Stock Market Price’s Growth Analysis

Team Video

Permissions

Place an `X` in the appropriate bracket below to specify if you would like your group's project to be made available to the public. (Note that student names will be included (but PIDs will be scraped from any groups who include their PIDs).

- [X] YES - make available
- [ ] NO - keep private

Overview

The stock prices of firms are direct representation of their well-being and potential development, which is associated with numerous indices calculated based on their balance sheet. In our research, we chose the return on equity (roe), price–earnings ratio (pe ratio), and net profit to total revenue out of all indices to investigate their influence on the changes in the stock prices. We conducted the analysis through the 2D scatterplot with overlaid regression line, drawing heatmap, and explore on the OLS regression result. To sum up, despite the statistically significant for some variables, our results prove little practical significance on the changes in the stock prices and the relationship between them are tenuous.

Names

- Irene Jiang
- Junlin Wu
- Shixuan Wu
- Mengyuan Zhang

Research Question

Is there a relationship between combination of (return on equity, pe ratio,net profit to total revenue) and Chinese stock price growth rate in percentage for companies within the consumer goods industry in 2014-2020?

Background & Prior Work

The Return on equity is highly deterministic for a company’s profitability, which is directly related to the company’s stock price and growth rate. Therefore we want to investigate if higher ROE would lead to higher stock price growth. More specifically, we want to investigate the relationship of ROE in the consumer industry as we found out that ROE varies a lot across industries and doesn’t make sense to compare different industries all together (1). As the article points out though, high ROE might not necessarily indicate the better performance of a company, so our group is going to investigate this specific ratio and get data from over hundreds of companies and over many years to see if ROE is in general, leading to higher stock price growth rate. We also looked at factors that could affect stock price. We discovered two factors that we totally did not know. Liquidity is one that tells how volatile a stock’s price is and demographics explains age groups’ preference to sell or keep or buy. We will try to incorporate data that have a measurement for stock’s volatility and average holder age to incorporate the two factors(2).

References (include links):

- 1) The ROE is the measure of a company's net income divided by its shareholders' equity. Investors usually look at whether the company’s ROE is bad or good before they make investment. ROE could vary in different industries. For example, in the utility industry, a good ROE could be around 10% while in the technology industry, companies usually have an ROE that is higher than 18%. However, A high ROE might not always be positive and could be misleading sometimes. An outsize ROE can be indicative of a number of issues—such as inconsistent profits or excessive debt, which is not directly linked to the company's profit.

  - https://www.investopedia.com/terms/r/returnonequ

- 2) This article talks about the factors that could affect the stock price and briefly discusses the reason behind each factor. The factors include inflation, substitutes, short-term trend, liquidity, etc. The most interesting two for us are liquidity which describes how volatile a stock's price is and demographics which shows that middle age people tend to buy into the market whereas older people like selling and quitting the market. These factors will help us in our course of finding the answer to our research question.

  - https://www.investopedia.com/articles/basics/04/10

Hypothesis

There is a positive correlation between the return on equity, pe ratio, and net profit to total revenue and the Chinese stock price's growth rate for companies within the consumer goods industry in 2014-2020.

Usually a company on the right track will have its stock price rising. The positive growth of that stock will attract more buyers which in turn will show the company what they are doing is the right move. As companies move in the right direction, its value will keep increasing and its stockholders can easily sell off the stock to get profit and thus higher return on equity.

Dataset(s)

- Dataset Name: JointQuant
- Link to the dataset: https://www.joinquant.com/
- Number of observations: 2470

The dataset consists all the stock's valuation metrics such as pe,pb ratios, and also the metrics that measures company's profit, such as the roe and net profit. It also consists the daily close price for each stock.

Variables Explanations

Variables Explanations

- Net Profit to Total Revenue
  - Gross profit is revenue - cost of the good or service. Net profit refers to gross profit minus the cost of the companies' operations like people's salaries, taxes, etc. Net profit to total revenue thus refers to the ratio of net profit versus the company's total revenue. If this number is high, it means that a large portion of the companies' income can be converted to profit, suggesting a high earning ability of the company.
- PE Ratio
  - PE ratio stands for Price-to-Earnings Ratio. It refers to the ratio between the price per share of the company versus earnings per share of the company. While price per share is easy to find by looking at the price of the stock, EPS (earnings per share) is calculated with the formula: a company's net profit divided by the number of common shares it has outstanding. Outstanding shares simply mean the total number of shares issued and actively held by stockholders—both outside investors and corporate insiders. Thus, a low PE ratio means that relative to the stock's money making ability, the price is too low, meaning that the stock is undervalued, and vice versa.
- ROE
  - ROE stands for Return On Equity. It is calculated by: net income of the company divided by shareholder equity. Net income refers to the total income of the company minus the cost, including both goods' or services' cost, companies' operation cost, and taxes. Shareholder equity is all the assets that the company owns minues all the liabilities like debts that the company has. ROE, therefore, measures the how well the company is using its equity to generate income. The higher the ROE, the better the company is in terms of generating profit.
- Price
  - In this analysis, price simply means price per

- In this analysis, price simply means price per stock measured in Chinese Yuan.

Setup

In [1]:

```
# Import seaborn and apply its plotting styles
import seaborn as sns
sns.set(font_scale=2, style="white")

# import matplotlib
import matplotlib as mpl
import matplotlib.pyplot as plt
import matplotlib.style as style
# set plotting size parameter
plt.rcParams['figure.figsize'] = (60, 20)

from jqdatasdk import *
import pandas as pd
from scipy import stats
import numpy as np
import patsy
import statsmodels.api as sm
```

In [2]:

```
auth("13160250367", "807207352xZ")


# get the stock price of a single stock
def get_single_stock_price(ticker, timePeriod, stad
    if stadate is None:
        stadate = get_security_info(ticker).start_d
    if endate is None:
        endate = datetime.datetime.today()
    data = get_price(ticker, start_date=stadate, en
                     frequency=timePeriod, panel=Fa
    return data
```

In [3]:

```
# check out the original raw data
# first type of table contains attributes such roe,
data1 = get_fundamentals(query(indicator), statDate
data1.head()
```

Out[3]:

	id	code	statDate	pubDate	statDate.1	
0	171577	002107.XSHE	2020-12-31	2021-01-21	2020-12-31	0.3
1	171578	300617.XSHE	2020-12-31	2021-01-27	2020-12-31	1.0
2	171579	002072.XSHE	2020-12-31	2021-01-30	2020-12-31	-1.1
3	171580	000001.XSHE	2020-12-31	2021-02-02	2020-12-31	1.4

4	171581	002984.XSHE	2020-12-31	2021-02-02	2020-12-31	1.64

5 rows × 36 columns

In [4]:

```
data1.describe()
```

Out[4]:

	id	eps	adjusted_profit	operatin
count	4250.000000	4250.000000	4.250000e+03	4.2500
mean	173753.451294	0.491632	8.169664e+08	9.2773
std	1244.741969	1.322690	8.943153e+09	8.8601
min	171577.000000	-10.270000	-5.569919e+10	-1.5619
25%	172676.250000	0.080000	1.027874e+07	2.8903
50%	173756.500000	0.320000	9.132726e+07	1.3102
75%	174831.750000	0.768825	3.181381e+08	4.1235
max	175946.000000	37.170000	3.140970e+11	2.7113

8 rows × 32 columns

In [5]:

```
# second type of table contains attributes such as p
data2 = get_fundamentals(query(valuation), statDate
data2.head()
```

Out[5]:

	id	code	pe_ratio	turnover_ratio	pb_ratio
0	62050271	600247.XSHG	-4.4233	1.7397	-3.5401
1	62047026	600634.XSHG	1.5413	NaN	5.4177
2	62046781	600146.XSHG	-5.0789	6.2738	0.6460
3	62047751	002071.XSHE	-0.5159	5.8861	-0.5961
4	62047720	603996.XSHG	-0.2552	1.0356	-0.5817

In [6]:

```
data2.describe()
```

Out[6]:

	id	pe_ratio	turnover_ratio	pb
count	4.140000e+03	4140.000000	4117.000000	4140.0

mean	6.204832e+07	-65.106651	2.604260	8.6
std	1.195259e+03	9631.093041	4.121261	286.1
min	6.204625e+07	-609435.630500	0.021500	-479.4
25%	6.204729e+07	10.961125	0.721400	1.6
50%	6.204832e+07	29.267950	1.412600	2.6
75%	6.204936e+07	58.077200	2.895000	4.4
max	6.205039e+07	95409.819900	75.692700	18394.5

In [7]:

```
# third type of table contains a single's stock info
get_single_stock_price("000001.XSHG", "daily",
                       stadate="2015-12-31",
                       endate="2015-12-31")
```

Out[7]:

	open	close	high	low	volume	
2015-12-31	3570.47	3539.18	3580.6	3538.35	1.769637e+10	2.54

Data Cleaning

1. We first acquire the value of roe, net_profit_to_total_revenue from the indicator part of database, then we also require the values of pe ratio from the valuation part of database
2. We change the column name to indicate the year of the data
3. By changing the type of data into a more workable type, the future operation will be easier.
4. We combine the column of different year of data into one table
5. We write the funciton to get the difference percentage point between the neighboring years.

In [8]:

```
#  here, we acquire data as well as clean them and p
def get_data(l: list, attr1: list, attr2: list) ->
    result = None
    # loop through each year
    for year in l:
        if result is None:
            # get each value using indiactor or valu
            roe = get_fundamentals(query(indicator)
            pe = get_fundamentals(query(valuation),
            # merge data we get from two query
            result = roe.merge(pe, on="code")
            result = roe.merge(pe, on="code")
            new_column_names = []
            # change the column names to indicate tl
            for i in result.columns:
                if i == "code":
                    new_column_names.append(i)
                else:
                    new_column_names.append(i + "_"
            result.columns = new_column_names
        else:
            # combines the roe and pe ratio into on
            roe = get_fundamentals(query(indicator)
            pe = get_fundamentals(query(valuation),
            new_data = roe.merge(pe, on="code")
            result = result.merge(new_data, on="cod

    new_column_names = []
    # append year into column name
    for i in result.columns:
        if i == "code":
            new_column_names.append(i)
            continue
        try:
            int(i[-4:])
            new_column_names.append(i)
        except ValueError as E:
            new_column_names.append(i + "_{}".forma
    result.columns = new_column_names

    # get price
    for year in l:

        # print so that we know the progress
        print("Start getting stock prices of {}".fo

        # 12-31 could be on a weekend and no trade
        # Therefore, we use try and except to get d
        try:
            result["price_{}".format(year)] = resul


        except KeyError as e:
            try:
                result["price_{}".format(year)] = r


            except KeyError as e:
                try:
                    result["price_{}".format(year)]


                except KeyError as e:
                    result["price_{}".format(year)]


        # print so that we know the progress
        print("Finished getting stock prices of {}".

    return result.reset_index(drop=True)
```

In [9]:

```
%%time
# get data we need
result = get_data(list(range(2014, 2021)), ["code",
```

In [10]:

```
# save the data into a csv file since getting it eve
result.to_csv("data/stock_info_through_the_years.cs
```

In [11]:

```
# read the csv file we have to see
# This is total around 2000 observation we acquire
data = pd.read_csv("data/stock_info_through_the_yea
data.shape
```

Out[11]:

In [12]:

```
# get a look of our data
data.head()
```

Out[12]:

	code	roe_2014	net_profit_to_total_revenue_2014
0	000099.XSHE	7.9019	15.3643
1	000096.XSHE	2.2570	4.3263
2	000090.XSHE	11.9606	7.9548
3	600783.XSHG	9.5333	132.5205
4	600782.XSHG	5.1865	1.3226

5 rows × 29 columns

In [13]:

```
# our data have no missing values since we used drop
data.isna().any().any()
```

Out[13]:

In [14]:

```
# take a look of all industries of stocks
get_industries().head()
```

Out[14]:

	name	start_date
L72	商务服务业	1996-08-29
L71	租赁业	1997-01-30
G53	铁路运输业	1998-05-11
G57	管道运输业	1996-11-04
G56	航空运输业	1997-11-05

In [15]:

```
# select consumer industries
consumer_industries = [i for i in get_industries().
```

In [16]:

```
# get stocks that are in consumer industry
consumer_industries_stocks = set()
for i in consumer_industries:
    [consumer_industries_stocks.add(stock) for stoc
pd.Series(list(consumer_industries_stocks)).head()
```

Out[16]:

In [17]:

```
# filter data
data = data.loc[data["code"].isin(consumer_industri
```

In [18]:

```
data.shape
```

Out[18]:

In [19]:

```
data.isna().any().any()
```

Out[19]:

How clean is the data?

We make sure the data type of the values in each columns is correct and ready to do any operation. For example, we changed the object type to string and check the values for roe and pe ratio are float. We will also calculate the changes in percentage of the roe, pe ratio, and net profit to total revenue, so that we could better compare them and draw any possible relation.

Then, for the preprocessing step we plan to do, we would check the distribution of our data. We will make sure that the roe, pe ratio, and net profit to total revenue are not linearly independent, which could potentially sabotage the accuracy of our research.

Also, we will plot the data to see whether they are normally distributed, and changes they into log scale if necessary. Once the data are normally distributed, we could apply the t test to find out the R square and other statistical parameters and answer our research questions.

Data Analysis & Results

EDA

In [20]:

```
# calculate growth of row, price, and pe ratio from
for year in range(2015, 2021):
    data["roe_growth_{}".format(year)] = (data["roe
    data["price_growth_{}".format(year)] = (data["p
    data["pe_ratio_growth_{}".format(year)] = (data
```

In [21]:

```
# look at the data with new columns
data.head()
```

Out[21]:

	code	roe_2014	net_profit_to_total_revenue_2014
1	000096.XSHE	2.2570	4.3263
4	600782.XSHG	5.1865	1.3226
5	600781.XSHG	4.0889	3.6644
8	600785.XSHG	11.8761	2.3250
9	600784.XSHG	3.0142	1.2086

5 rows × 47 columns

In [22]:

```
# look all data statistics
data.describe()
```

Out[22]:

	roe_2014	net_profit_to_total_revenue_2014	pe_ratio_
count	1597.00000	1597.000000	1597.0
mean	6.56847	6.510792	1.0
std	14.46934	39.368182	1400.7
min	-235.36910	-756.599400	-43476.5
25%	2.48860	1.929500	22.7
50%	6.47910	5.333900	41.8
75%	11.67520	11.877200	81.7
max	102.96910	693.621100	5496.1

8 rows × 46 columns

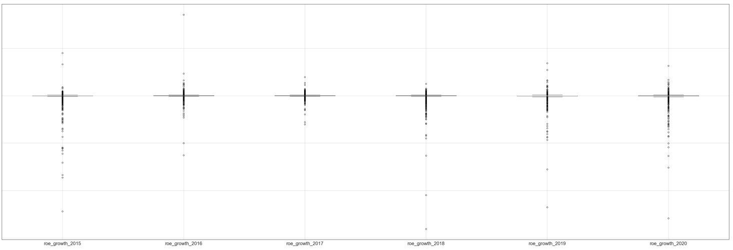
Roe_growth vs Price_growth

We plot the box plot for the roe_growth throught out the years to find out range for filtering the outlier.

In [23]:

```
data[["roe_growth_2015", "roe_growth_2016", "roe_gr
```

Out[23]:



The boxplot clearly shows us the outlier data. According to the boxplot, most of the stocks see a growth or decline no more than single digit percent. The distribution mostly centers at zero. Many of our outliers have roe growth more than 30 percent, which is why we will choose to eliminate data points that deviate from the 0 by 30 more. The shape of the distribution is hard to see in this graph due to outliers, so we will explore the shape after excluding the outliers in scatter plot for better visualization.

Through that, we can filter the roe_growth data using the range selected.

In [24]:

```
data_drop_outliers = data
for i in range(2015, 2021):
    data_drop_outliers = data_drop_outliers[(data_d
```

Note that most of the outliers are companies that has been extremely successful or struggling in the recorded year. They will not be helpful in finding out the overall population trend that we are trying to seek as most companies only see mild growth or decline in a year.

Take a look at the data without outlier

In [25]:

```
data_drop_outliers.head()
```

Out[25]:

	code	roe_2014	net_profit_to_total_revenue_2014
1	000096.XSHE	2.2570	4.3263
4	600782.XSHG	5.1865	1.3226
5	600781.XSHG	4.0889	3.6644
8	600785.XSHG	11.8761	2.3250
9	600784.XSHG	3.0142	1.2086

5 rows × 47 columns

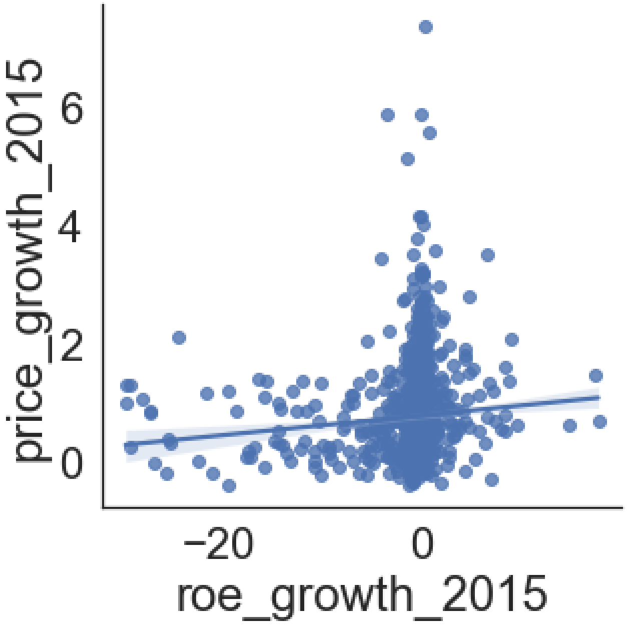
5 rows × 47 columns

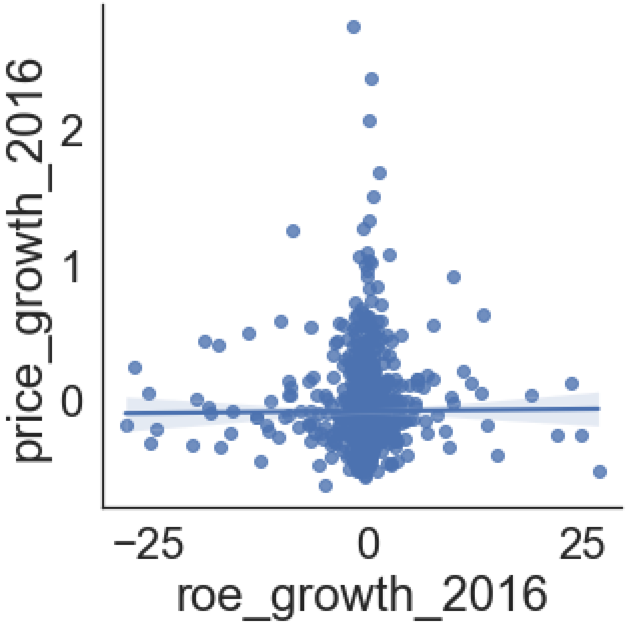
We use scatter plot to see the distribution of the roe_growth after we drop the outlier and its relationship with the changes in prices.

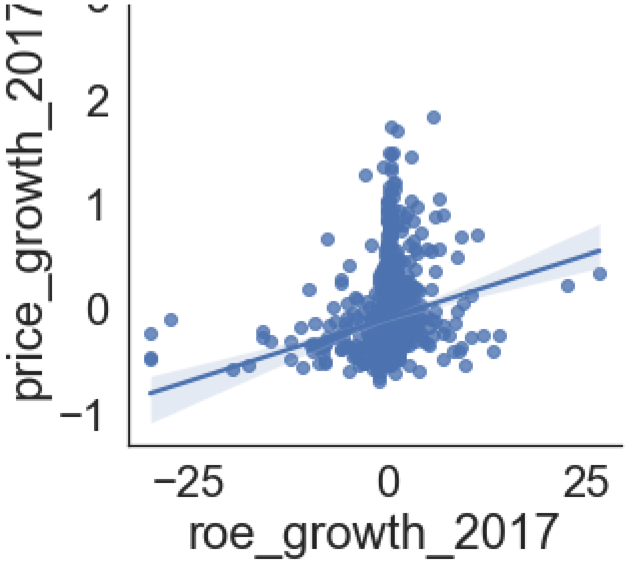
In [26]:

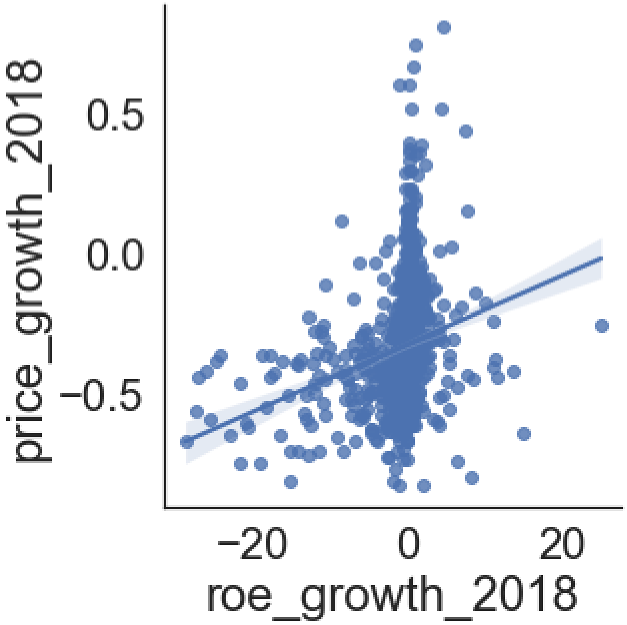
```
sns.lmplot(x="roe_growth_2015", y="price_growth_201
sns.lmplot(x="roe_growth_2016", y="price_growth_201
sns.lmplot(x="roe_growth_2017", y="price_growth_201
sns.lmplot(x="roe_growth_2018", y="price_growth_201
sns.lmplot(x="roe_growth_2019", y="price_growth_201
sns.lmplot(x="roe_growth_2020", y="price_growth_202
```

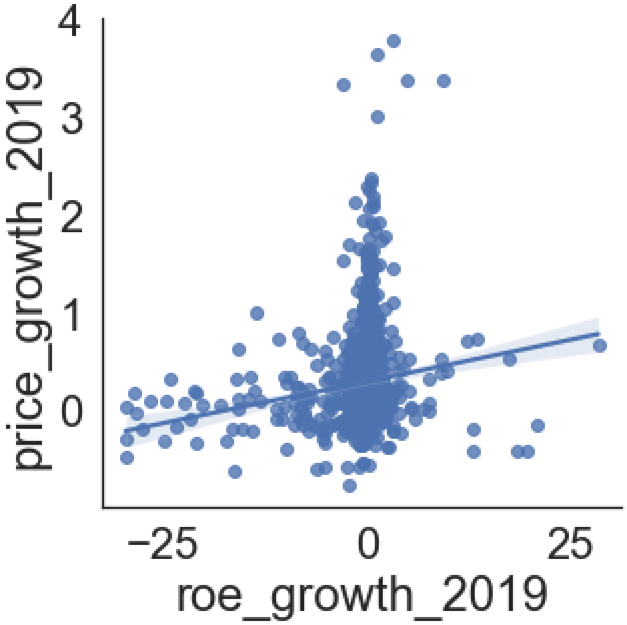
Out[26]:



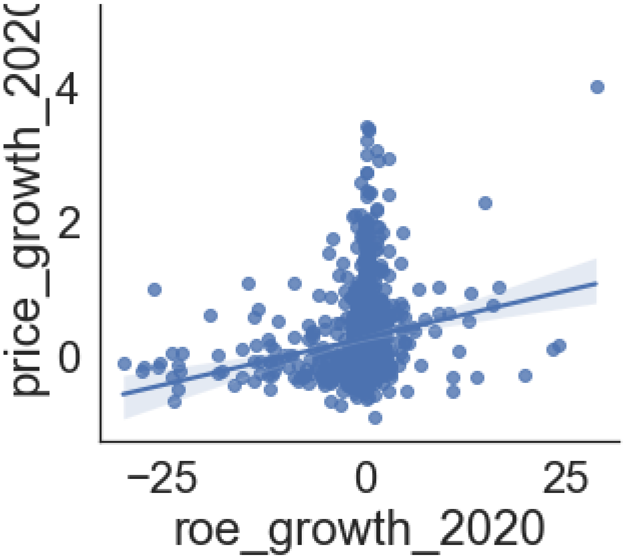



price_growth_2017
2
1
0
−1
−25
0
25
roe_growth_2017


price_growth_2018
0.5
0.0
−0.5
−20
0
20
roe_growth_2018


price_growth_2019
4
3
2
1
0
−25
0
25
roe_growth_2019

6



The trend is very similar across the year. Most of the stocks have roe growth centers around zero. However, their price growth has a larger variance. We can often see stocks that have zero roe growth but with high price growth. After analysis, we hypothesized that this may be due to those stocks' companies already having a high roe growth in the previous year. If such companies can keep that high roe, more investors will be attracted which will lead to higher demand for the stocks and thus higher price.

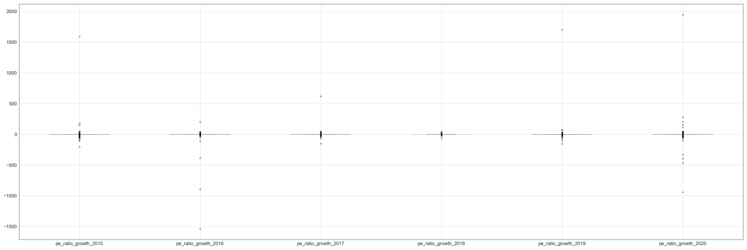
pe_ratio_growth vs price_growth

We plot the box plot for the pe_ratio_growth throught out the years to find out range for filtering the outlier.

In [27]:

```
data_drop_outliers[[″pe_ratio_growth_2015″, ″pe_rat
```

Out[27]:



From the boxplot we can clearly see that the most of the data is around -100 to 100. The distribution of the data mostly on -300 and 300. We can see there have some outliers, most of outiers are far away from the center of data. Since the outlier will affect our result, we will drop the outliers. Since most data is between -300 and 300, we decide to filter the data outside this range.

The boxplot cannot be clearly show the relation ship between pe_ratio_growth and price_growth. We want to use the lmplot to show the corralation between those to variables after we drop the outliers.

In [28]:

```
for i in range(2015, 2021):
    data_drop_outliers = data_drop_outliers[(data_d
```

In [29]:

```
data_drop_outliers.head()
```

Out[29]:

	code	roe_2014	net_profit_to_total_revenue_2014
1	000096.XSHE	2.2570	4.3263
4	600782.XSHG	5.1865	1.3226
5	600781.XSHG	4.0889	3.6644
8	600785.XSHG	11.8761	2.3250
9	600784.XSHG	3.0142	1.2086

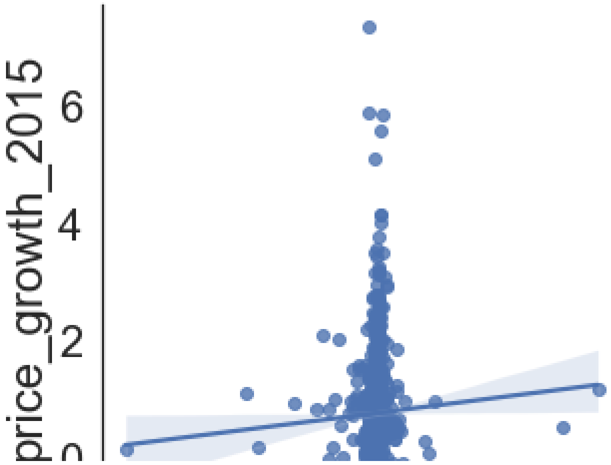
5 rows × 47 columns

We want to plot the relationship between pe_ratio_growth and the price growth using line plot. By observing the estimated line, we could tell their relationship based on the gradient.

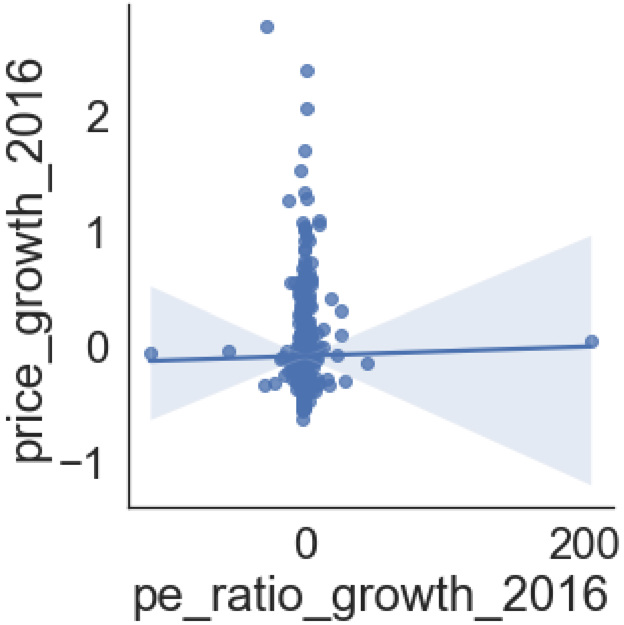
In [30]:

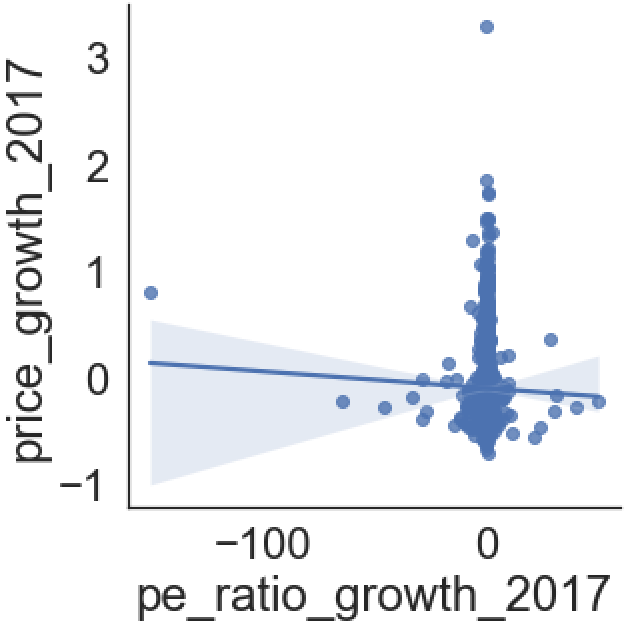
```
sns.lmplot(x="pe_ratio_growth_2015", y="price_growt
sns.lmplot(x="pe_ratio_growth_2016", y="price_growt
sns.lmplot(x="pe_ratio_growth_2017", y="price_growt
sns.lmplot(x="pe_ratio_growth_2018", y="price_growt
sns.lmplot(x="pe_ratio_growth_2019", y="price_growt
sns.lmplot(x="pe_ratio_growth_2020", y="price_growt
```

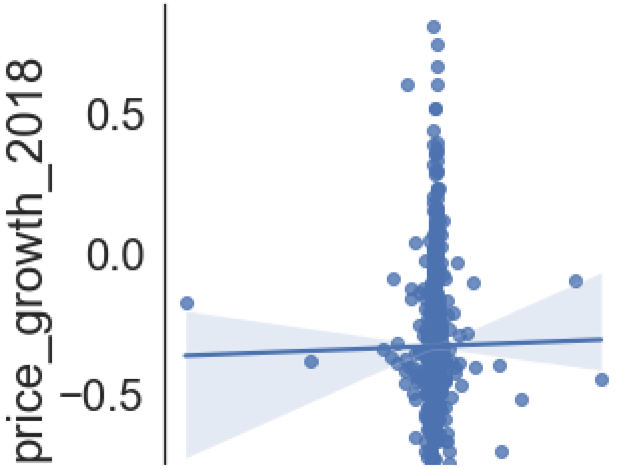
Out[30]:

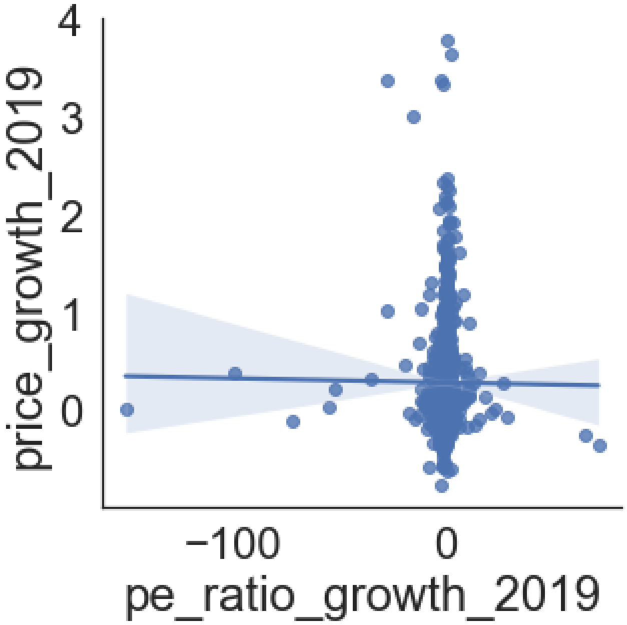


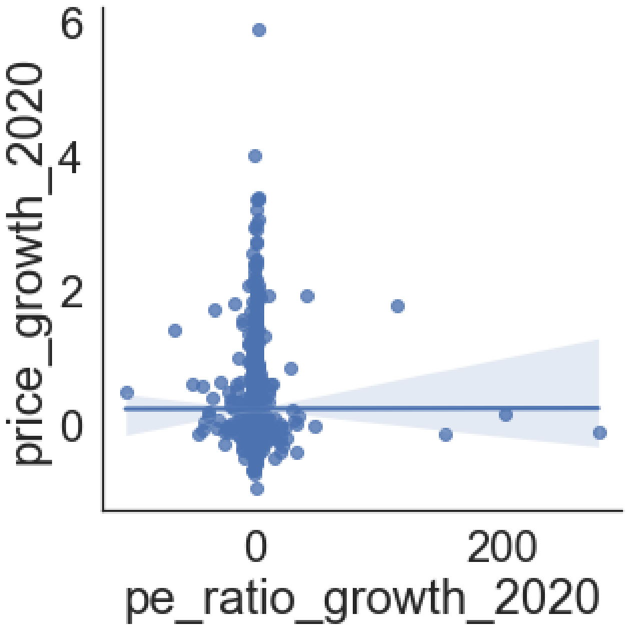
0
−200
0
pe_ratio_growth_2015


price_growth_2016
2
1
0
−1
0
200
pe_ratio_growth_2016


price_growth_2017
3
2
1
0
−1
−100
0
pe_ratio_growth_2017


price_growth_2018
0.5
0.0
−0.5





From the lmplot, we can directly see that all years' pe_ratio_growth is around the 0. Although most of the pe_ratio_growth is around 0, but it has the different value of price_growth. The trend of each year is similar but still have a little difference. The relationship between pe_ratio_growth and price_growth in 2015, 2016, 2018, and 2020 is positive. With the pe_ratio_growth increse, the price_groth will also increse. Otherwise, the relationship between pe_ratio_growth and price_growth in 2017 and 2019 is negative. With the pe_ratio_growth increse, the price_groth will also decrese.

increse, the price_groth will also decrese.

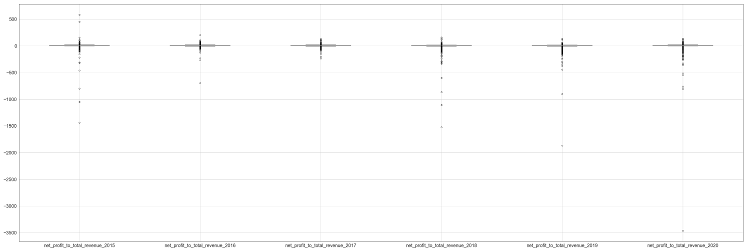
net_profit_to_total_revenue vs price_growth

We plot the box plot for the net_profit_to_total_revenue throughout out the years to find out range for filtering the outlier.

In [31]:

```
data_drop_outliers[["net_profit_to_total_revenue_20
```

Out[31]:



The boxplot clearly shows us the range of all data, which is useful to identify the outlier. According to the boxplot, most of the stocks see the net_profit_to_total_revenue growth or decline no more than 10%. The distribution mostly centers at zero. The extreme values of the net_profit_to_total_revenue are up to -3500 to 550. Based on the plot, we decide to filter the range from 300 to -300. In this case, we can remove almost all values of the extreme data. The shape of the distribution is all concentrated at 0 and hard to see in this graph due to outliers, so we will explore the shape after excluding the outliers in scatter plot for better visualization.

After this, we can filter the net_profit_to_total_revenue data using the range selected [300, -300].

In [32]:

```
for i in range(2015, 2021):
    data_drop_outliers = data_drop_outliers[(data_d
```

Check the data after we drop the outliers

In [33]:

```
data_drop_outliers.head()
```

Out[33]:

	code	roe_2014	net_profit_to_total_revenue_2014
1	000096.XSHE	2.2570	4.3263
4	600782.XSHG	5.1865	1.3226
5	600781.XSHG	4.0889	3.6644
8	600785.XSHG	11.8761	2.3250

8	600783.XSHG	11.0701	2.5256
9	600784.XSHG	3.0142	1.2086

5 rows × 47 columns

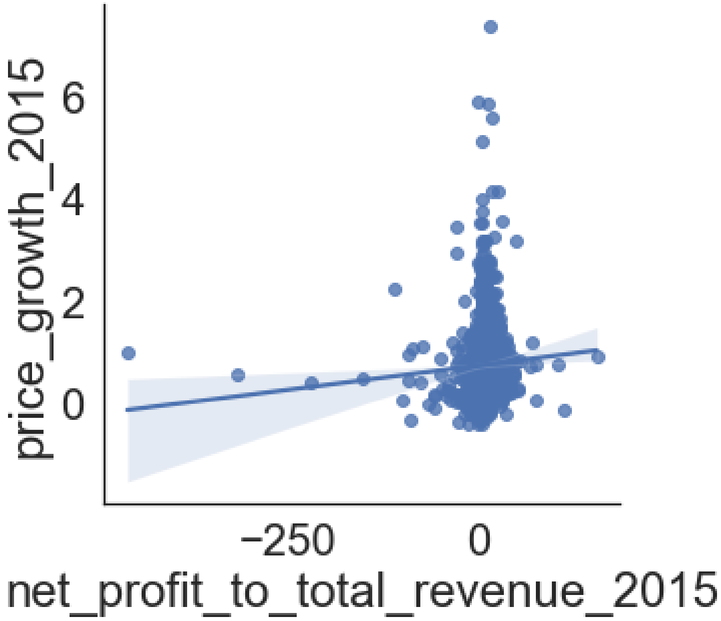
Note that most of the outliers are companies that has been barely keeping up with the market and losing enormous revenue, and only a few of them generate abnormal profit. These outliers will not be helpful in finding out the general trend between net_profit_to_total_revenue and the changes in stock price that we are trying to seek, since most companies merely see small growth or decline in a year.

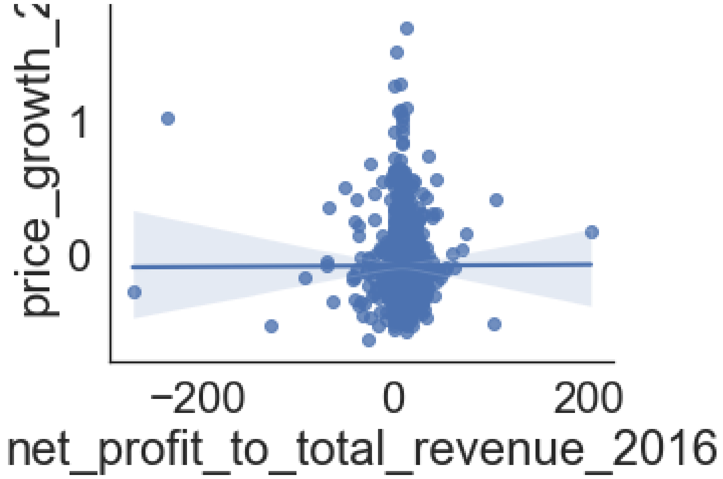
We want to plot the relationship between net_profit_to_total_revenue and the price growth using line plot. By observing the estimated line, we could tell their relationship based on the gradient.

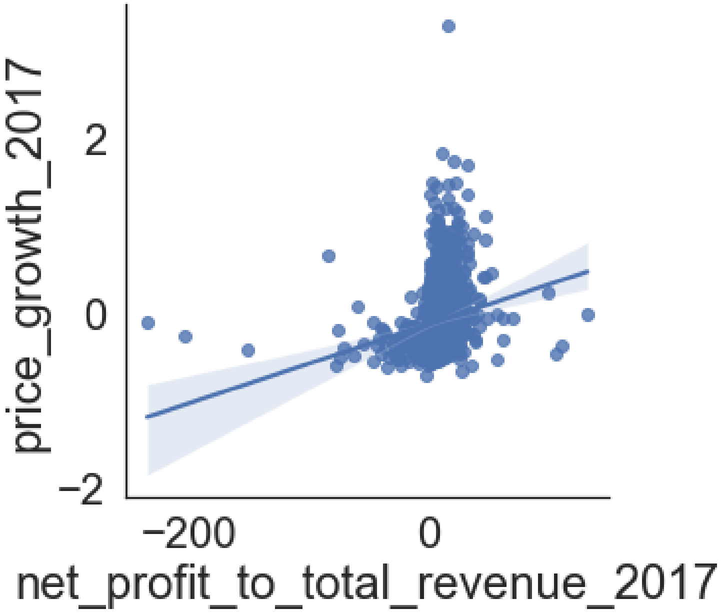
In [34]:

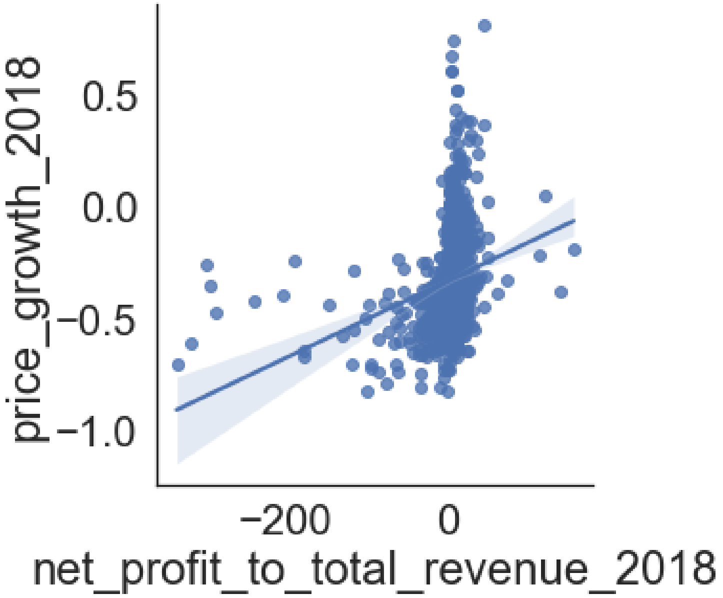
```
# plot the net_profit_to_total_revenue against the p
sns.lmplot(x="net_profit_to_total_revenue_2015", y=
sns.lmplot(x="net_profit_to_total_revenue_2016", y=
sns.lmplot(x="net_profit_to_total_revenue_2017", y=
sns.lmplot(x="net_profit_to_total_revenue_2018", y=
sns.lmplot(x="net_profit_to_total_revenue_2019", y=
sns.lmplot(x="net_profit_to_total_revenue_2020", y=
```

Out[34]:

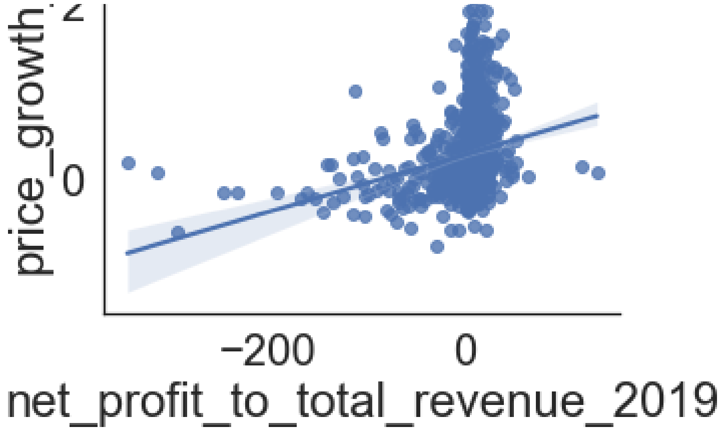


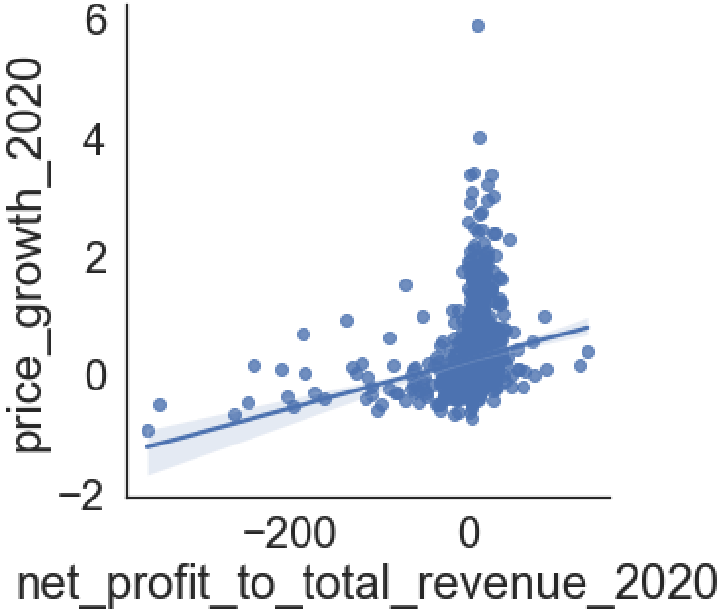

price_growth_2
1
0
−200
0
200
net_profit_to_total_revenue_2016


price_growth_2017
2
0
−2
−200
0
net_profit_to_total_revenue_2017


price_growth_2018
0.5
0.0
−0.5
−1.0
−200
0
net_profit_to_total_revenue_2018

_2019
4





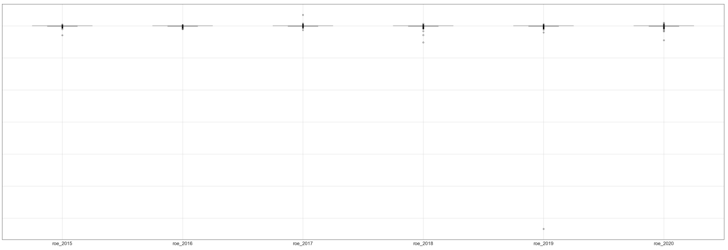
From the implot, we can see overall positive relationship between net_profit_to_total_revenue and the changes in stock price. Most of the stocks have net_profit_to_total_revenue centers around zero, which means that they are barely earning any economic profit. Only the data from 2016 shows a flat line that means net_profit_to_total_revenue does not change in stock price. This special case could be contributed to the 2015–2016 stock market selloff. By early 2016, global stock markets were falling hard. Negative economic reports from China caused panic selling. Interest rates fell sharply, and there were widespread warnings of deflation and depression. This economic recession has global impact, so we could ignore this when we summarize the general trend. For firms that do have net_profit_to_total_revenue greater than 0, no matter the degree, these data manage to move up the gradient and show an overall positive relationship between net_profit_to_total_revenue and the changes in stock price. The blur shadow area is the marginal of error. Within that area, we can still witness a positive correlation. This trend is understandable, since the higher ratio means that the firms that make higher profit with less revenue and have higher profit margin will have higher stock price on that year. The investor believes that firms with higher profit marginal have better potential, and therefore more people buy their stock. Consequently, the stock prices raise.

Price_growth v.s. roe

In [35]:

```
data[["roe_2015", "roe_2016", "roe_2017", "roe_2018
```

Out[35]:



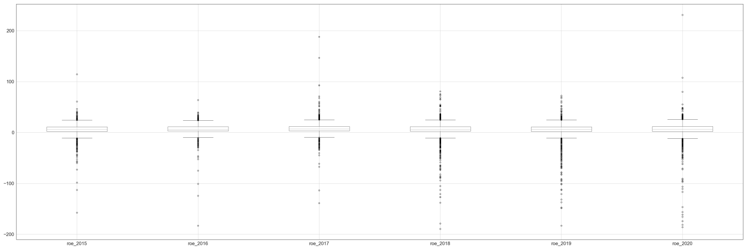
In [36]:

```
for i in range(2015, 2021):
    data_drop_outliers = data_drop_outliers[(data_d
```

In [37]:

```
data_drop_outliers[["roe_2015", "roe_2016", "roe_20
```

Out[37]:

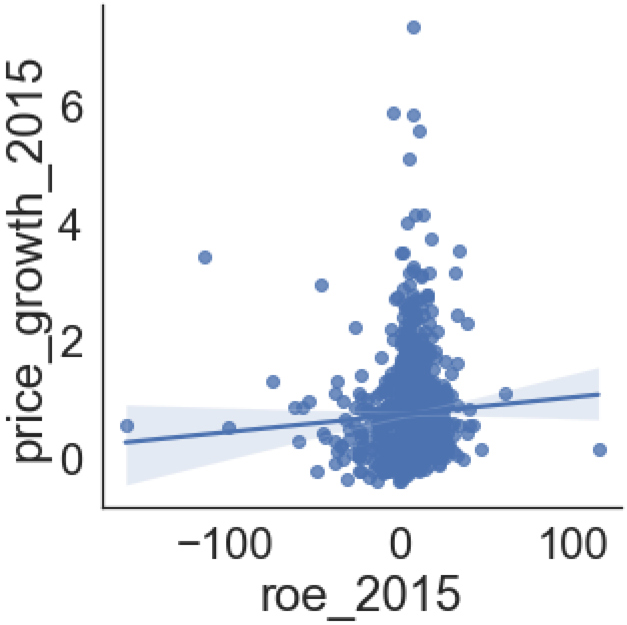


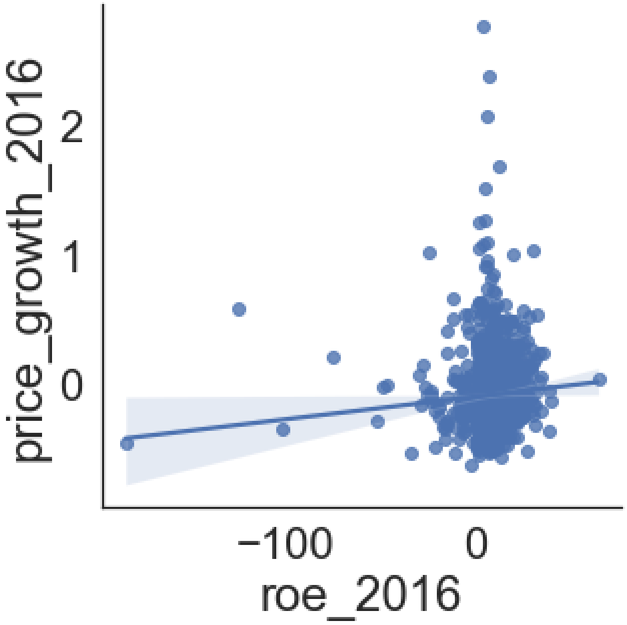
This boxplot shows us the outlier data for the roe distribution. According to the boxplot, most of the stocks have a roe value between -100 to 50. The distribution mostly centers at zero. Thus we eliminate the outliers that have roe more than 300 and outliers that have roe less than -500. The shape of the distribution is hard to see in this graph due to outliers, so we will explore the shape after excluding the outliers in the scatter plot for better visualization. After we excluded the outlier, we could see the median roe value for each year is a little bit above 0, around 5, with 25% quartile of around 7%, 75% quartile of around 1%.

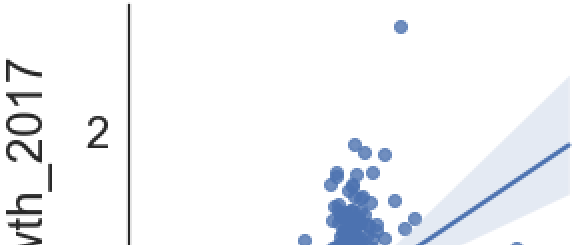
In [38]:

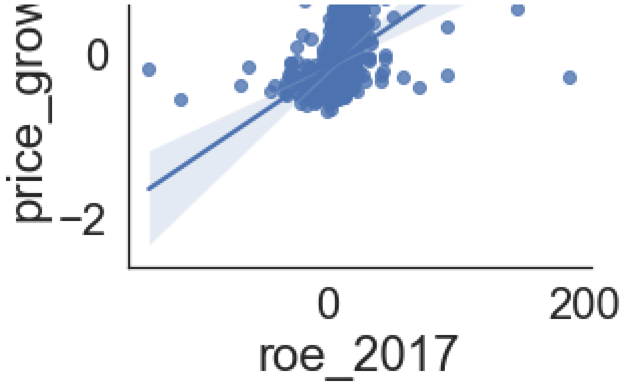
```
sns.lmplot(x="roe_2015", y="price_growth_2015", dat
sns.lmplot(x="roe_2016", y="price_growth_2016", dat
sns.lmplot(x="roe_2017", y="price_growth_2017", dat
sns.lmplot(x="roe_2018", y="price_growth_2018", dat
sns.lmplot(x="roe_2019", y="price_growth_2019", dat
sns.lmplot(x="roe_2020", y="price_growth_2020", dat
```

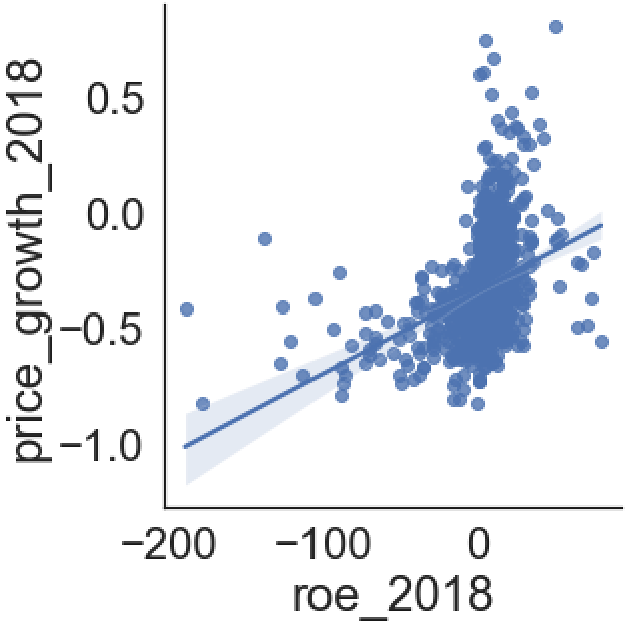
Out[38]:

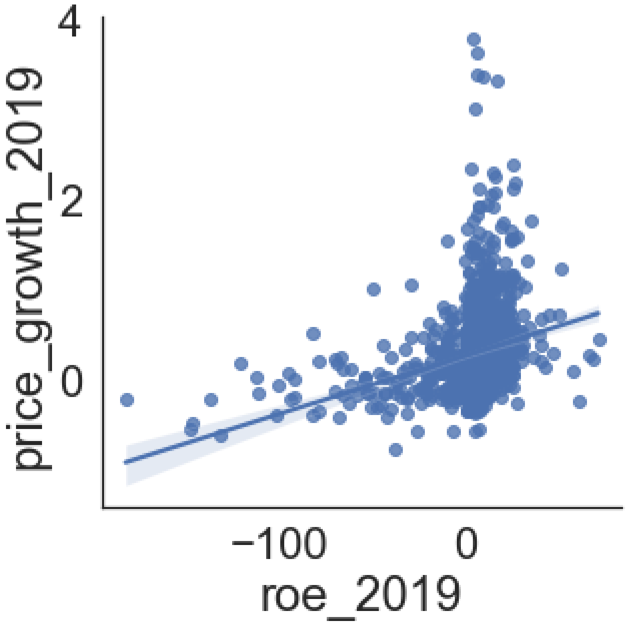


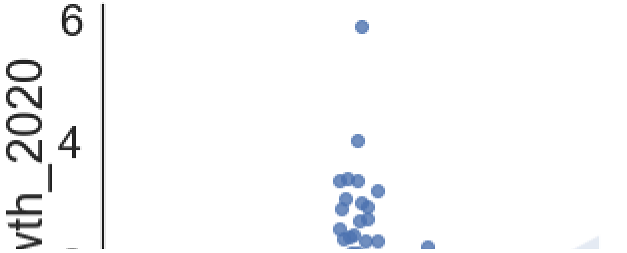


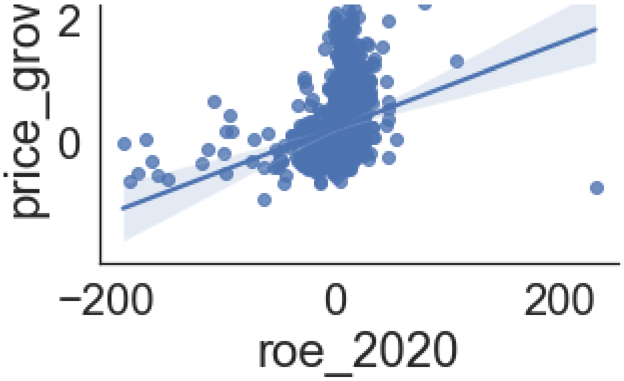



price_grow
0
−2
0
200
roe_2017


price_growth_2018
0.5
0.0
−0.5
−1.0
−200
−100
0
roe_2018


price_growth_2019
4
2
0
−100
0
roe_2019


vth_2020
6
4



The trend is very similar across the year. On average, we see an upward trend with higher roe relate to higher price growth for each year. Most of the stocks have roe growth centers around zero. However, their price growth has a larger variance. We can often see stocks that have around zero roe but with high price growth.

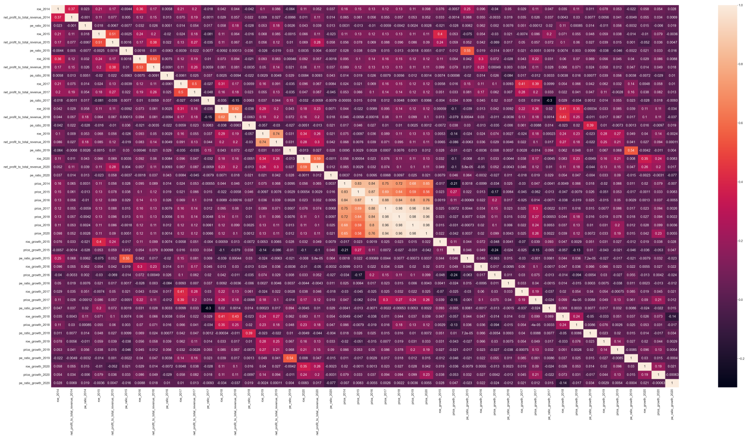
Visualize the Correlation using Heatmap

We want to use the heatmap to see is there any variables which has higher relationship with the price_growth.

In [39]:

```
plt.subplots(figsize=(100,50))
sns.heatmap(data_drop_outliers.corr(), annot=True)
```

Out[39]:



The lighter color indicates the stronger positive relationship. From the heatmap, we can see that the correlation coefficients of price_growth and other variables are not high, that means the price_growth do not have strong relation with the other variable. This also shows that the relationship between price_growth and other variable is little.

OLS

We make the function to see whether net_profit_to_total_revenue, roe_growth, and pe_ratio_growth will lead the corresponding changes in the stock prices. The OLS result summary will generate the information will tell us the whether we can reject the null hypothesis and there is a causal relationship between our variables.

In [40]:

```
for i in range(2015, 2021):
    df = data_drop_outliers[['net_profit_to_total_r
                             'price_growth_{}'.form
                            ]]
    df.columns = ['net_profit_to_total_revenue', 'r
    outcome, predictors = patsy.dmatrices('price_gro
    mod_log = sm.OLS(outcome, predictors)
    res_log = mod_log.fit()
    print(res_log.summary())
```

Result

Across the yers, we see pe_ratio growth and net profit to total revenue always having p values. This suggests that pe ratio growth and net profit to total revenue have little correlation with price growth. For roe growth and roe, both do not show statistical significance in some years. Even in the years that they do have statistical significance, the estimated coefficient is very small, with the largest being 0.0209, which means that at best, with one unit increase in roe growth or net profit to total revenue, we will see at most 0.0209 unit increase in price growth. However with r-squared and adjusted R-squared values being so close to zero for all the years, we can conclude that these variables only explains a tiny portion of the variance in price growth and provides only negligible benefits in real world applications. Thus, we see no strong or useful relationship between price growth vs. roe growth, pe ratio growth, net profit to total revenue, and roe.

Further Analysis

Since many of the data points have a wide range of x values, we want to replot the relationship above but with log of the x values and see if there are any interesting relationship.

In [41]:

```
cols_to_be_logged = data_drop_outliers.columns.toli
```

In [42]:

```
for i in cols_to_be_logged:
    data_drop_outliers[i + "_logged"] = np.log(data
```

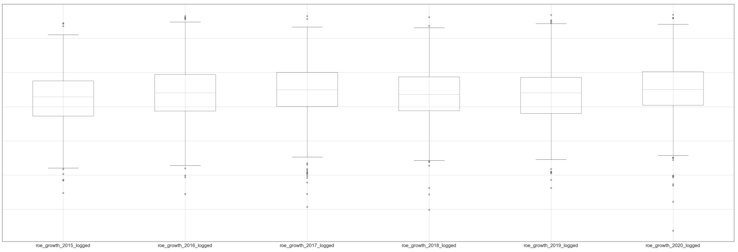
```
C:\ProgramData\Anaconda3\lib\site-packages\pandas\co
re\arraylike.py:364: RuntimeWarning: invalid value e
ncountered in log
  result = getattr(ufunc, method)(*inputs, **kwargs)
C:\ProgramData\Anaconda3\lib\site-packages\pandas\co
re\arraylike.py:364: RuntimeWarning: divide by zero
encountered in log
  result = getattr(ufunc, method)(*inputs, **kwargs)
```

Price Growth vs. log(roe_growth)

In [43]:

```
data_drop_outliers[["roe_growth_2015_logged", "roe_g
```

Out[43]:



For roe growth logged, most of the yars have means around -1. The data generally ranges from 2 to -6 with points outside of the range being outliers. We can see that the range from 0 to -2 covers most of the years' middle 50 percent values. Based on this boxplot, we see not too many outliers comparing to graphs in the previous sections. Thus we decided not to drop any more points

Now we redraw the line plot to see if there are any relationship between roe growth logged and price growth.

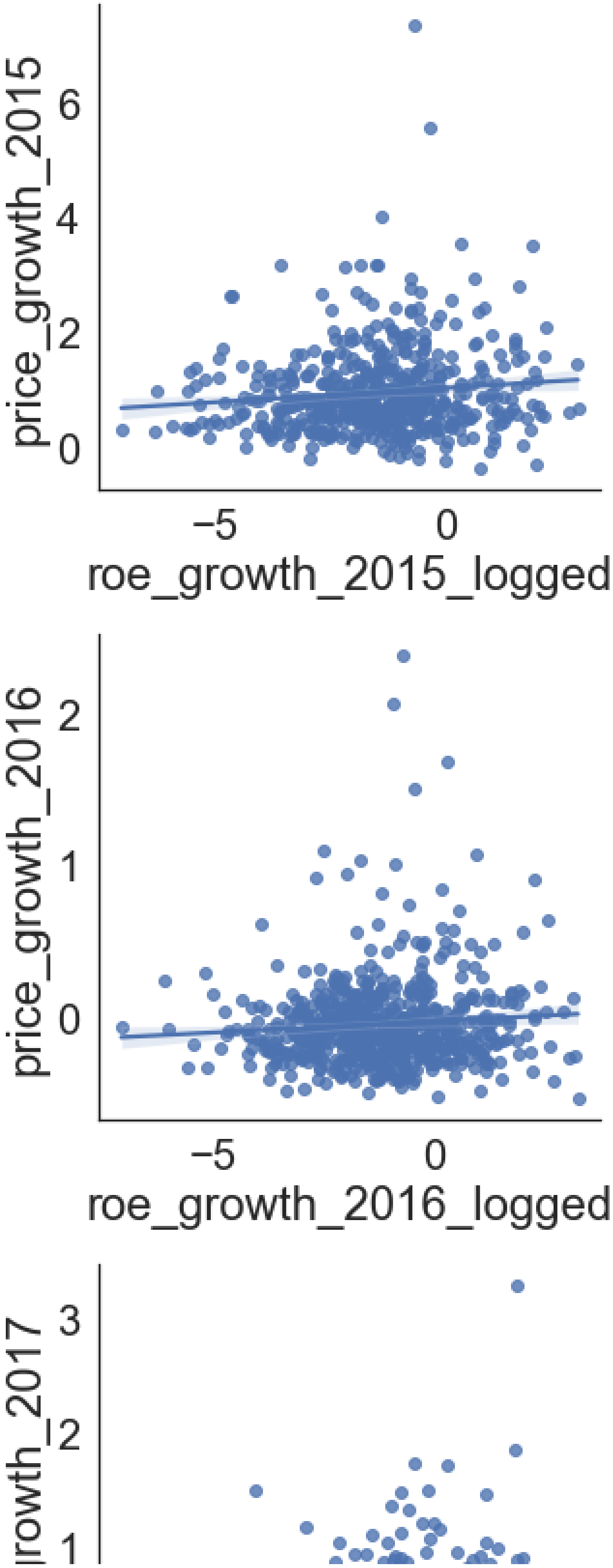
In [44]:

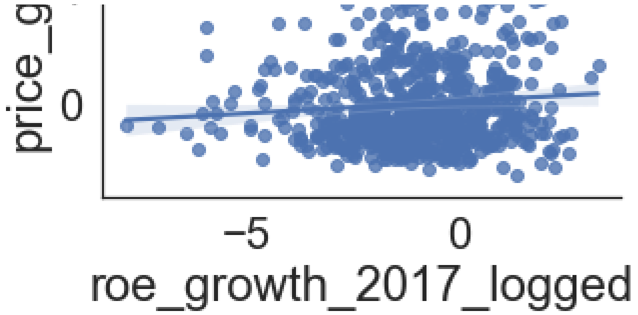
```
sns.lmplot(x="roe_growth_2015_logged", y="price_gro
sns.lmplot(x="roe_growth_2016_logged", y="price_gro
sns.lmplot(x="roe_growth_2017_logged", y="price_gro
sns.lmplot(x="roe_growth_2018_logged", y="price_gro
sns.lmplot(x="roe_growth_2019_logged", y="price_gro
sns.lmplot(x="roe_growth_2020_logged", y="price_gro
```

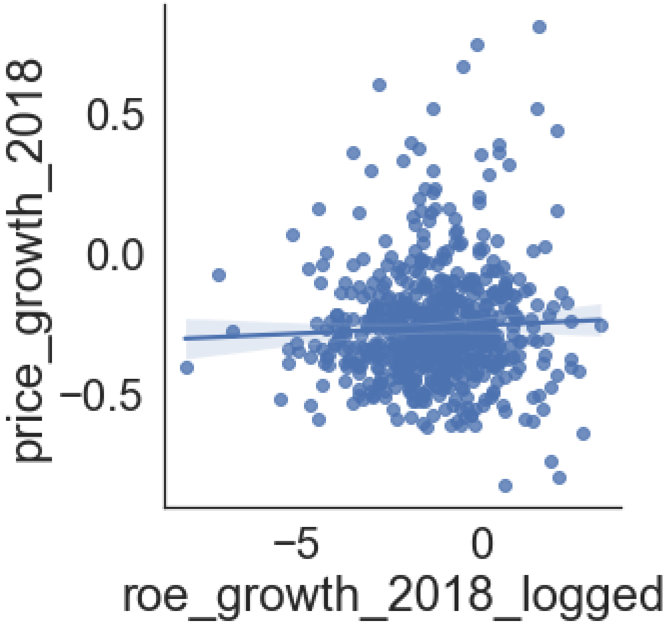
```
C:\ProgramData\Anaconda3\lib\site-packages\numpy\cor
e\function_base.py:151: RuntimeWarning: invalid valu
e encountered in multiply
  y *= step
C:\ProgramData\Anaconda3\lib\site-packages\numpy\cor
```

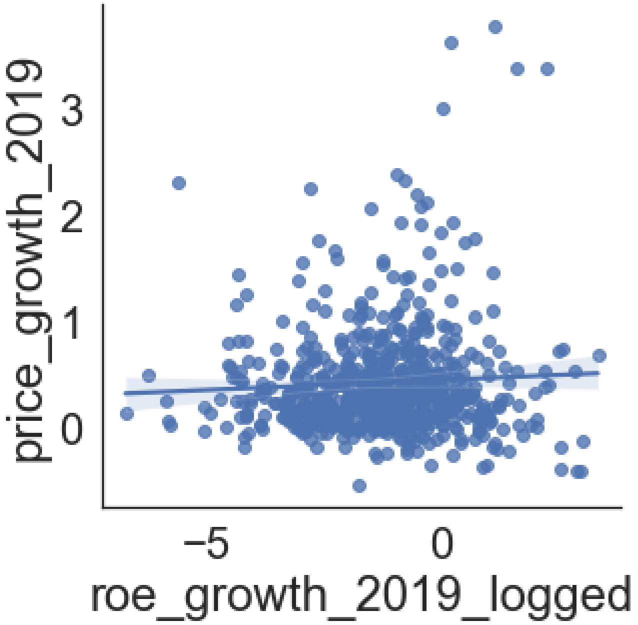
```
C:\ProgramData\Anaconda3\lib\site-packages\numpy\cor
e\function_base.py:161: RuntimeWarning: invalid valu
e encountered in add
  y += start
C:\ProgramData\Anaconda3\lib\site-packages\numpy\lib
\nanfunctions.py:1389: RuntimeWarning: All-NaN slice
encountered
  result = np.apply_along_axis(_nanquantile_1d, axi
s, a, q,
```

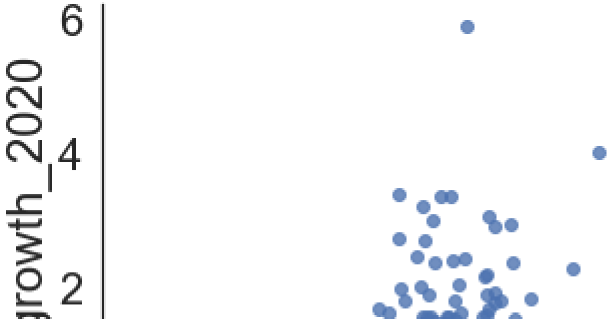
Out[44]:

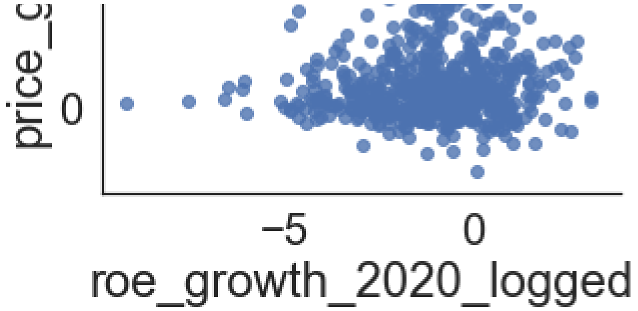



price_g
0
−5
0
roe_growth_2017_logged


price_growth_2018
0.5
0.0
−0.5
−5
0
roe_growth_2018_logged


price_growth_2019
3
2
1
0
−5
0
roe_growth_2019_logged


growth_2020
6
4
2



The relationship is now much clearer with the data points more clustered together around the central line than before. Most of the outliers are easy to spot in the graph. Many of them are companies with high price growth but with low roe growth logged. These are the companies that most likely have a high roe in the previous year and keeping that high roe is already a feat, which will definitely attract more investors and increase stock price.

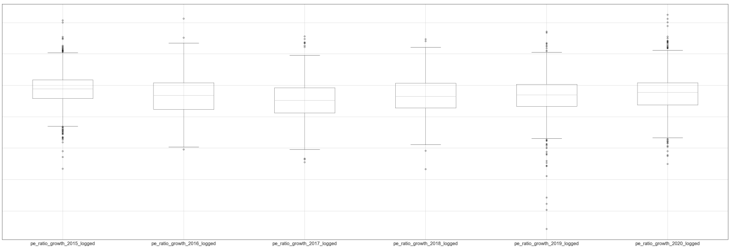
Price Growth vs. log(pe_ratio_growth)

We plot the box plot for the price_growth and log(pe_ratio_growth) to find there has any outliers.

In [45]:

```
data_drop_outliers[["pe_ratio_growth_2015_logged", "
```

Out[45]:

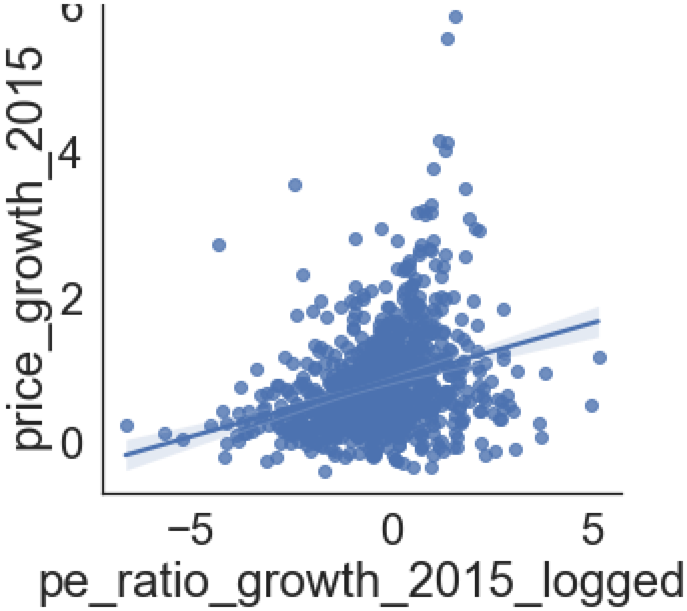


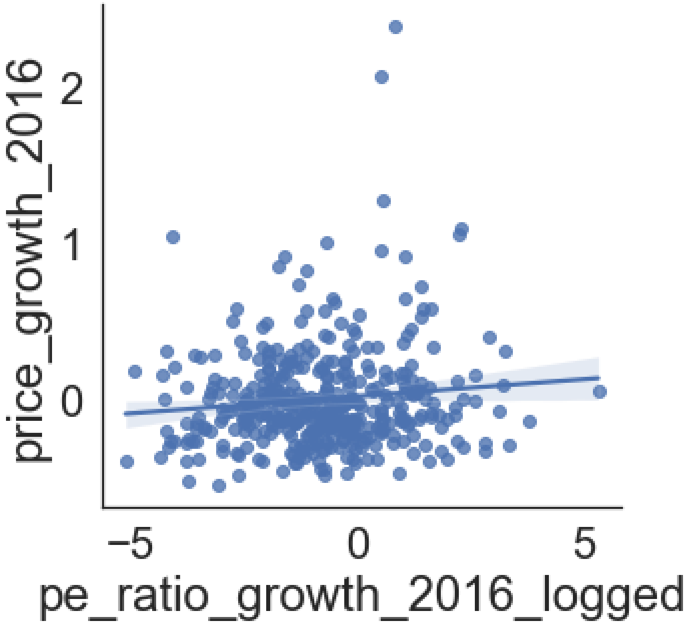
After log the pe_ratio_growth, we can see that the range of each year's pe_ratio growth are not very big. The distribution of the data are more concentrated on the -5 to 5. There still exits some outliers for each year's data, but those are not extramly far from the center of data, we don't remove those outliers.

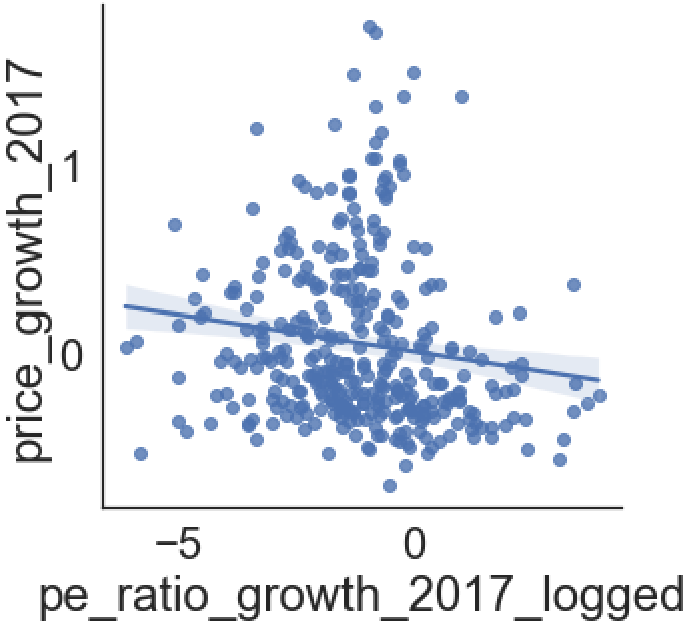
In [46]:

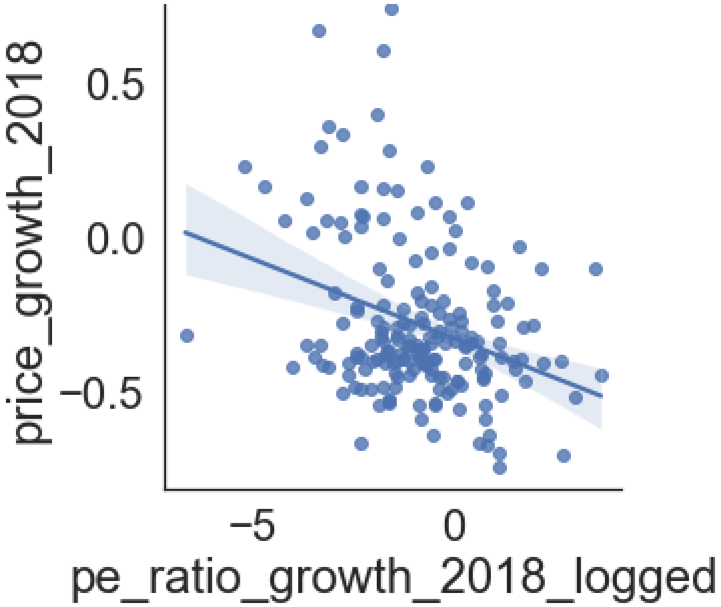
```
sns.lmplot(x="pe_ratio_growth_2015_logged", y="pric
sns.lmplot(x="pe_ratio_growth_2016_logged", y="pric
sns.lmplot(x="pe_ratio_growth_2017_logged", y="pric
sns.lmplot(x="pe_ratio_growth_2018_logged", y="pric
sns.lmplot(x="pe_ratio_growth_2019_logged", y="pric
sns.lmplot(x="pe_ratio_growth_2020_logged", y="pric
```

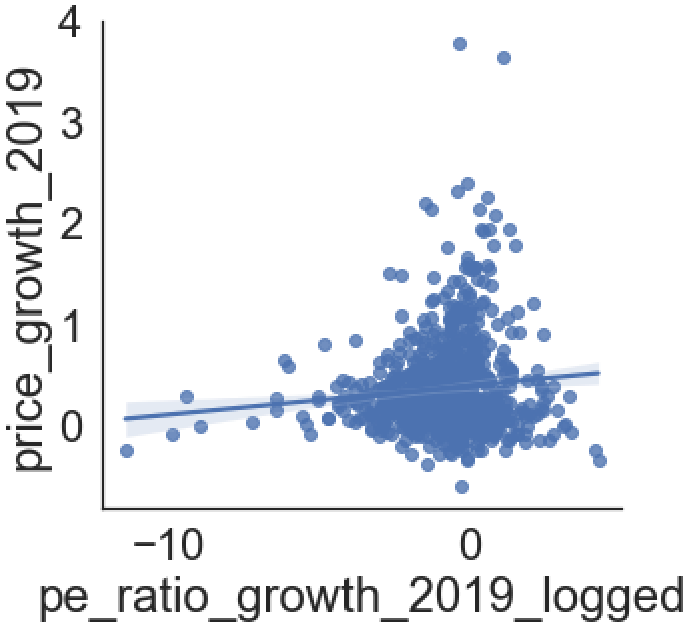
Out[46]:

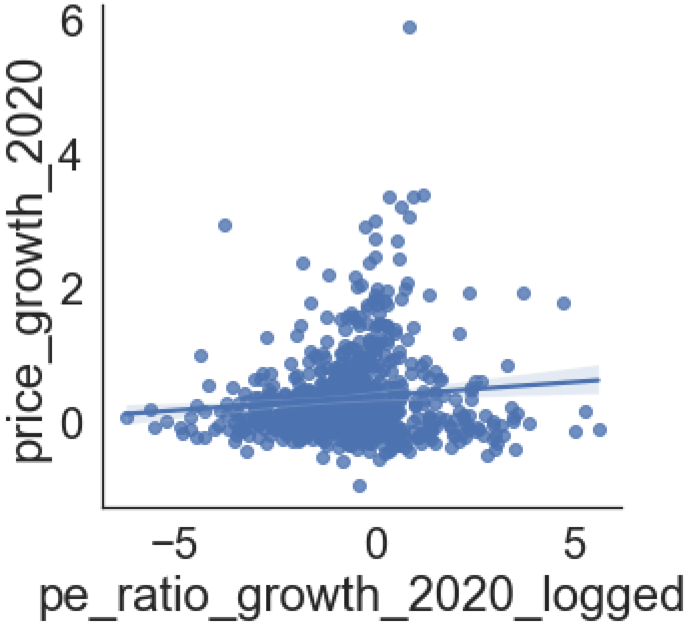

price_growth_2015
6
4
2
0
−5
0
5
pe_ratio_growth_2015_logged


price_growth_2016
2
1
0
−5
0
5
pe_ratio_growth_2016_logged


price_growth_2017
1
0
−5
0
pe_ratio_growth_2017_logged







We use the lmplot to find the relationship between

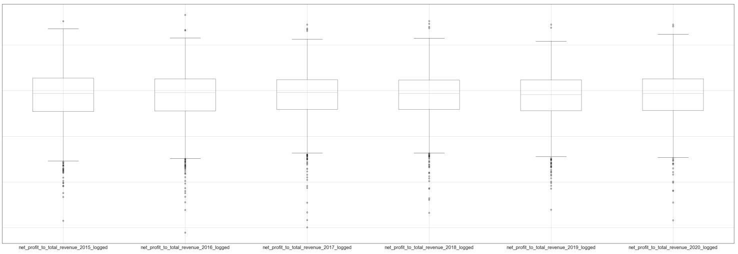
We use the lmplot to find the relationship between log(pe_ratio_growth) and price_growth. From the lmplot, we can see that the relationship between log(pe_ratio_growth) and price_growth in 2015, 2016, 2019, and 2020 are positive. And the relationship between log(pe_ratio_growth) and price_growth in 2017 and 2018 is negative. This result is a little different with the relationship between original pe_ratio_growth and price_growth. From the lmplot, we can more clearly see that all years' log pe_ratio_growth is in the range -5 to 5. The linear regression line is more fitted than pe_ratio_growth vs price_growth.

Price Growth vs. log(net_profit_to_total_revenue)

In [47]:

```
data_drop_outliers[["net_profit_to_total_revenue_20
```

Out[47]:

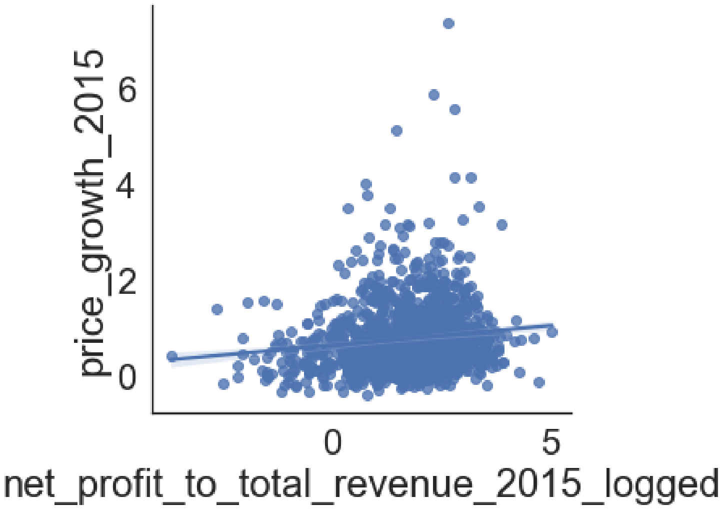


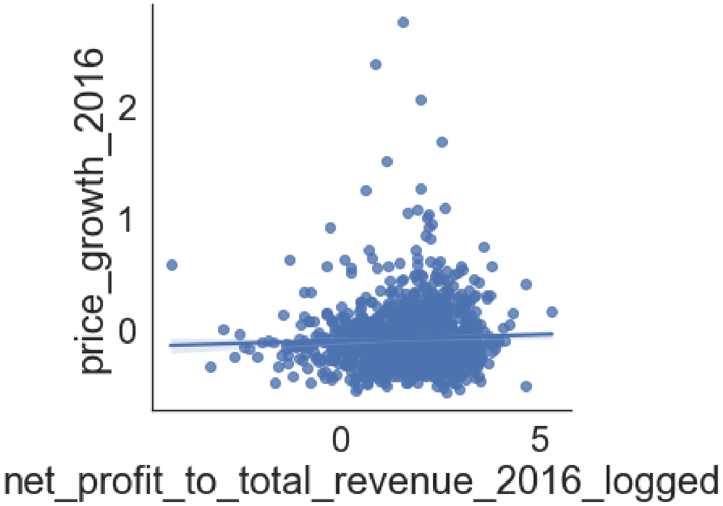
The boxplot clearly shows us the range of all log values of net_profit_to_total_revenue, which excludes outlier. According to the boxplot, most of the stocks see the log values of the net_profit_to_total_revenue are between 0 and 3. The distribution mostly centers at around 1.8 or 1.9, since the box plot does not contain the firms that net_profit_to_total_revenue is 0 have a NaN values for log. The extreme values of the net_profit_to_total_revenue are up to -4 and 4. Since we are using the data after we drop the outliers and excludes the outlier data, the outlier data shown here are only the outlier for this set of data, instead of the raw data we are using. Therefore, we will not drop the outlier data here.

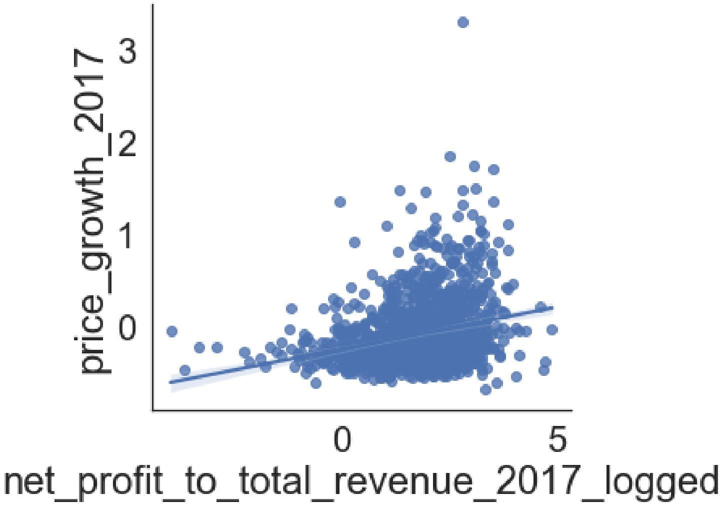
In [48]:

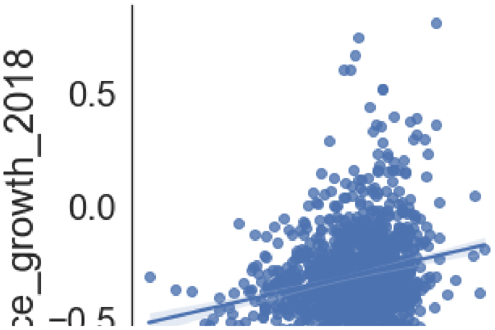
```
sns.lmplot(x="net_profit_to_total_revenue_2015_logg
sns.lmplot(x="net_profit_to_total_revenue_2016_logg
sns.lmplot(x="net_profit_to_total_revenue_2017_logg
sns.lmplot(x="net_profit_to_total_revenue_2018_logg
sns.lmplot(x="net_profit_to_total_revenue_2019_logg
sns.lmplot(x="net_profit_to_total_revenue_2020_logg
```

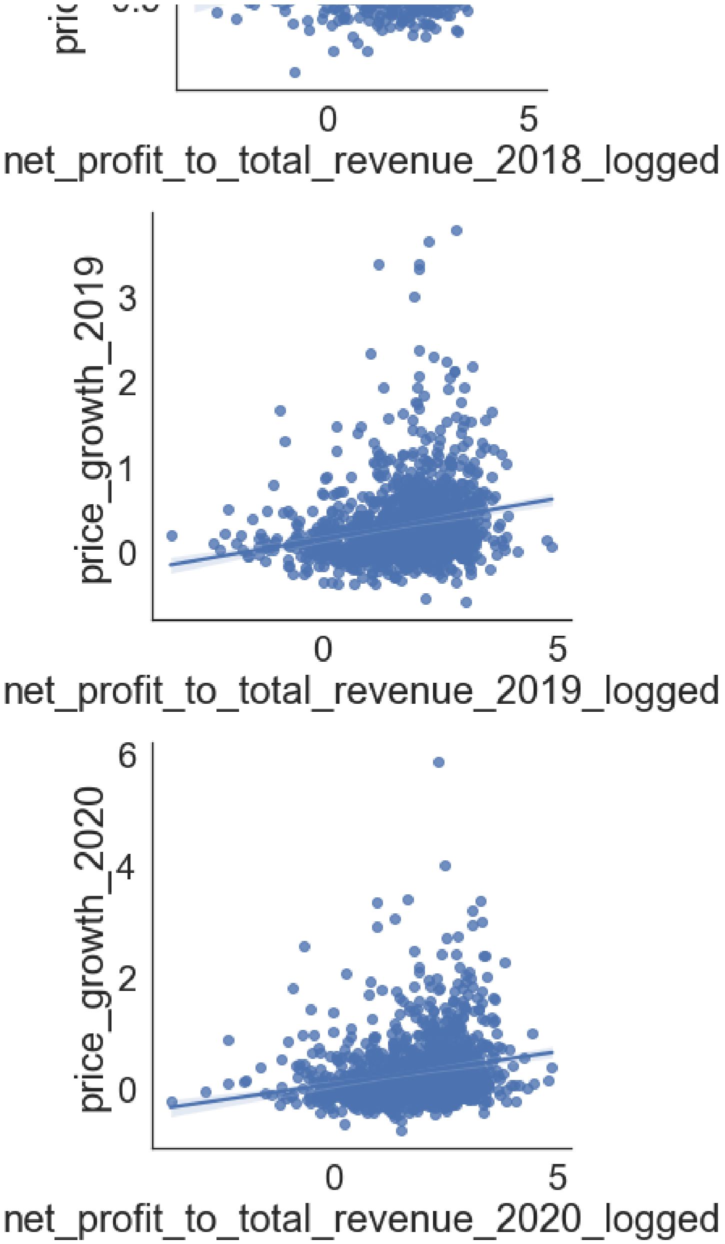
Out[48]:


price_growth_2015
0
2
4
6
0
5
net_profit_to_total_revenue_2015_logged


price_growth_2016
0
1
2
0
5
net_profit_to_total_revenue_2016_logged


price_growth_2017
0
1
2
3
0
5
net_profit_to_total_revenue_2017_logged


ce_growth_2018
−0.5
0.0
0.5



From the implot for the log scale of the net_profit_to_total_revenue, we can still see overall slight positive relationship between net_profit_to_total_revenue and the changes in stock price. Most of the stocks have net_profit_to_total_revenue centers around 2 or 3, which means that they are earning economic profit and doing quite well. Since we do not include the firms that net_profit_to_total_revenue is 0, most firms are at the right side of 0. The upward gradient is clear, and the shadow blue area is clearly smaller than the implot before without log scale, which means that the positive correlation is more evident. This trend is reasonable, since the higher ratio means that the firms that have higher profit margin will have higher stock price on that year, and therefore more people buy their stock. Consequently, the stock prices raise. Using the log scale

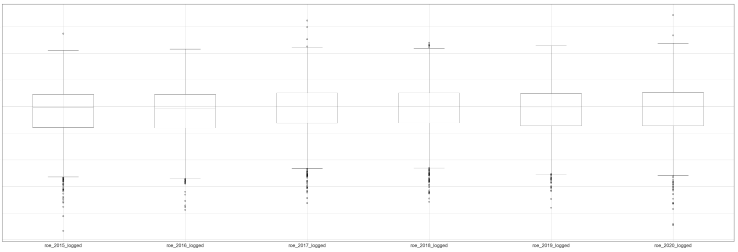
Consequently, the stock prices raise. Using the log scale, we can see the correlation clearer and better arrive at our conclusion.

Price Growth vs. log(roe)

In [49]:

```
data_drop_outliers[["roe_2015_logged", "roe_2016_log
```

Out[49]:

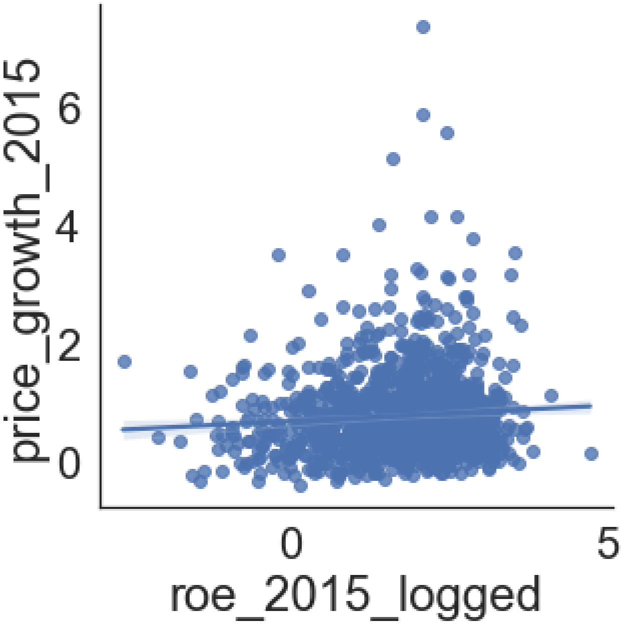


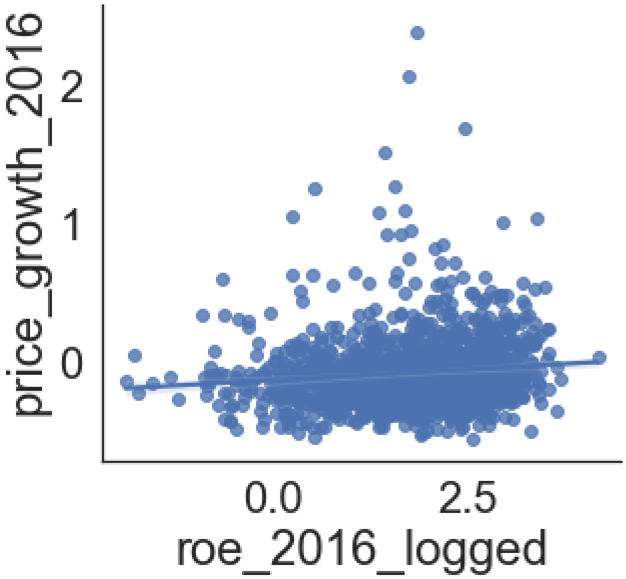
This boxplot shows us the outlier data for the roe-logged distribution. According to the boxplot, most of the stocks have a roe-logged value between -1 to 4, centered around 2. The distribution looks less spread than before the transformation. The shape of the distribution are similar and normal for each year for the roe.

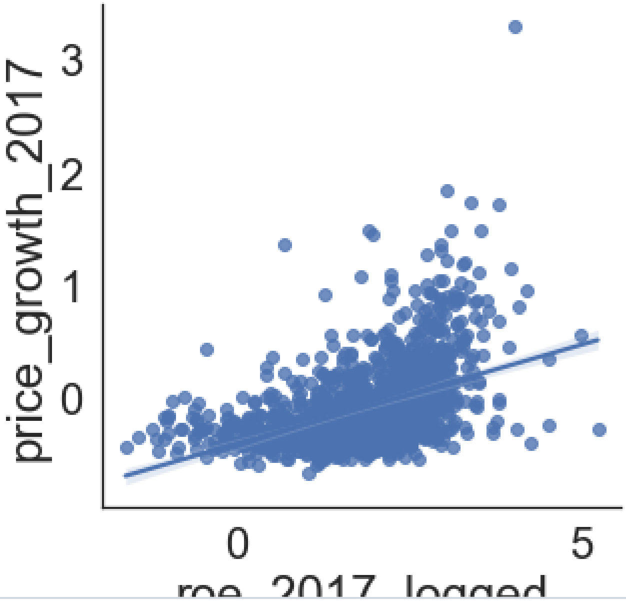
In [50]:

```
sns.lmplot(x="roe_2015_logged", y="price_growth_201
sns.lmplot(x="roe_2016_logged", y="price_growth_201
sns.lmplot(x="roe_2017_logged", y="price_growth_201
sns.lmplot(x="roe_2018_logged", y="price_growth_201
sns.lmplot(x="roe_2019_logged", y="price_growth_201
sns.lmplot(x="roe_2020_logged", y="price_growth_202
```

Out[50]:




price_growth_2016
2
1
0
0.0
2.5
roe_2016_logged


price_growth_2017
3
2
1
0
0
5
roe_2017_logged