================================================
## Omnibus:                        3.046   Durbin-Watson:                   0.108
## Prob(Omnibus):                  0.218   Jarque-Bera (JB):                2.661
## Skew:                           0.250   Prob(JB):                        0.264
## Kurtosis:                       2.517   Cond. No.                     2.40e+04
## ==============================================================================
## 
## Notes:
## [1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
## [2] The condition number is large, 2.4e+04. This might indicate that there are
## strong multicollinearity or other numerical problems.
## """
```

```
# regression model on revenue on trend + trendsq + trendcb
model3 = smf.ols('revtq ~ trend + trendsq + trendcb', data = data).fit()
model3.summary()
```

```
## <class 'statsmodels.iolib.summary.Summary'>
## """
##                             OLS Regression Results                            
## ==============================================================================
## Dep. Variable:                  revtq   R-squared:                       0.851
## Model:                            OLS   Adj. R-squared:                  0.847
## Method:                 Least Squares   F-statistic:                     243.2
## Date:                Mon, 13 Mar 2023   Prob (F-statistic):           1.14e-52
## Time:                        12:04:29   Log-Likelihood:                -1076.9
## No. Observations:                 132   AIC:                             2162.
## Df Residuals:                     128   BIC:                             2173.
## Df Model:                           3                                         
## Covariance Type:            nonrobust                                         
## ==============================================================================
##                  coef    std err          t      P>|t|      [0.025      0.975]
## ------------------------------------------------------------------------------
## Intercept    879.9814    307.504      2.862      0.005     271.533    1488.430
## trend        230.2580     19.948     11.543      0.000     190.788     269.728
## trendsq       -4.3454      0.348    -12.494      0.000      -5.034      -3.657
## trendcb        0.0245      0.002     14.238      0.000       0.021       0.028
## ==============================================================================
## Omnibus:                        2.749   Durbin-Watson:                   0.275
## Prob(Omnibus):                  0.253   Jarque-Bera (JB):                2.228
## Skew:                           0.187   Prob(JB):                        0.328
## Kurtosis:                       2.486   Cond. No.                     3.63e+06
## ==============================================================================
##
## Notes:
## [1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
## [2] The condition number is large, 3.63e+06. This might indicate that there are
## strong multicollinearity or other numerical problems.
## """
```

```
fig, ax = plt.subplots()
ax.plot(data['Date'], data['revtq'])
ax.plot(data['Date'], model1.fittedvalues, alpha = 0.5)
ax.plot(data['Date'], model2.fittedvalues, alpha = 0.5)
ax.plot(data['Date'], model3.fittedvalues, alpha =0.5)
ax.legend(['Revenue Time Series', 'Linear Trend', 'Quadratic Trend', 'Cubic Trend'])
plt.show()
```

Plotting of Fitted Revenue Values



Out of all the three different models, I prefer using model 3, although this is a non linear model, the model is able to explain 84.7% of the data. Looking at the plot we see that this model has the closes residuals to the actual data. This Model also has the lowest AIC and BIC at 2162 and 2173. The best fit models are those who AIC and BIC results are the lowest.

Differences between the models

Linear : The linear model shown in orange is not efficient, as it assumes that the rate of change is constant in this case at 41 million dollars.

Quadratic : The quadratic model in green has some curvature with can be used to show acceleration or decelerating in revenue growth. This model has a very high intercept starting above the first revenue figures.

Cubic : the cubic is able to expand among the quadratic model which a more complex curve allowing it to explain the high variability of the revenues.

Q4

Seasonal Quarters

```
# Creating seasonal dummy variables

seasonal = pd.get_dummies(data['quarter'], prefix = "Q")
data["Q1"] = seasonal["Q_1"]
data["Q2"] = seasonal["Q_2"]
data["Q3"] = seasonal["Q_3"]
data["Q4"] = seasonal["Q_4"]

model4 = smf.ols('revtq ~ trend + trendsq + trendcb + Q2 + Q3 + Q4', data = data).fit()

model4.summary()
```

```
## <class 'statsmodels.iolib.summary.Summary'>
## """
##                             OLS Regression Results
## ==============================================================================
## Dep. Variable:                  revtq   R-squared:                       0.852
## Model:                            OLS   Adj. R-squared:                  0.845
## Method:                 Least Squares   F-statistic:                     120.0
## Date:                Mon, 13 Mar 2023   Prob (F-statistic):           2.00e-49
## Time:                        12:04:31   Log-Likelihood:                -1076.3
## No. Observations:                 132   AIC:                             2167.
## Df Residuals:                     125   BIC:                             2187.
## Df Model:                           6
## Covariance Type:            nonrobust
## ==============================================================================
##                  coef    std err          t      P>|t|      [0.025      0.975]
## ------------------------------------------------------------------------------
## Intercept    880.2623    332.343      2.649      0.009     222.515    1538.010
## trend        229.8146     20.113     11.426      0.000     190.008     269.621
## trendsq       -4.3375      0.351    -12.368      0.000      -5.032      -3.643
## trendcb        0.0244      0.002     14.098      0.000       0.021       0.028
## Q2           -23.3426    212.871     -0.110      0.913    -444.640     397.955
## Q3           -86.0526    212.962     -0.404      0.687    -507.532     335.426
## Q4           134.5718    213.115      0.631      0.529    -287.210     556.353
## ==============================================================================
## Omnibus:                        3.081   Durbin-Watson:                   0.247
## Prob(Omnibus):                  0.214   Jarque-Bera (JB):                2.457
## Skew:                           0.204   Prob(JB):                        0.293
## Kurtosis:                       2.471   Cond. No.                     4.35e+06
## ==============================================================================
##
## Notes:
## [1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
## [2] The condition number is large, 4.35e+06. This might indicate that there are
## strong multicollinearity or other numerical problems.
## """
```

Quarter 1 is not among the regressors as its the reference, including the reference will result in a dummy variable trap.

Looking at the coefficients of the quarter 2 tells us that revenues of this quarter are -23 million dollars below reference quarter 1. As we see this quarter is not one of its best. In quarter 3 we get -86 million dollars making this the worst quarter when it comes to revenues. Finally in Q4 we get 134 million dollars, this provides evidences to say that there is seasonality since quarter 4 conducts the highest revenue of all quarters. Meaning if the revenues are within quarter 4 it will see a 134 million dollar increase.

The coefficients of the trend variables has not seen much of a difference in the addition of the quarters dummy variables since the 3 trend models are tied to overall data. the quarters are just 4 periods of time within that data, meaning the addition of these dummy variables should not have overall significant impact on the observable data.

Q5

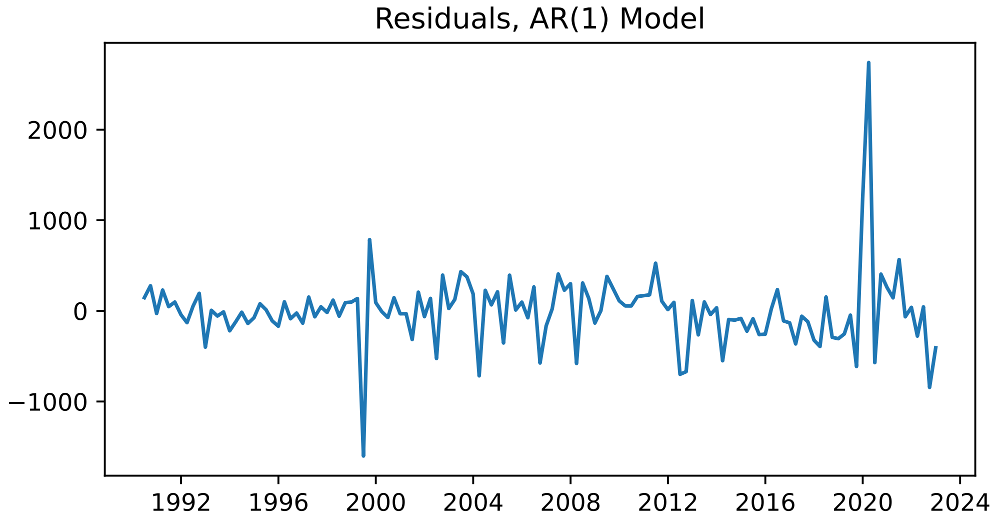
AutoCorrelation Regression

```
model5 = smf.ols('revtq ~ trend + trendsq + trendcb + Q2 + Q3 + Q4 + revtq.shift(1)', data=data).fit()

# Print model summary
model5.summary()
```

```
## <class 'statsmodels.iolib.summary.Summary'>
## """
##                             OLS Regression Results
## ==============================================================================
## Dep. Variable:                  revtq   R-squared:                       0.966
## Model:                            OLS   Adj. R-squared:                  0.964
## Method:                 Least Squares   F-statistic:                     501.3
## Date:                Mon, 13 Mar 2023   Prob (F-statistic):           3.32e-87
## Time:                        12:04:32   Log-Likelihood:                -971.44
## No. Observations:                 131   AIC:                             1959.
## Df Residuals:                     123   BIC:                             1982.
## Df Model:                           7
## Covariance Type:            nonrobust
## ==================================================================================
##                      coef    std err          t      P>|t|      [0.025      0.975]
## ----------------------------------------------------------------------------------
## Intercept        -10.9299    174.599     -0.063      0.950    -356.537     334.678
## trend             31.0523     14.465      2.147      0.034       2.419      59.686
## trendsq           -0.6185      0.259     -2.388      0.018      -1.131      -0.106
## trendcb            0.0036      0.001      2.669      0.009       0.001       0.006
## Q2               154.8774    103.199      1.501      0.136     -49.398     359.153
## Q3               111.3594    103.253      1.079      0.283     -93.024     315.743
## Q4               385.4095    103.507      3.724      0.000     180.525     590.294
## revtq.shift(1)     0.8683      0.043     20.185      0.000       0.783       0.953
## ==============================================================================
## Omnibus:                       90.779   Durbin-Watson:                   1.972
## Prob(Omnibus):                  0.000   Jarque-Bera (JB):             1644.994
## Skew:                           1.954   Prob(JB):                         0.00
## Kurtosis:                      19.915   Cond. No.                     4.70e+06
## ==============================================================================
##
```

```
## Notes:
## [1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
## [2] The condition number is large, 4.7e+06. This might indicate that there are
## strong multicollinearity or other numerical problems.
## """
```

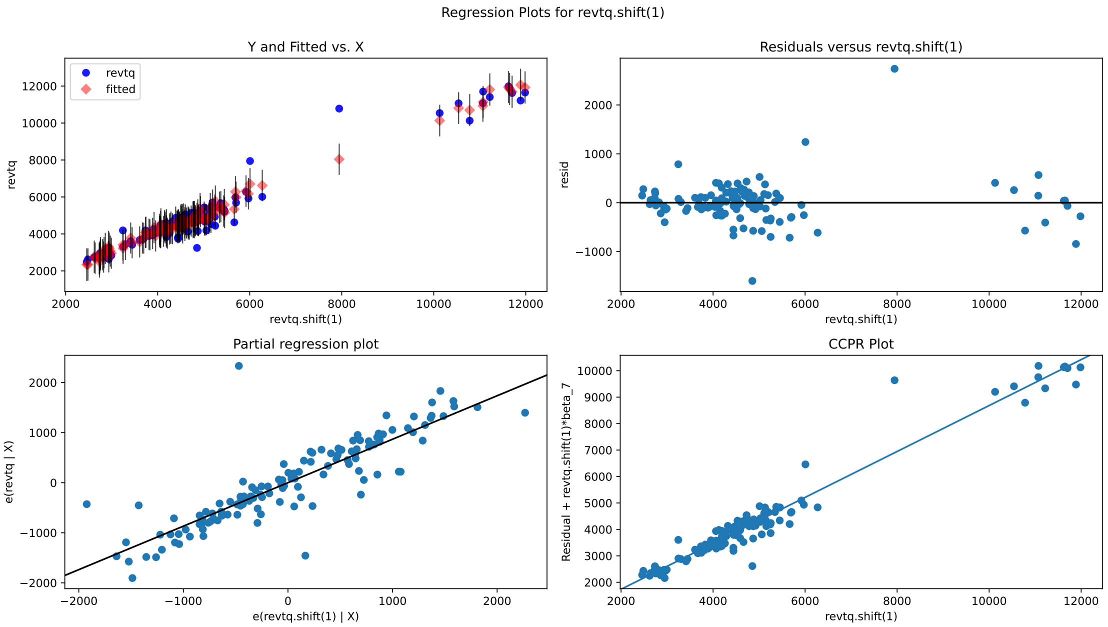
```
residuals = model5.resid
fig, ax = plt.subplots()

ax.plot(data['Date'][1:], residuals)
ax.set_title("Residuals, AR(1) Model")
plt.show()


# creating regression plots
```



```
fig = plt.figure(figsize=(14, 8))
fig = sm.graphics.plot_regress_exog(model5,'revtq.shift(1)',fig=fig)
```

```
## eval_env: 1
```

```
plt.show()
```



Taking a look at the "residuals versus revtq.shift(1)" graph, we see that many of the residuals values are near the mean 0 but we also see many outliers on these graphs, i do think there is better ways to minimize this but this will require more data and probably some external and internal figures not presented in the data.

We can evaluate the independence of the residuals using the Durbin-Watson. In the OLS summary we see our DW stat is 1.972 quite high almost near the no autocorrelation which is = 2. since our value is below 2 but yet so close to 2 which means that there's a slight positive autocorrelation.

Q6

Using Log(revtq) instead of revtq (log of the revenues)

```
model6 = smf.ols('np.log(revtq) ~ trend + trendsq + trendcb + Q2 + Q3 + Q4 + np.log(revtq).shift(1)', da

# Print model summary
model6.summary()
```

```
## <class 'statsmodels.iolib.summary.Summary'>
## """
##                             OLS Regression Results
## ==============================================================================
## Dep. Variable:          np.log(revtq)   R-squared:                       0.960
## Model:                            OLS   Adj. R-squared:                  0.957
## Method:                 Least Squares   F-statistic:                     418.0
## Date:                Mon, 13 Mar 2023   Prob (F-statistic):           1.55e-82
## Time:                        12:04:33   Log-Likelihood:                 160.81
## No. Observations:                 131   AIC:                            -305.6
## Df Residuals:                     123   BIC:                            -282.6
## Df Model:                           7
## Covariance Type:            nonrobust
## ==========================================================================================
##                              coef    std err          t      P>|t|      [0.025      0.975]
## ------------------------------------------------------------------------------------------
## Intercept                  1.4163      0.393      3.601      0.000       0.638       2.195
## trend                      0.0085      0.003      2.805      0.006       0.003       0.015
## trendsq                   -0.0001   5.09e-05     -2.917      0.004      -0.000   -4.77e-05
## trendcb                 8.007e-07   2.58e-07      3.105      0.002     2.9e-07    1.31e-06
## Q2                         0.0375      0.018      2.050      0.042       0.001       0.074
## Q3                         0.0350      0.018      1.910      0.058      -0.001       0.071
## Q4                         0.0849      0.018      4.628      0.000       0.049       0.121
## np.log(revtq).shift(1)     0.8093      0.052     15.579      0.000       0.707       0.912
## ==============================================================================
## Omnibus:                       42.254   Durbin-Watson:                   2.121
## Prob(Omnibus):                  0.000   Jarque-Bera (JB):              470.894
## Skew:                          -0.641   Prob(JB):                    5.58e-103
## Kurtosis:                      12.199   Cond. No.                     5.49e+07
## ==============================================================================
##
## Notes:
## [1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
## [2] The condition number is large, 5.49e+07. This might indicate that there are
## strong multicollinearity or other numerical problems.
## """
```

Model 5 and Model 6 are quite different we see that one big difference is the Durbin-Watson Stat which does from suggesting a positive autocorrellation to a negative one. We see very similar quarterly data but a difference in the intercepts. Model 5 had a negative -10.93 intercept, while Model 6 is now at a positive 1.4163

I prefer the log model (Model 6) since the standard errors are very low. but both the models do not have independent residuals since they show autocorrelation.

Q7

Calculating Forecast for revenue of the company in the following quarter

```
beta0 = model4.params['Intercept']
beta1 = model4.params['trend']
beta2 = model4.params['trendsq']
beta3 = model4.params['trendcb']
beta4 = model4.params['Q2']
beta5 = model4.params['Q3']
beta6 = model4.params['Q4']

# Extracting last values of variables
t = data['trend'].iloc[-1] + 1
tsq = t ** 2
tcb = t ** 3
q2 = 0
q3 = 1
q4 = 0

# Calculating forecast
revenue_forecast = beta0 + beta1*t + beta2*tsq + beta3*tcb + beta4*q2 + beta5*q3 + beta6*q4

print(revenue_forecast)
```

```
## 12135.040840909856
```

Above is the equation used to forecast Q1 2023 Revenue. My model estimates that the next quarter revenue will be 12.135 billion dollars, a bit higher then the analysis found in Yahoo finance. the low estimate = 11.3 billion, high estimate = 12.02 billion. my estimate is 0.95% greater than the estimated high. It is quite possible that i could be lacking some factors in my data that is not present to the general public.