README.md

Stock Price Analysis | Earnings Announcements

Background

The purpose of this analysis is to examine how earnings announcements affect the behavior of future stock prices.

Authors

- Axander Wilson

Files and Folders

`./data_exploration` - contains Jupyter Notebooks. Use for data Exploration, visuzulations, model training

`./data` - contains dataset used for analysis

Dependencies

- numpy
- pandas
- sklearn
- yahoo-finance

Analysis Question

Does movement on the day after earnings predict movement in the following periods (day, week, month)?

Dependent Variable:

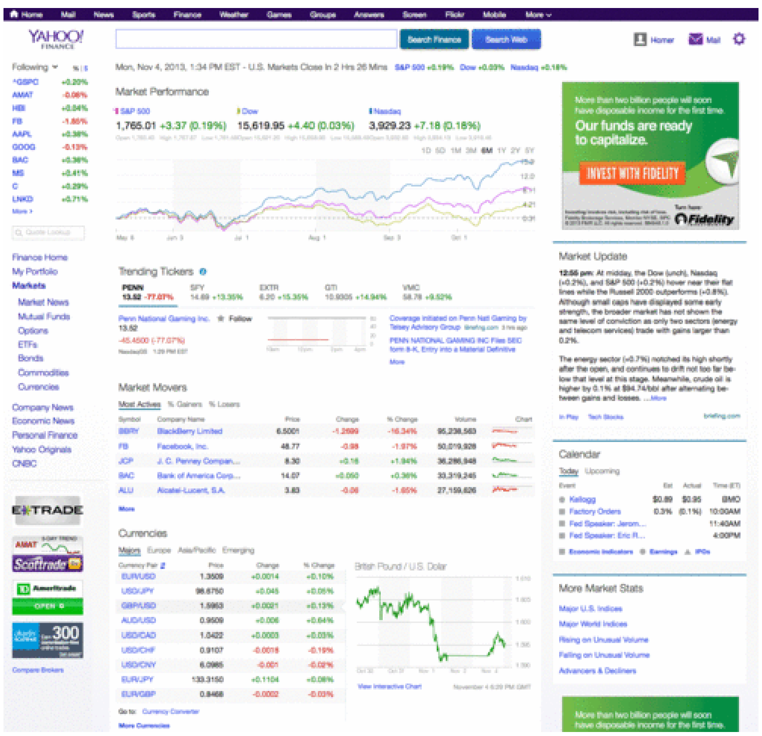
Price change in close price of one day after earnings announcement and close price of two days after earnings announcement date

Independent Variables:

1. Price change in the week preceding the announcement date (announcement day close price - week preceding close price)
2. Price change on announcement day. (day proceeding announcement day close price - next trade day after announcement day open price)
3. Price change on next trade after announcement day (next trade day after announcement day open price - next trade day after announcement day close price)

Data Source and Data Collection:

Yahoo Finance earnings calendar API was used to collect earnings date data. Yahoo Finance stock price API was used to collect history stock price information.



Testing categorization

The companies were broken up into four categories:

Category 1:

Increase in preceding week, increase in first trading day after announcement

Category 2:

Increase in preceding week, decrease in first trading day after announcement

Category 3:

Decrease in preceding week, increase in first trading day after announcement

Category 4:

Decrease in preceding week, decrease in first trading day after announcement

Criteria:

In order to be considered in the analysis, a company must fall into category 1-4 and its change in the trade day after announcement day must be greater than or equal to two percent.

Results

A total of 234 companies met the criteria for analysis.

View dataset

In [21]:
```
full_df.head(3)
```

Out[21]:

	ticker	company	earnings_flag	earnings_ann_date	ann_price_open	ann_price_close	week preceeding date	week_price_open	week_price_close	pct_chng_t-7	weel
0	AAMC	Altisource Asset Management Corp	Before Market Open	2020-11-05	22.98	22.98000	2020-10-28	22.290001	23.000000	-0.000870	
1	AES	AES Corp	Before Market Open	2020-11-06	20.91	20.41000	2020-10-29	19.320000	19.530001	0.043116	
2	ALE	ALLETE Inc	Before Market Open	2020-11-09	55.66	57.22398	2020-10-30	51.799999	51.004200	0.108692	

A Linear Regression Model was trained and tested on the dataset.

Category 1:

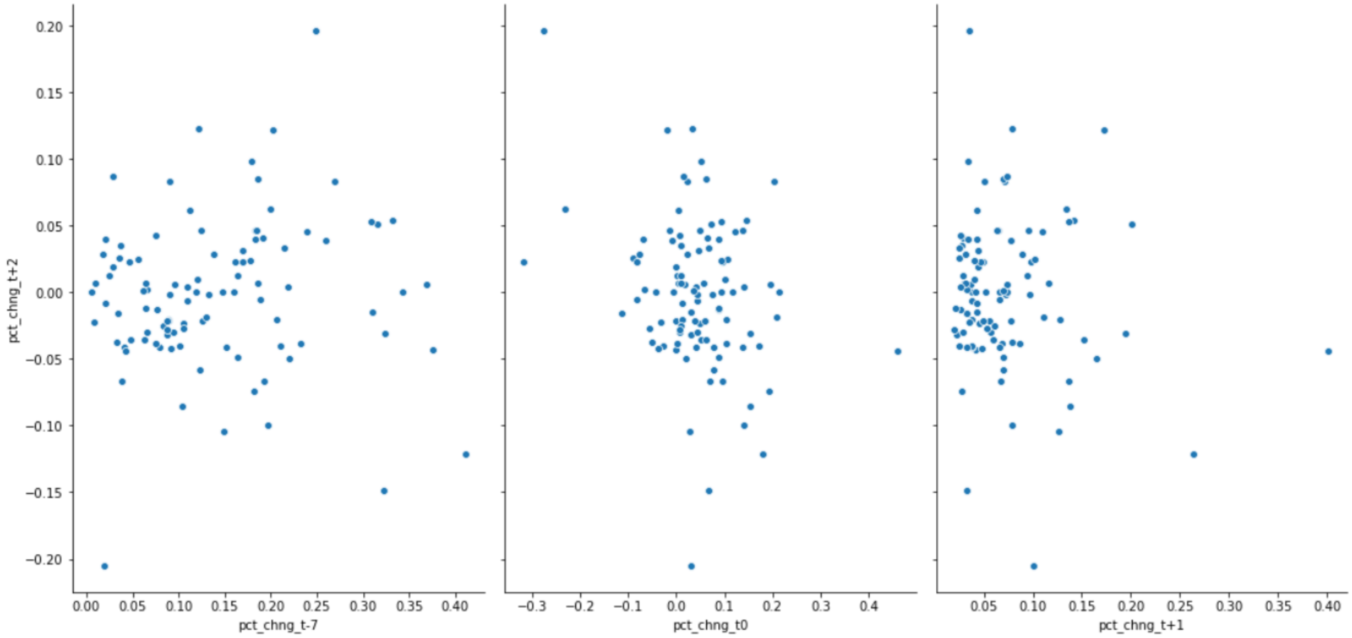
100 of the 234 data points fell into category 1.

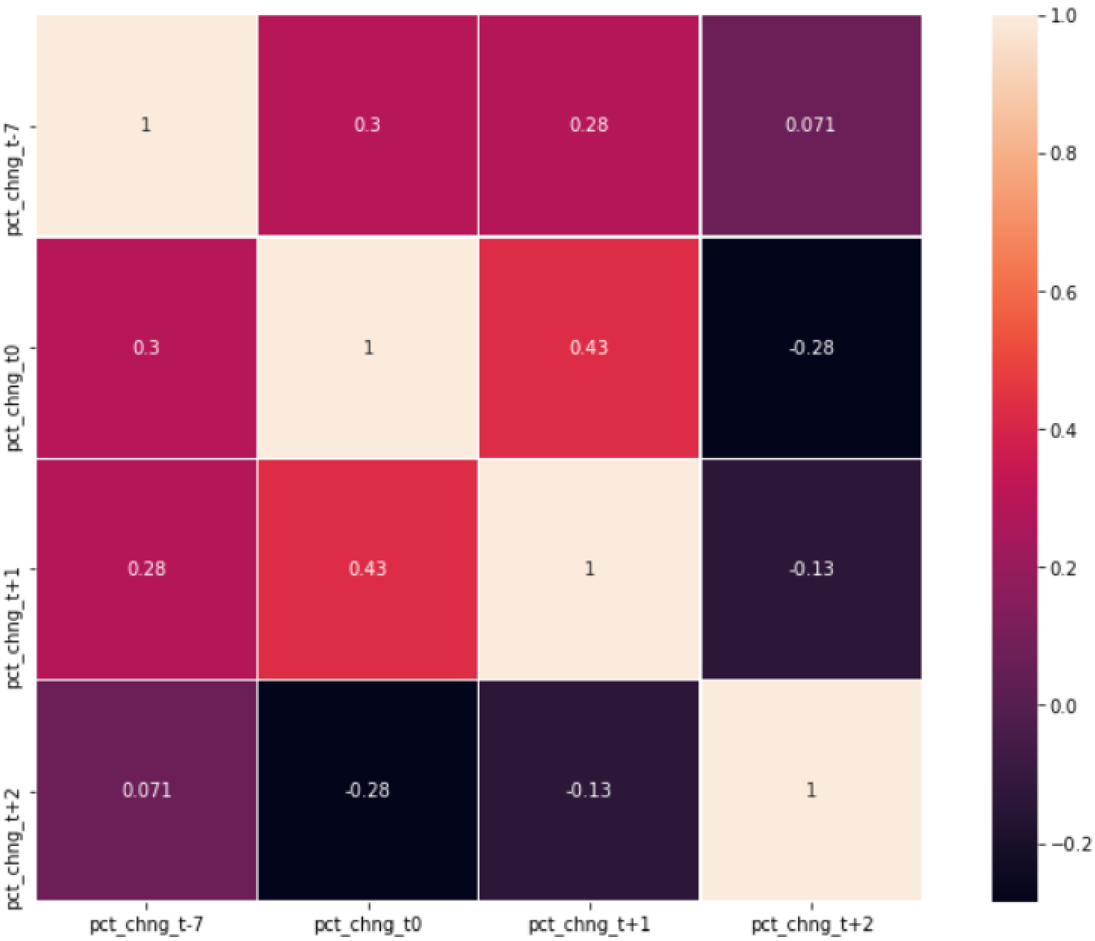
None of the independent variables showed a strong correlation with the response variable.

The strongest correlation variable was the price change on announcement day (-0.28).

Visualizations:

Out[38]: <seaborn.axisgrid.PairGrid at 0x26dc63c8730>





In [39]:
```
# Compute the correlation matrix
corr = cat_1_df.corr()
corr
```

Out[39]:

	pct_chng_t-7	pct_chng_t0	pct_chng_t+1	pct_chng_t+2
pct_chng_t-7	1.000000	0.297020	0.283093	0.070657
pct_chng_t0	0.297020	1.000000	0.432319	-0.284620
pct_chng_t+1	0.283093	0.432319	1.000000	-0.129817
pct_chng_t+2	0.070657	-0.284620	-0.129817	1.000000

Coefficients:

View Model coefficients

In [47]:
```
for idx, col_name in enumerate(X_train.columns):
    print("The coefficient for {} is {}".format(col_name, regression_model.coef_[0][idx]))
```

```
The coefficient for pct_chng_t-7 is 0.0373419572150715
The coefficient for pct_chng_t0 is -0.18610297949516952
The coefficient for pct_chng_t+1 is 0.04557248164814025
```

Intercept:

View Model Intercept

In [48]:
```
intercept = regression_model.intercept_[0]
```

In [49]:
```
print("The intercept for our model is {}".format(intercept))
```

```
The intercept for our model is 0.0009502620742763319
```

Category 1 Conclusion:

The model yielded a negative r-squared score of -0.003. Note that it is possible to get a negative R-square for equations that do not contain a constant term. Because R-square is defined as the proportion of variance explained by the fit, if the fit is worse than just fitting a horizontal line then R-square is negative. In this case, R-square cannot be interpreted as the square of a correlation. Such situations indicate that a constant term should be added to the model. The mean square error was 0.058. So, the model was an average of 0.06 percent change away from the ground truth percentage when making predictions on our test set.

Category 2:

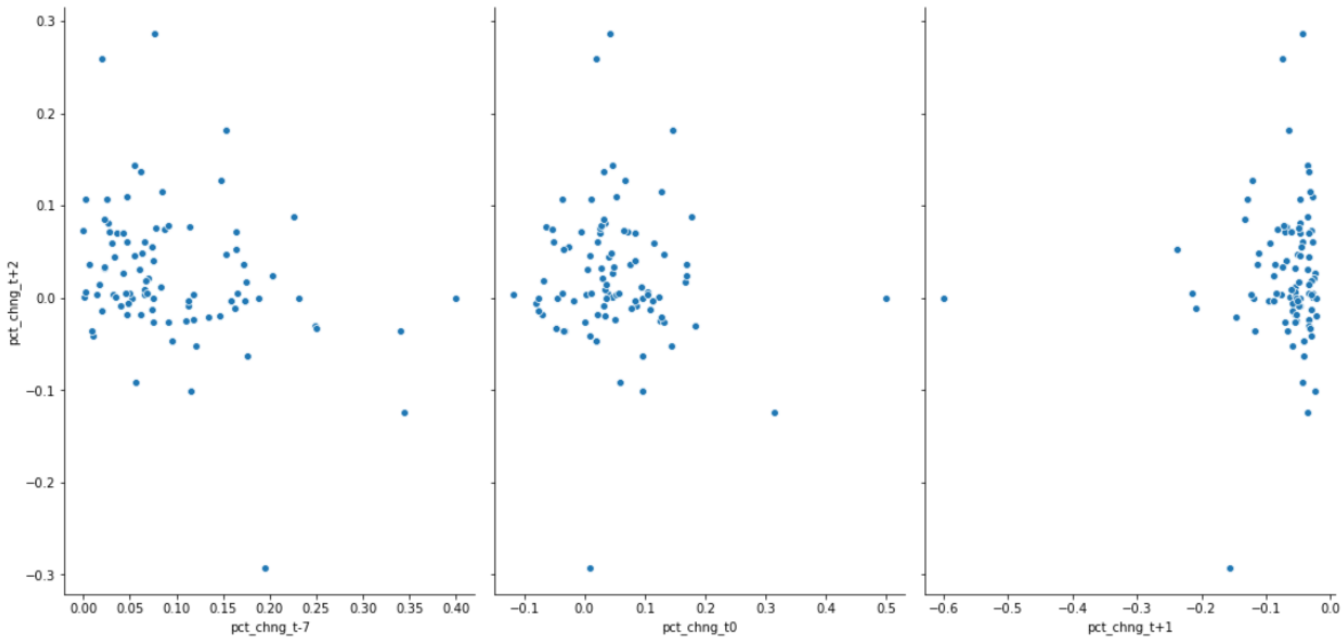
90 of the 234 data points fell into category 2.

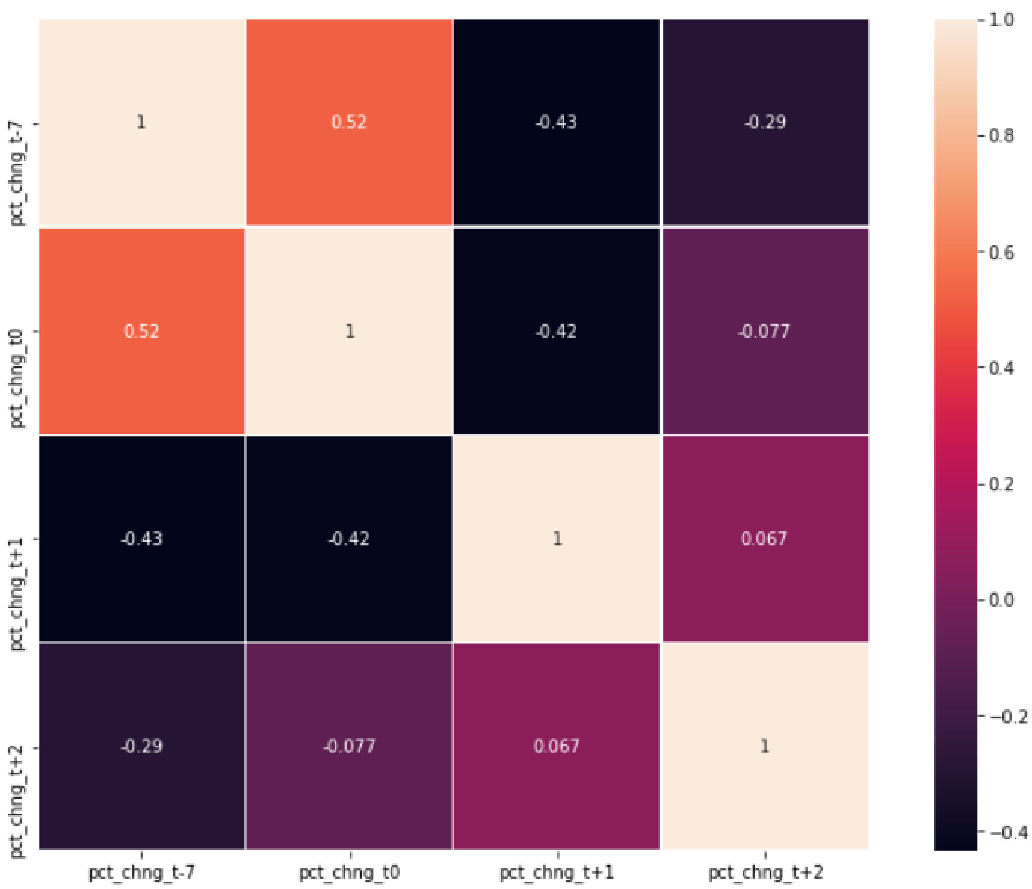
None of the independent variables showed a strong correlation with the response variable.

The strongest correlation variable was the price change in the week preceding the announcement date (-0.29).

Visualizations:

```
Out[25]: <seaborn.axisgrid.PairGrid at 0x27cc66e5280>
```





In [26]:
```
# Compute the correlation matrix
corr = cat_2_df.corr()
corr
```

Out[26]:

	pct_chng_t-7	pct_chng_t0	pct_chng_t+1	pct_chng_t+2
pct_chng_t-7	1.000000	0.523553	-0.433784	-0.292146
pct_chng_t0	0.523553	1.000000	-0.421381	-0.076940
pct_chng_t+1	-0.433784	-0.421381	1.000000	0.066683
pct_chng_t+2	-0.292146	-0.076940	0.066683	1.000000

Coefficients:

View Model coefficients

In [34]:
```
for idx, col_name in enumerate(X_train.columns):
    print("The coefficient for {} is {}".format(col_name, regression_model.coef_[0][idx]))
```

```
The coefficient for pct_chng_t-7 is -0.3293531056665659
The coefficient for pct_chng_t0 is 0.08970603564094382
The coefficient for pct_chng_t+1 is -0.05170379963963407
```

Intercept:

View Model Intercept

In [35]:
```
intercept = regression_model.intercept_[0]
```

In [36]:
```
print("The intercept for our model is {}".format(intercept))
```

```
The intercept for our model is 0.04787061113998166
```

Category 2 Conclusion:

The model yielded an r-squared score of 0.15739. So in our model, 15.74% of the variability in Y can be explained using X. The mean square error was 0.06074. So, the model was an average of 0.06 percent change away from the ground truth percentage when making predictions on our test set.

Category 3:

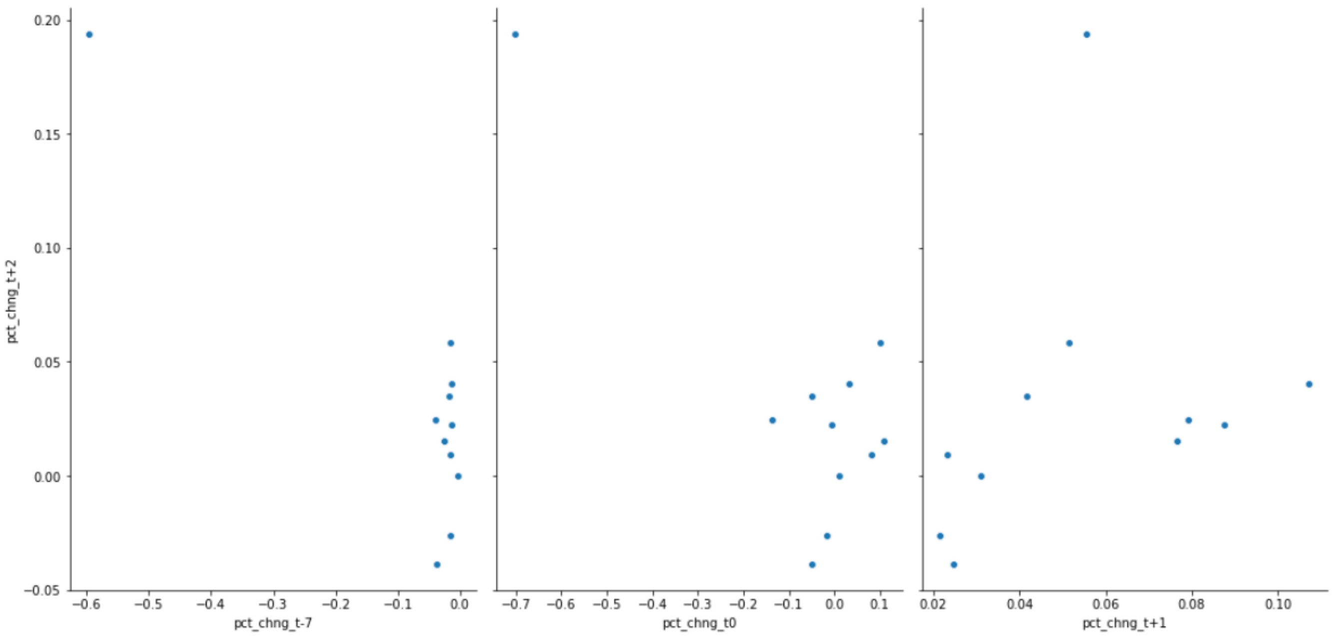
11 of the 234 data points fell into category 3.

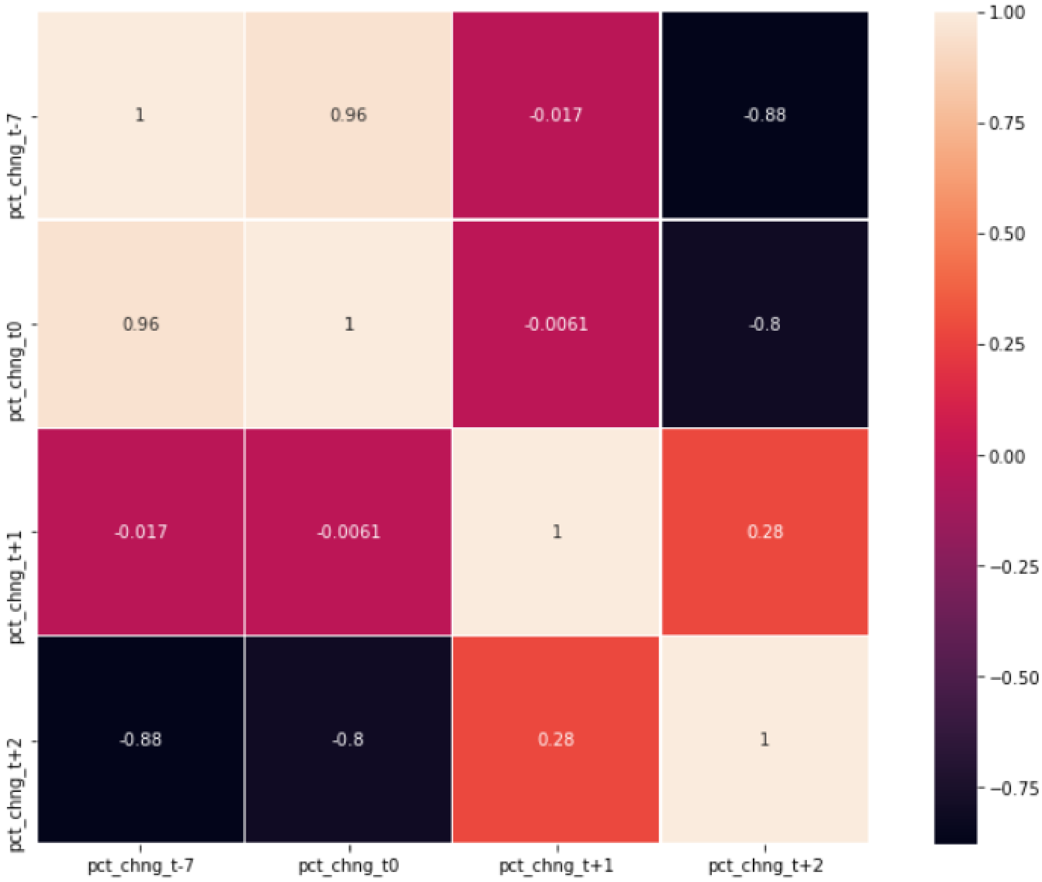
Price change in the week preceding the announcement date and Price change on announcement day independent variables showed a strong negative correlation with the response variable.

The strongest correlation variable was the price change in the week preceding the announcement date (-0.88).

Visualizations:

Out[26]: <seaborn.axisgrid.PairGrid at 0x1f5bcdc1040>





In [27]:
```
# Compute the correlation matrix
corr = cat_3_df.corr()
corr
```

Out[27]:

	pct_chng_t-7	pct_chng_t0	pct_chng_t+1	pct_chng_t+2
pct_chng_t-7	1.000000	0.955389	-0.016655	-0.879264
pct_chng_t0	0.955389	1.000000	-0.006079	-0.799415
pct_chng_t+1	-0.016655	-0.006079	1.000000	0.283144
pct_chng_t+2	-0.879264	-0.799415	0.283144	1.000000

Coefficients:

View Model coefficients

In [35]:
```
for idx, col_name in enumerate(X_train.columns):
    print("The coefficient for {} is {}".format(col_name, regression_model.coef_[0][idx]))
```

```
The coefficient for pct_chng_t-7 is -0.4714419614490647
The coefficient for pct_chng_t0 is 0.2779243484752656
The coefficient for pct_chng_t+1 is 0.7295297525841625
```

Intercept:

View Model Intercept

In [36]:
```
intercept = regression_model.intercept_[0]
```

In [37]:
```
print("The intercept for our model is {}".format(intercept))
```

```
The intercept for our model is -0.037216837027316194
```

Category 3 Conclusion:

The model yielded an r-squared score of 0.20497. So in our model, 20.50% of the variability in Y can be explained using X. The mean square error was 0.07126. So, the model was an average of 0.07 percent change away from the ground truth percentage when making predictions on our test set.

Category 4:

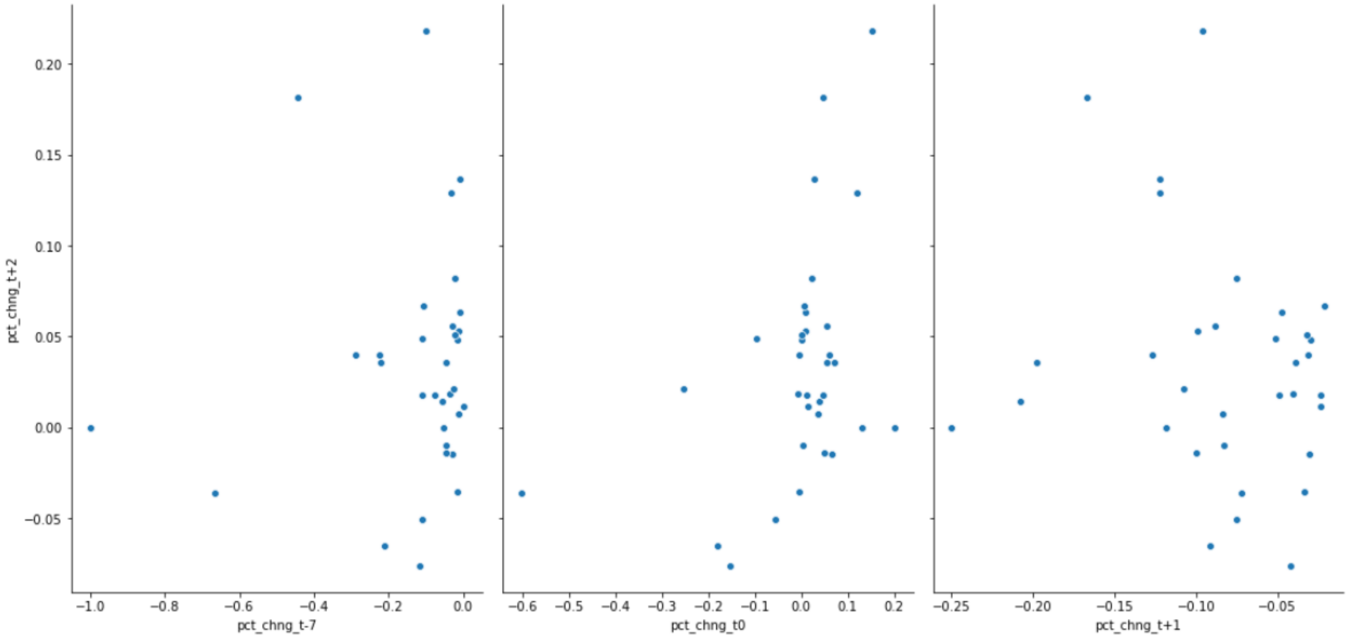
33 of the 234 data points fell into category 4.

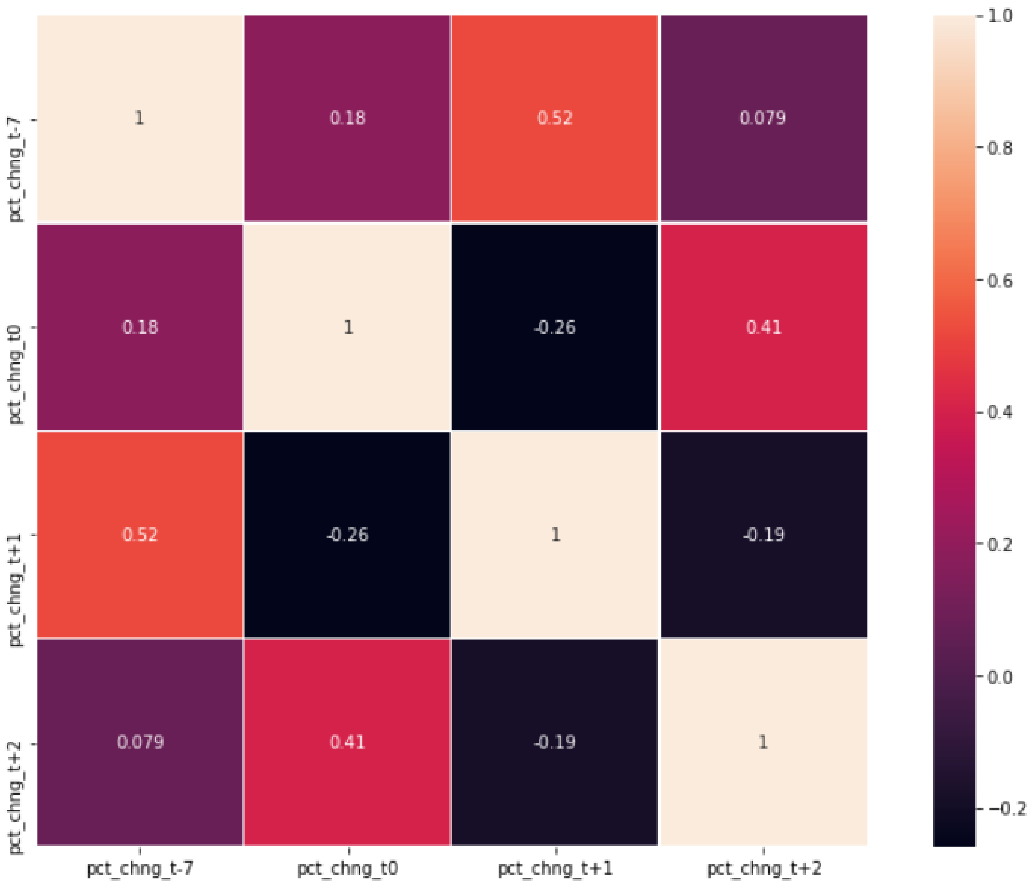
None of the independent variables showed a strong correlation with the response variable.

The strongest correlation variable was the price change on announcement day (0.41).

Visualizations:

```
Out[26]: <seaborn.axisgrid.PairGrid at 0x2b2b784b2e0>
```





In [27]:
```
# Compute the correlation matrix
corr = cat_4_df.corr()
corr
```

Out[27]:

	pct_chng_t-7	pct_chng_t0	pct_chng_t+1	pct_chng_t+2
pct_chng_t-7	1.000000	0.180648	0.522073	0.079451
pct_chng_t0	0.180648	1.000000	-0.259395	0.406892
pct_chng_t+1	0.522073	-0.259395	1.000000	-0.185269
pct_chng_t+2	0.079451	0.406892	-0.185269	1.000000

Coefficients:

View Model coefficients

In [35]:
```
for idx, col_name in enumerate(X_train.columns):
    print("The coefficient for {} is {}".format(col_name, regression_model.coef_[0][idx]))
```

```
The coefficient for pct_chng_t-7 is 0.026823552328484563
The coefficient for pct_chng_t0 is 0.10057009972436581
The coefficient for pct_chng_t+1 is -0.09917448179492437
```

Intercept:

View Model Intercept

In [36]:
```
intercept = regression_model.intercept_[0]
```

In [37]:
```
print("The intercept for our model is {}".format(intercept))
```

```
The intercept for our model is 0.029908904828306177
```

Category 4 Conclusion:

The model yielded an r-squared score of 0.19152. So in our model, 19.15% of the variability in Y can be explained using X. The mean square error was 0.07302. So, the model was an average of 0.07 percent change away from the ground truth percentage when making predictions on our test set.

Future Analysis:

We were unable to observe an earnings announcements affect on future stock prices in our dataset and model. A more robust dataset could produce different results, as this dataset was limited.

Additionally, it would be interesting to observe the daily and weekly changes and compare it to the daily change on announcement day and each day after the announcement day for the next month. And then compare these changes to the average daily, weekly change.

For example, find ALE daily, weekly, monthly price changes from 01/01/2020 - 10/31/2020. Take the avg so that there is an avg daily change, avg weekly, avg monthly change. Then compare these changes to pct changes on announcement day and after. And observe if the avg change comparison is significant (e.g. if the change is >=2%).