A Regression-Based Assessment of Revenue Growth Percentages

By: *Pedram Bazargani*

Data Description: This data consists of 7 columns and 100 rows, listing the top 100 American companies by Revenue (USD), according to the 2023 Fortune 500 list published by Fortune Magazine. The list was scraped off Wikipedia and has a usability score of 10.00 on Kaggle. The data was provided in CSV format and was prepped and cleaned for via the Alteryx Designer Platform. Tools utilized include the Data Cleansing, Select, and Formula tools. The data was then outputed back into csv file format and loaded to this notebook on Google Colab.

Source: https://www.kaggle.com/datasets/claymaker/us-largest-companies

Motivations and Findings: During my undergrad courses in econometrics I learned a lot about building and interpreting regressions. To showcase my obtained technical accumen I searched Kaggle for some fun data I could work with and shortly found this data on 2023 Fortune 500 Companies published by Fortune.

The column I found most interesting was the data on Revenue Growth Percentages. Although I recognize that revenue growth is determined primarily by market conditions, investor sentiments, and numerous finacial metrics, I wanted to see to what extent do the number of employees a company have impact its revenue growth percentage. I further expanded my research to include the Industry and Location columns and also decided to run a few fun statistics regarding the data at the end of this notebook.

My regressions demonstrated that the number of employees have minimal impact on revenue growth percentage. Even though adding categorical variables 'Industry' and 'Headquarters (as State)' to the regression via one-hot encoding did result in a larger goodness of fit indicating possible correlation, ultimately these findings may possibly be inconclusive due to issues of multicollinearity.

```
#Importing Packages; Loading Data; Creating a DataFrame

import pandas as pd
import numpy as np
import statsmodels.api as sm
import matplotlib.pyplot as plt

data = pd.read_csv('/content/Fortune500CleanedData.csv')
df = pd.DataFrame(data)
print(df)
```

```
    Rank                     Name                     Industry  \
0      1                  Walmart                       Retail
1      2                   Amazon  Retail and Cloud Computing
2      3              Exxon Mobil           Petroleum industry
3      4                    Apple          Electronics industry
4      5        UnitedHealth Group                   Healthcare
..   ...                      ...                          ...
95    96                 Best Buy                       Retail
96    97       BristolMyers Squibb       Pharmaceutical industry
97    98          United Airlines                      Airline
98    99  Thermo Fisher Scientific      Laboratory instruments
99   100                 Qualcomm                   Technology

    Revenue (USD millions)  Revenue growth  Employees            Headquarters
0                   611289           0.067    2100000    Bentonville, Arkansas
1                   513983           0.094    1540000       Seattle, Washington
2                   413680           0.448      62000            Spring, Texas
3                   394328           0.078     164000     Cupertino, California
4                   324162           0.127     400000     Minnetonka, Minnesota
..                     ...             ...        ...                      ...
95                   46298           0.106      71100      Richfield, Minnesota
96                   46159           0.005      34300   New York City, New York
97                   44955           0.825      92795         Chicago, Illinois
98                   44915           0.145     130000   Waltham, Massachusetts
99                   44200           0.317      51000     San Diego, California

[100 rows x 7 columns]
```

```
#Calculating the correlation between 'Employees' and 'Revenue Growth Percentage (RGP)'
correlation_coefficient = df['Employees'].corr(df['Revenue growth'])

print(f"Correlation coefficient between number of employees and revenue growth percentages (RGP): {correlation_coefficient:.3f}")

# Interpreting the correlation coefficient
if 0.7 <= correlation_coefficient <= 1:
```

```
        interpretation = "strong positive correlation"
    elif 0.3 < correlation_coefficient < 0.7:
        interpretation = "moderate positive correlation"
    elif 0 < correlation_coefficient <= 0.3:
        interpretation = "weak positive correlation"
    elif -0.3 < correlation_coefficient <= 0:
        interpretation = "weak negative correlation"
    elif -0.7 < correlation_coefficient <= -0.3:
        interpretation = "moderate negative correlation"
    else:
        interpretation = "strong negative correlation"

    print(f"There's a {interpretation} between the number of employees and revenue growth percentage.")
    print("\n")
    print("This suggests that there may be a weak inverse relationship in that companies with more employees may experience a lower RGP and vice
```

```
Correlation coefficient between number of employees and revenue growth percentages (RGP): -0.230
There's a weak negative correlation between the number of employees and revenue growth percentage.


This suggests that there may be a weak inverse relationship in that companies with more employees may experience a lower RGP and vice ve
```

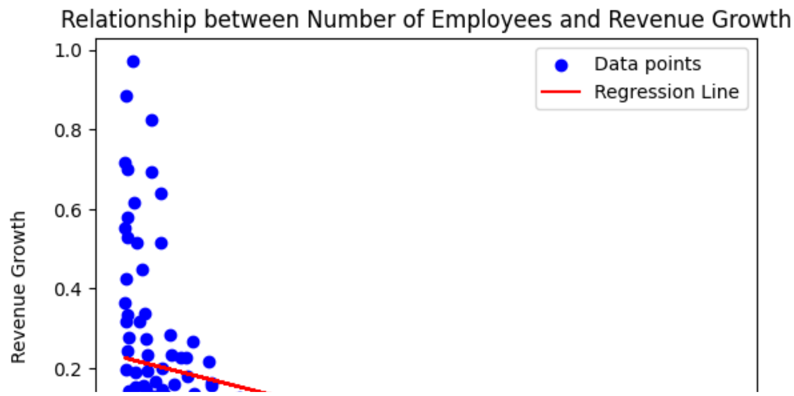
```
#Performing simple linear regression (OLS) analysis to understand the impact of the number of employees on RGP
X = df['Employees']
y = df['Revenue growth']


X = sm.add_constant(X)
model = sm.OLS(y, X).fit()
predictions = model.predict(X)
print(model.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:         Revenue growth   R-squared:                       0.053
Model:                            OLS   Adj. R-squared:                  0.043
Method:                 Least Squares   F-statistic:                     5.478
Date:                Sun, 27 Aug 2023   Prob (F-statistic):             0.0213
Time:                        23:15:28   Log-Likelihood:                 15.210
No. Observations:                 100   AIC:                            -26.42
Df Residuals:                      98   BIC:                            -21.21
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          0.2263      0.025      9.155      0.000       0.177       0.275
Employees  -1.817e-07   7.76e-08     -2.340      0.021   -3.36e-07   -2.76e-08
==============================================================================
Omnibus:                       36.930   Durbin-Watson:                   2.083
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               63.683
Skew:                           1.622   Prob(JB):                     1.48e-14
Kurtosis:                       5.181   Cond. No.                     3.75e+05
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 3.75e+05. This might indicate that there are
strong multicollinearity or other numerical problems.
```

```
#Plotting the scatter plot with regression line
plt.scatter(df['Employees'], df['Revenue growth'], color='blue', label='Data points')
plt.plot(df['Employees'], predictions, color='red', label='Regression Line')
plt.xlabel('Number of Employees (millions)')
plt.ylabel('Revenue Growth')
plt.title('Relationship between Number of Employees and Revenue Growth')
plt.legend()
plt.show()
```




```
#Performing Polynomial Regression to Capture non-linear relationship
df['Employees_squared'] = df['Employees'] ** 2

x_poly = df[['Employees', 'Employees_squared']]
x_poly = sm.add_constant(x_poly)

model2 = sm.OLS(y, x_poly).fit()
print("\nPolynomial Regression:\n", model2.summary())
```

```
Polynomial Regression:
                             OLS Regression Results
==============================================================================
Dep. Variable:         Revenue growth   R-squared:                       0.128
Model:                            OLS   Adj. R-squared:                  0.110
Method:                 Least Squares   F-statistic:                     7.144
Date:                Sun, 27 Aug 2023   Prob (F-statistic):            0.00128
Time:                        18:21:20   Log-Likelihood:                 19.361
No. Observations:                 100   AIC:                            -32.72
Df Residuals:                      97   BIC:                            -24.91
Df Model:                           2
Covariance Type:            nonrobust
=====================================================================================
                        coef    std err          t      P>|t|      [0.025      0.975]
-------------------------------------------------------------------------------------
const                 0.2829      0.031      9.179      0.000       0.222       0.344
Employees         -7.053e-07   1.96e-07     -3.606      0.000   -1.09e-06   -3.17e-07
Employees_squared  3.095e-13   1.07e-13      2.898      0.005    9.75e-14    5.21e-13
==============================================================================
Omnibus:                       30.989   Durbin-Watson:                   2.247
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               48.228
Skew:                           1.414   Prob(JB):                     3.37e-11
Kurtosis:                       4.890   Cond. No.                     7.70e+11
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 7.7e+11. This might indicate that there are
strong multicollinearity or other numerical problems.
```

```
#Log-Transformed Regression
df['log_Employees'] = np.log(df['Employees'])

X_log = df['log_Employees']
X_log = sm.add_constant(X_log)

model3 = sm.OLS(y, X_log).fit()
print("\nLog-transformed Regression:\n", model3.summary())
```

```
Log-transformed Regression:
                             OLS Regression Results
==============================================================================
Dep. Variable:         Revenue growth   R-squared:                       0.240
Model:                            OLS   Adj. R-squared:                  0.232
Method:                 Least Squares   F-statistic:                     30.86
Date:                Sun, 27 Aug 2023   Prob (F-statistic):           2.38e-07
Time:                        18:23:21   Log-Likelihood:                 26.180
No. Observations:                 100   AIC:                            -48.36
Df Residuals:                      98   BIC:                            -43.15
Df Model:                           1
Covariance Type:            nonrobust
```

```
=================================================================================
                    coef    std err          t      P>|t|      [0.025      0.975]
---------------------------------------------------------------------------------
const             1.1289      0.169      6.679      0.000       0.793       1.464
log_Employees    -0.0824      0.015     -5.555      0.000      -0.112      -0.053
==============================================================================
Omnibus:                       34.323   Durbin-Watson:                   2.258
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               59.977
Skew:                           1.464   Prob(JB):                     9.46e-14
Kurtosis:                       5.412   Cond. No.                         103.
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

```
#Adding Categorical variables 'Industry' and 'Headquarters (as State)' to Regression for a Richer model
df['State'] = df['Headquarters'].str.split(', ').str[-1]

#One-hot encoding for 'Industry' and 'State'
industry_dummies = pd.get_dummies(df['Industry'], drop_first=True, prefix='Industry')
state_dummies = pd.get_dummies(df['State'], drop_first=True, prefix='State')

df_encoded = pd.concat([df, industry_dummies, state_dummies], axis=1)
X = df_encoded[['Employees'] + list(industry_dummies.columns) + list(state_dummies.columns)]
X = sm.add_constant(X)
y = df['Revenue growth']

model = sm.OLS(y, X).fit()
print(model.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:         Revenue growth   R-squared:                       0.863
Model:                            OLS   Adj. R-squared:                  0.644
Method:                 Least Squares   F-statistic:                     3.932
Date:                Sun, 27 Aug 2023   Prob (F-statistic):           9.59e-06
Time:                        22:39:13   Log-Likelihood:                 111.96
No. Observations:                 100   AIC:                            -99.92
Df Residuals:                      38   BIC:                             61.60
Df Model:                          61
Covariance Type:            nonrobust
===============================================================================================================
                                                coef    std err          t      P>|t|      [0.025      0.975]
---------------------------------------------------------------------------------------------------------------
const                                        -0.1242      0.189     -0.658      0.515      -0.506       0.258
Employees                                  1.376e-07   1.21e-07      1.138      0.262   -1.07e-07    3.82e-07
Industry_Aerospace and defense               -0.1811      0.140     -1.293      0.204      -0.465       0.102
Industry_Agriculture cooperative              0.0789      0.155      0.510      0.613      -0.234       0.392
Industry_Agriculture manufacturing           -0.0460      0.140     -0.329      0.744      -0.329       0.237
Industry_Airline                              0.5251      0.084      6.230      0.000       0.354       0.696
Industry_Apparel                              0.0811      0.110      0.735      0.467      -0.142       0.305
Industry_Automotive                           0.1481      0.114      1.304      0.200      -0.082       0.378
Industry_Automotive and Energy                0.3030      0.139      2.175      0.036       0.021       0.585
Industry_Automotive industry                  0.0722      0.113      0.637      0.528      -0.157       0.302
Industry_Beverage                            -0.2248      0.138     -1.633      0.111      -0.503       0.054
Industry_Chemical industry                   -0.0332      0.119     -0.279      0.781      -0.273       0.207
Industry_Conglomerate                        -0.1111      0.087     -1.277      0.209      -0.287       0.065
Industry_Consumer products Manufacturing     -0.0817      0.142     -0.575      0.569      -0.369       0.206
Industry_Electronics industry                -0.1411      0.143     -0.989      0.329      -0.430       0.148
Industry_Financial                           -0.0218      0.098     -0.222      0.826      -0.220       0.177
Industry_Financial services                  -0.0918      0.137     -0.670      0.507      -0.369       0.186
Industry_Financials                          -0.0931      0.053     -1.764      0.086      -0.200       0.014
Industry_Food Processing                      0.2793      0.144      1.938      0.060      -0.012       0.571
Industry_Food Service                         0.1349      0.139      0.970      0.338      -0.147       0.416
Industry_Food industry                       -0.0890      0.100     -0.894      0.377      -0.290       0.113
Industry_Health                              -0.1521      0.139     -1.093      0.281      -0.434       0.130
Industry_Health Insurance                    -0.2702      0.208     -1.302      0.201      -0.690       0.150
Industry_Healthcare                          -0.0708      0.102     -0.696      0.490      -0.277       0.135
Industry_Infotech                             0.7260      0.184      3.942      0.000       0.353       1.099
Industry_Insurance                           -0.1432      0.066     -2.172      0.036      -0.277      -0.010
Industry_Laboratory instruments              -0.0115      0.150     -0.077      0.939      -0.315       0.292
Industry_Logistics                           -0.3482      0.197     -1.770      0.085      -0.747       0.050
Industry_Machinery                           -0.0777      0.140     -0.555      0.582      -0.361       0.205
Industry_Media                                0.0035      0.143      0.025      0.980      -0.286       0.293
Industry_Petroleum industry                   0.2898      0.062      4.639      0.000       0.163       0.416
Industry_Petroleum industry and Logistics     0.6412      0.184      3.487      0.001       0.269       1.013
Industry_Pharmaceutical industry             -0.1865      0.067     -2.800      0.008      -0.321      -0.052
Industry_Retail                              -0.1404      0.073     -1.915      0.063      -0.289       0.008
Industry_Retail and Cloud Computing          -0.3732      0.241     -1.548      0.130      -0.861       0.115
Industry_Technology                          -0.1043      0.067     -1.558      0.128      -0.240       0.031
Industry_Telecom Hardware Manufacturing      -0.1730      0.142     -1.218      0.231      -0.460       0.114
```

```
Industry_Telecommunications          -0.2665      0.110     -2.419      0.020      -0.490      -0.043
Industry_Transportation              -0.0993      0.136     -0.731      0.469      -0.375       0.176
State_California                      0.3207      0.204      1.572      0.124      -0.092       0.734
State_Connecticut                     0.4217      0.261      1.618      0.114      -0.106       0.949
State_D.C.                            0.4131      0.239      1.728      0.092      -0.071       0.897
State_Florida                         0.3663      0.241      1.523      0.136      -0.121       0.853
State_Georgia                         0.2392      0.207      1.157      0.254      -0.179       0.658
```

Potential Issues

This multivariable regression can result in Multicollinearity where two or more variables are highly correlated. It is important to note that adding irrelevant variables to a regression model often causes the coefficient estimates to become less precise, therefore losing precision in the overall model.

Fun Statistics on 2023 Fortune 500 Company Data

```
#Which industries are most represented in the top 20 companies by revenue?
industry_counts = df['Industry'].value_counts()
print("Industries most represented in top 20: ")
print(industry_counts)
print('\n')

#Which companies have the highest and lowest revenue growth?
max_growth_company = df.loc[df['Revenue growth'].idxmax()]['Name']
min_growth_company = df.loc[df['Revenue growth'].idxmin()]['Name']
max_growth_value = df.loc[df['Revenue growth'].idxmax()]['Revenue growth']
min_growth_value = df.loc[df['Revenue growth'].idxmin()]['Revenue growth']

print(f"\nCompany with highest revenue growth: {max_growth_company} with a growth rate of {max_growth_value * 100:.2f}%")
print(f"Company with lowest revenue growth: {min_growth_company} with a growth rate of {min_growth_value * 100:.2f}%")

#Which companies have the highest and lowest employee counts?

max_employees_row = df.loc[df['Employees'].idxmax()]
min_employees_row = df.loc[df['Employees'].idxmin()]
max_employees_company = max_employees_row['Name']
max_employees_count = max_employees_row['Employees']
min_employees_company = min_employees_row['Name']
min_employees_count = min_employees_row['Employees']

print('\n')
print(f"Company with most employees: {max_employees_company} with {max_employees_count:,} employees")
print(f"Company with least employees: {min_employees_company} with {min_employees_count:,} employees")


#What is the average revenue and revenue growth for companies headquartered in different states?
df['State'] = df['Headquarters'].str.split(', ').str[1]
average_revenue_by_state = df.groupby('State')['Revenue (USD millions)'].mean()
average_growth_by_state = df.groupby('State')['Revenue growth'].mean()

print("\nAverage revenue by state:")
print(average_revenue_by_state)
print("\nAverage revenue growth by state:")
print(average_growth_by_state)
```

```
Industries most represented in top 20:
Financials                         11
Retail                             10
Petroleum industry                 10
Technology                          8
Pharmaceutical industry             7
Healthcare                          6
Insurance                           5
Conglomerate                        4
Telecommunications                  3
Airline                             3
Transportation                      2
Food industry                       2
Health Insurance                    2
Financial                           2
Food Processing                     2
Chemical industry                   1
Petroleum industry and Logistics    1
```

```
Machinery                          1
Agriculture manufacturing          1
Aerospace and Defense              1
Telecom Hardware Manufacturing     1
Agriculture cooperative            1
Apparel                            1
Infotech                           1
Automotive and Energy              1
Aerospace and defense              1
Food Service                       1
Logistics                          1
Consumer products Manufacturing    1
Retail and Cloud Computing         1
Media                              1
Beverage                           1
Financial services                 1
Automotive                         1
Automotive industry                1
Health                             1
Electronics industry               1
Laboratory instruments             1
Name: Industry, dtype: int64



Company with highest revenue growth: TD Synnex with a growth rate of 97.20%
Company with lowest revenue growth: Wells Fargo with a growth rate of 0.50%


Company with most employees: Walmart with 2,100,000 employees
Company with least employees: StoneX Group with 3,605 employees

Average revenue by state:
State
Arkansas          332285.500000
California        142740.000000
Connecticut       117269.000000
Cook County        51412.000000
D.C.              100108.000000
Florida            58776.333333
```

Colab paid products - Cancel contracts here

✓ 0s completed at 6:06 PM