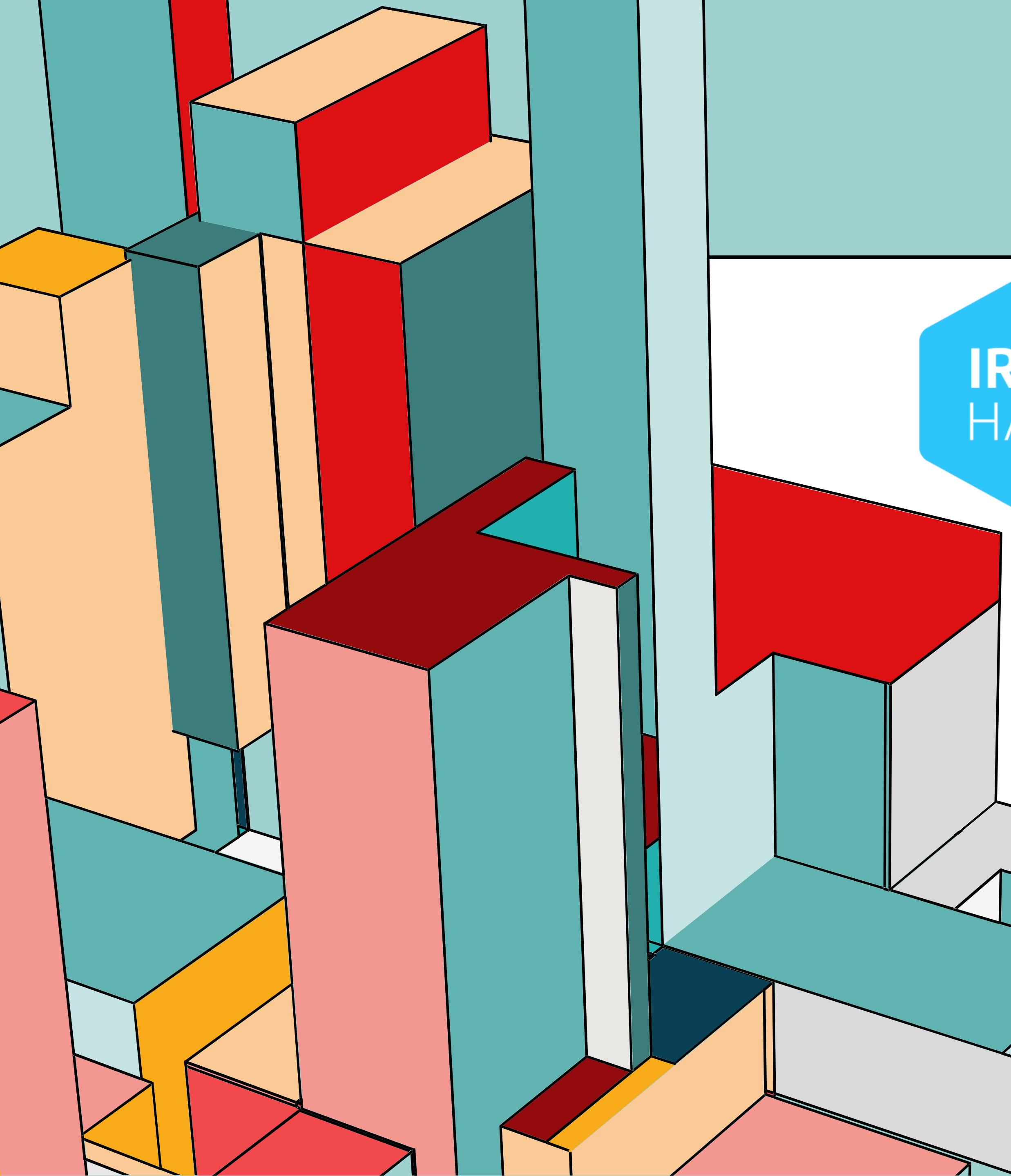


IRON HACK

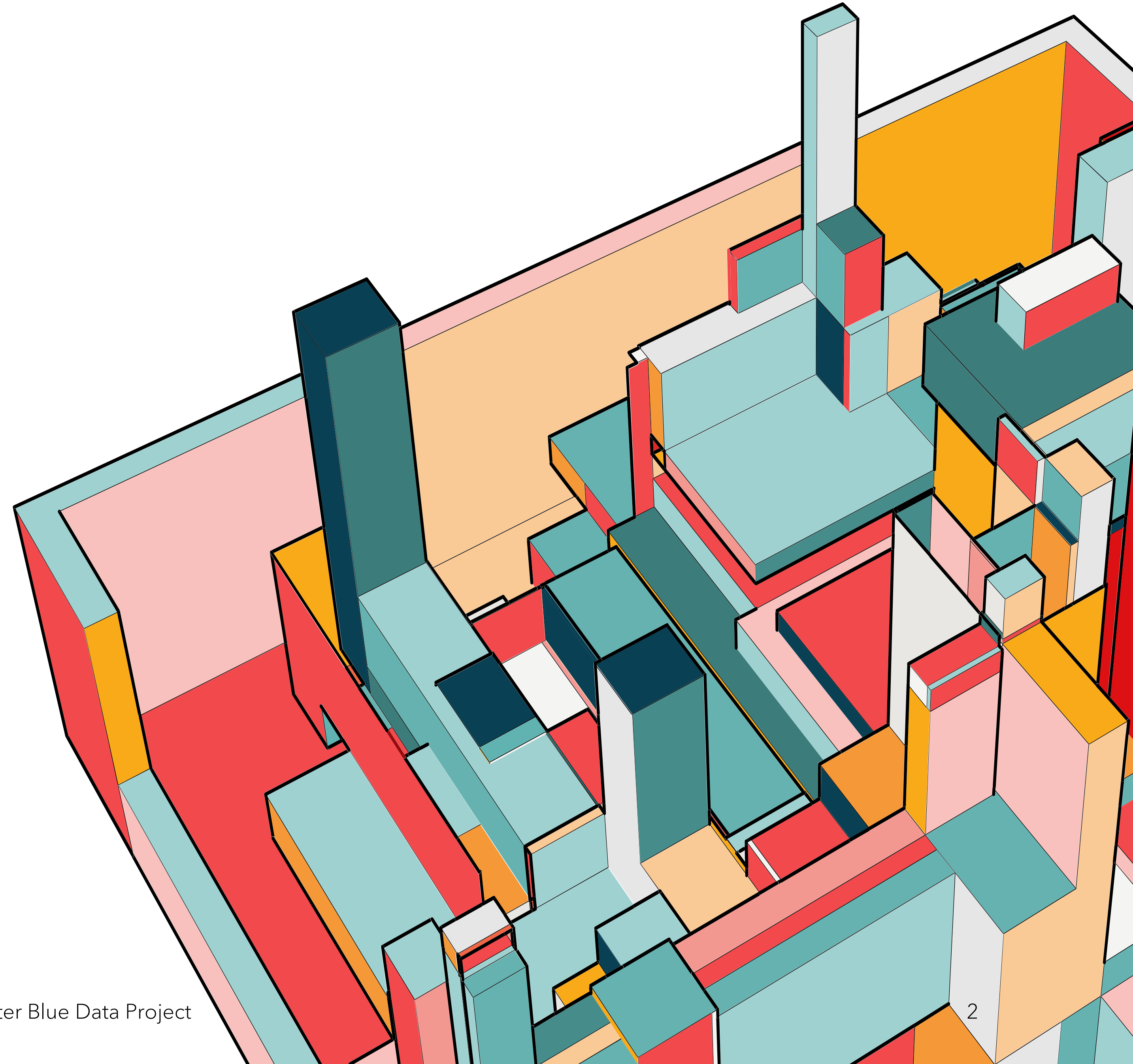
FINAL PROJECT

Twitter Blue's Impact on Stocks

By Evan Strait

PROJECT PROPOSAL

Last month, a media frenzy followed the abrupt rollout of a new Twitter Blue feature: the ability to get a verified account by subscribing for $8 a month. According to the NYT, “The social media service, which is undergoing changes from its new owner Elon Musk, has descended into a messy swirl of spoof messages and parody accounts.” This project analyzes its impacts.



PROBLEM

BRAND REPUTATION

Companies were often targeted based on public weak points: like their industry (i.e., weapons manufacturers) or the leadership of the company (companies run by Musk)

TWITTER TROLLS

Anyone willing to “commit $8 to the bit” was instantly given access to major brand awareness – they were only a few dozen retweets away from virality

STOCK PRICES

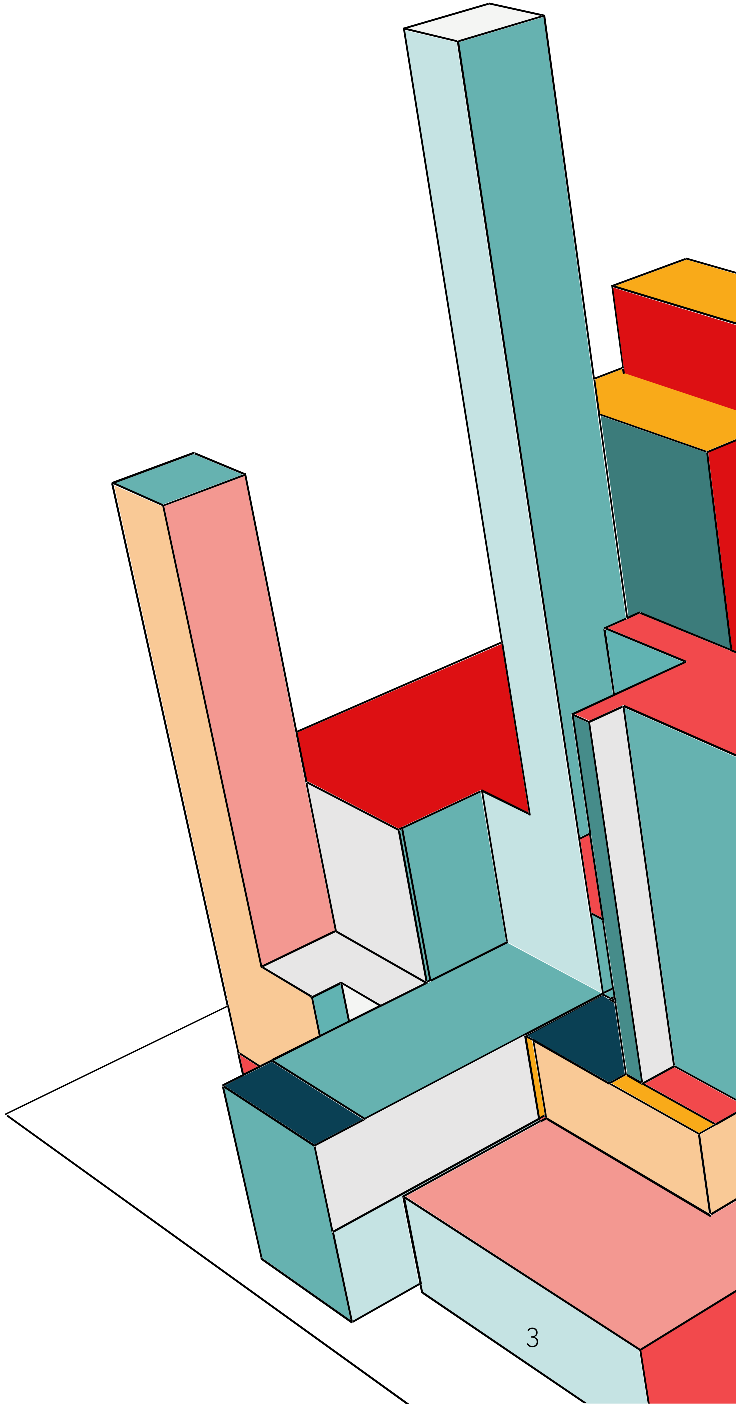
Entrepreneur.com posted the headline “Eli Lilly Stock Plummets After Parody Twitter Account Says Insulin is Now Free”

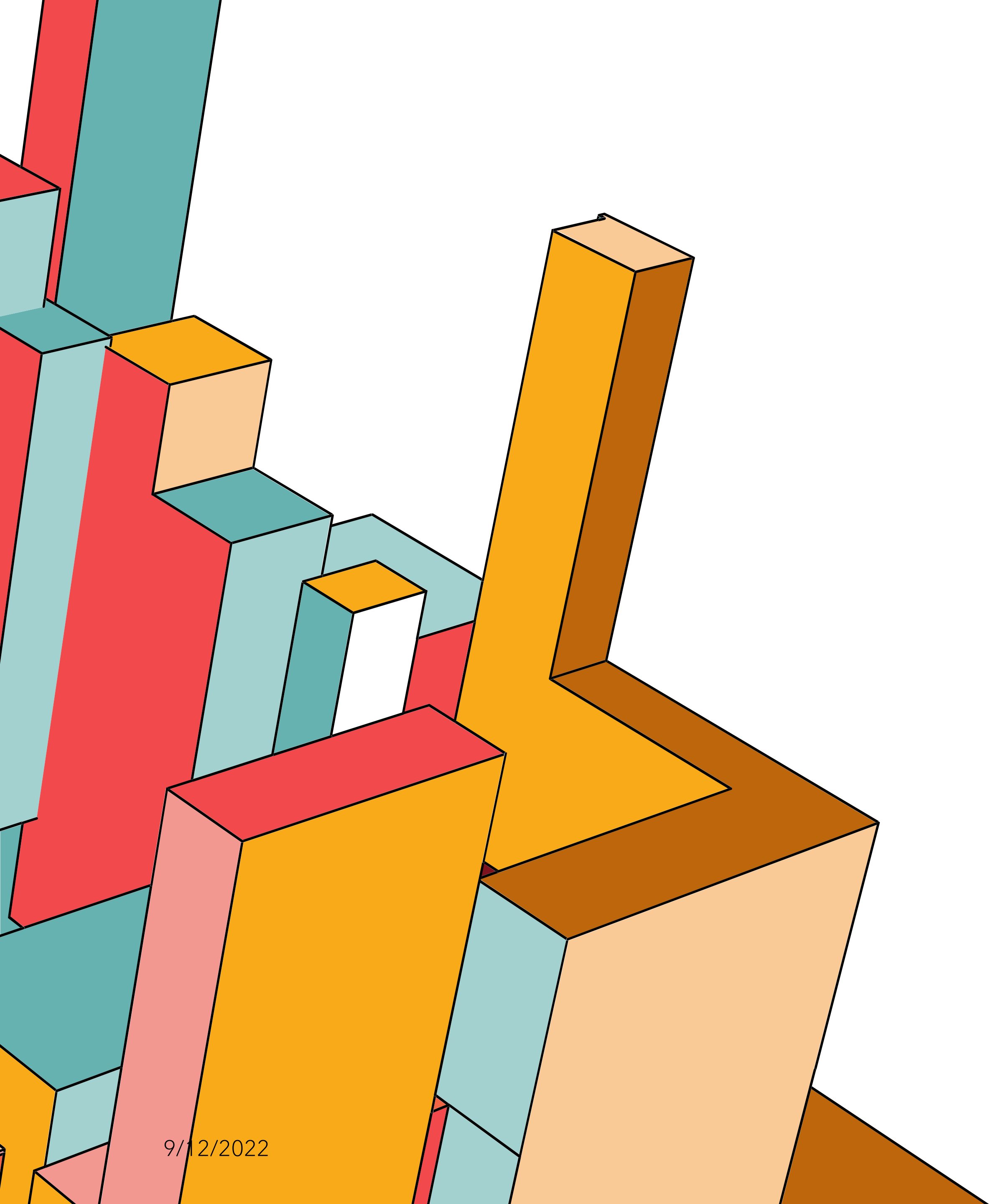
AUDIENCE ENGAGEMENT

Despite accounts being banned and tweets being taken down, screenshots were posted across the Twitter and conversations continue to this day

TWITTER MANAGEMENT

Accounts could post multiple fake tweets – often up for hours before being removed





USE CASE

THE GOAL

This project will spotlight how parody accounts were able to present themselves as reputable companies enough to significantly affect stock price valuation. Companies' responses will be demonstrated, along with long-term effects of the parody Tweets on the company's stock.

THE PLAN

Taking data from the Twitter API for Developers and comparing it with Yahoo Finance information from the same time period, I can look at the impacts seen in the hours and days following parody tweets been posted and made viral. From there, I can build prediction models and visualizations to better communicate the effects.

TWEET EXAMPLES

TWEET EXAMPLES

TWEET EXAMPLES

TWEET EXAMPLES

THE DATA

Final Project.ipynb
File Edit View Insert Runtime Tools Help Last saved at 11:04 AM

```
[ ] hist
```

Date	Open	High	Low	Close	Volume	Dividends	Stock Splits
1977-01-03 00:00:00-05:00	0.332907	0.337406	0.332907	0.337406	165282	0.0	0.0
1977-01-04 00:00:00-05:00	0.337406	0.341904	0.319411	0.337406	179952	0.0	0.0
1977-01-05 00:00:00-05:00	0.337406	0.346403	0.337406	0.346403	189732	0.0	0.0
1977-01-06 00:00:00-05:00	0.346403	0.364398	0.346403	0.364398	384354	0.0	0.0
1977-01-07 00:00:00-05:00	0.364398	0.368897	0.359899	0.368897	345234	0.0	0.0
...	...	...	...	...	...	...	...
2022-12-01 00:00:00-05:00	486.519989	488.000000	480.670013	483.690002	913600	0.0	0.0
2022-12-02 00:00:00-05:00	481.809998	498.950012	481.809998	496.230011	1744400	0.0	0.0
2022-12-05 00:00:00-05:00	490.869995	494.980011	488.320007	490.670013	1357200	0.0	0.0
2022-12-06 00:00:00-05:00	491.320007	491.320007	482.200012	484.000000	1076500	0.0	0.0
2022-12-07 00:00:00-05:00	484.299988	489.239990	482.679993	484.940002	463550	0.0	0.0

11584 rows × 7 columns

```
[ ] stock = yf.Ticker('LMT')
data = stock.history(start='2022-11-9', end='2022-11-16', interval = '60m')

# Build a list of stock values
lmt = [info for date, info in data.iterrows()]

# Print the list
print(lmt)
```

```
[Open              489.750000
High              492.440002
Low               487.989990
Close             488.989990
Volume         266824.000000
Dividends           0.000000
Stock Splits        0.000000
Name: 2022-11-09 09:30:00-05:00, dtype: float64, Open              489.070007
High              489.309998
Low               484.239990
Close             487.959991
Volume         210827.000000
Dividends           0.000000
Stock Splits        0.000000
Name: 2022-11-09 10:30:00-05:00, dtype: float64, Open              487.980011
```

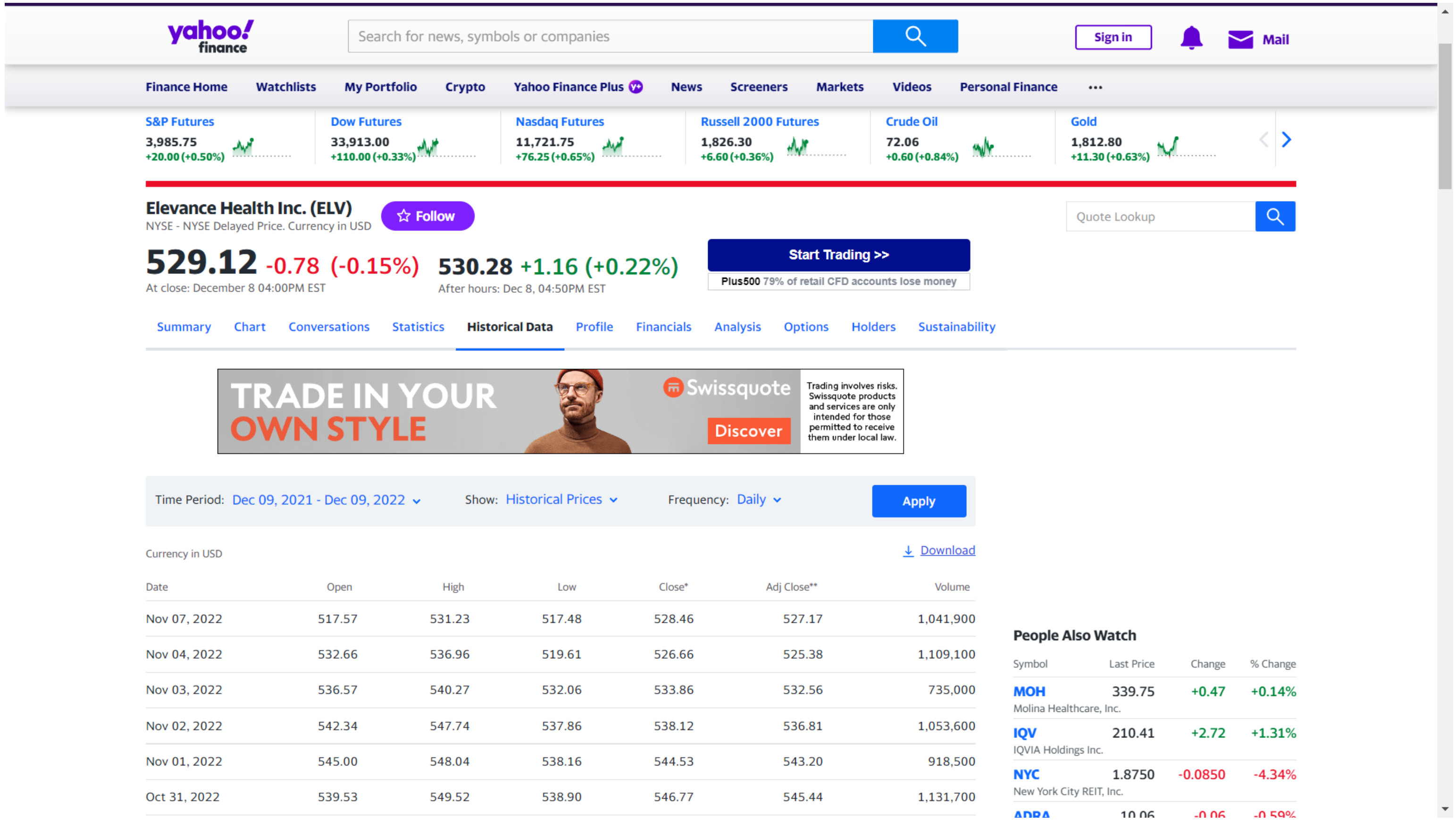
✓ 0s completed at 7:41 PM

YFINANCE STOCK DATA

Downloading hourly stock movements of the stock companies required use of the yfinance library in Python, which uses Yahoo Finance's API to allow for detailed queries.

I pulled hourly data of each stock for a week around the Twitter Blue incident, and a year of daily statistics for each company, prior to said week.

DOWNLOADING REFERENCE DATA



COMPANIES CHOSEN FOR ANALYSIS

GSK GlaxoSmithKilne	**SOJA3.SA** Boa Safra Sementes	**LLY** Eli Lilly and Co
NTDOY Nintendo (Game Freak)	**KO** Coca Cola	**LMT** Lockheed Martin Corp
ELV Blue Cross Blue Shield	**TSLA** Tesla	**BP** BP

THE DATA

```python
import tweepy
```

```python
[ ] user = api.get_user("Tesla")

    print(f"Follower count: {user.followers_count}")
    print(f"Following count: {user.favourites_count}")
    print(f"Statuses count: {user.statuses_count}")
```

```
Follower count: 18537709
Following count: 33
Statuses count: 7927
```

```python
[ ] user
```


```
User(_api=<tweepy.api.API object at 0x7f08e7cdb5e0>, _json={'id': 13298072, 'id_str': '13298072', 'name': 'Tesla', 'screen_name': 'Tesla', 'location': '', 'profile_location': None, 'description': 'electric vehicles, giant batteries
and solar', 'url': 'https://t.co/EJC6qVVMpC', 'entities': {'url': {'urls': [{'url': 'https://t.co/EJC6qVVMpC', 'expanded_url': 'https://www.tesla.com', 'display_url': 'tesla.com', 'indices': [0, 23]}]}, 'description': {'urls': []}},
'protected': False, 'followers_count': 18537666, 'friends_count': 33, 'listed_count': 21379, 'created_at': 'Sun Feb 10 01:12:32 +0000 2008', 'favourites_count': 4469, 'utc_offset': None, 'time_zone': None, 'geo_enabled': True,
'verified': True, 'statuses_count': 7927, 'lang': None, 'status': {'created_at': 'Mon Dec 05 01:57:32 +0000 2022', 'id': 1599583722034311169, 'id_str': '1599583722034311169', 'text': '@OwenSparks_ ✌🎄🎶', 'truncated': False,
'entities': {'hashtags': [], 'symbols': [], 'user_mentions': [{'screen_name': 'OwenSparks_', 'name': 'Owen Sparks 🌎', 'id': 829829626180599808, 'id_str': '829829626180599808', 'indices': [0, 12]}], 'urls': []}, 'source': '<a
href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a>', 'in_reply_to_status_id': 1599331117580320768, 'in_reply_to_status_id_str': '1599331117580320768', 'in_reply_to_user_id': 829829626180599808,
'in_reply_to_user_id_str': '829829626180599808', 'in_reply_to_screen_name': 'OwenSparks_', 'geo': None, 'coordinates': None, 'place': None, 'contributors': None, 'is_quote_status': False, 'retweet_count': 24, 'favorite_count': 358,
'favorited': False, 'retweeted': False, 'lang': 'und'}, 'contributors_enabled': False, 'is_translator': False, 'is_translation_enabled': False, 'profile_background_color': '666666', 'profile_background_image_url':
'http://abs.twimg.com/images/themes/theme1/bg.png', 'profile_background_image_url_https': 'https://abs.twimg.com/images/themes/theme1/bg.png', 'profile_background_tile': False, 'profile_image_url':
'http://pbs.twimg.com/profile_images/1337607516008501250/6Ggc4S5n_normal.png', 'profile_image_url_https': 'https://pbs.twimg.com/profile_images/1337607516008501250/6Ggc4S5n_normal.png', 'profile_banner_url':
'https://pbs.twimg.com/profile_banners/13298072/1669832572', 'profile_link_color': 'C90000', 'profile_sidebar_border_color': 'FFFFFF', 'profile_sidebar_fill_color': 'F5EDF5', 'profile_text_color': '555555',
'profile_use_background_image': True, 'has_extended_profile': False, 'default_profile': False, 'default_profile_image': False, 'following': False, 'follow_request_sent': False, 'notifications': False, 'translator_type': 'none',
'withheld_in_countries': []}, id=13298072, id_str='13298072', name='Tesla', screen_name='Tesla', location='', profile_location=None, description='electric vehicles, giant batteries and solar', url='https://t.co/EJC6qVVMpC', entities=
{'url': {'urls': [{'url': 'https://t.co/EJC6qVVMpC', 'expanded_url': 'https://www.tesla.com', 'display_url': 'tesla.com', 'indices': [0, 23]}]}, 'description': {'urls': []}}, protected=False, followers_count=18537666,
friends_count=33, listed_count=21379, created_at=datetime.datetime(2008, 2, 10, 1, 12, 32), favourites_count=4469, utc_offset=None, time_zone=None, geo_enabled=True, verified=True, statuses_count=7927, lang=None, status=Status(_api=
<tweepy.api.API object at 0x7f08e7cdb5e0>, _json={'created_at': 'Mon Dec 05 01:57:32 +0000 2022', 'id': 1599583722034311169, 'id_str': '1599583722034311169', 'text': '@OwenSparks_ ✌🎄🎶', 'truncated': False, 'entities':
{'hashtags': [], 'symbols': [], 'user_mentions': [{'screen_name': 'OwenSparks_', 'name': 'Owen Sparks 🌎', 'id': 829829626180599808, 'id_str': '829829626180599808', 'indices': [0, 12]}], 'urls': []}, 'source': '<a
href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a>', 'in_reply_to_status_id': 1599331117580320768, 'in_reply_to_status_id_str': '1599331117580320768', 'in_reply_to_user_id': 829829626180599808,
'in_reply_to_user_id_str': '829829626180599808', 'in_reply_to_screen_name': 'OwenSparks_', 'geo': None, 'coordinates': None, 'place': None, 'contributors': None, 'is_quote_status': False, 'retweet_count': 24, 'favorite_count': 358,
'favorited': False, 'retweeted': False, 'lang': 'und'}, created_at=datetime.datetime(2022, 12, 5, 1, 57, 32), id=1599583722034311169, id_str='1599583722034311169', text='@OwenSparks_ ✌🎄🎶', truncated=False, entities={'hashtags':
[], 'symbols': [], 'user_mentions': [{'screen_name': 'OwenSparks_', 'name': 'Owen Sparks 🌎', 'id': 829829626180599808, 'id_str': '829829626180599808', 'indices': [0, 12]}], 'urls': []}, source='Twitter for iPhone',
source_url='http://twitter.com/download/iphone', in_reply_to_status_id=1599331117580320768, in_reply_to_status_id_str='1599331117580320768', in_reply_to_user_id=829829626180599808, in_reply_to_user_id_str='829829626180599808',
in_reply_to_screen_name='OwenSparks_', geo=None, coordinates=None, place=None, contributors=None, is_quote_status=False, retweet_count=24, favorite_count=358, favorited=False, retweeted=False, lang='und'), contributors_enabled=False,
is_translator=False, is_translation_enabled=False, profile_background_color='666666', profile_background_image_url='http://abs.twimg.com/images/themes/theme1/bg.png',
profile_background_image_url_https='https://abs.twimg.com/images/themes/theme1/bg.png', profile_background_tile=False, profile_image_url='http://pbs.twimg.com/profile_images/1337607516008501250/6Ggc4S5n_normal.png',
profile_image_url_https='https://pbs.twimg.com/profile_images/1337607516008501250/6Ggc4S5n_normal.png', profile_banner_url='https://pbs.twimg.com/profile_banners/13298072/1669832572', profile_link_color='C90000',
profile_sidebar_border_color='FFFFFF', profile_sidebar_fill_color='F5EDF5', profile_text_color='555555', profile_use_background_image=True, has_extended_profile=False, default_profile=False, default_profile_image=False,
following=False, follow_request_sent=False, notifications=False, translator_type='none', withheld_in_countries=[])
```


```python
[ ] tweet = api.user_timeline(screen_name="LillyPad", q="misleading", count=1, exclude_replies=True, tweet_mode="extended")[0]

    print(tweet)
```


```
Status(_api=<tweepy.api.API object at 0x7f08e7cdb5e0>, _json={'created_at': 'Thu Nov 10 21:09:01 +0000 2022', 'id': 1590813806275469333, 'id_str': '1590813806275469333', 'full_text': 'We apologize to those who have been served a misle
```


✓ 0s completed at 7:41 PM

TWITTER DATA

Using Twitter's Elevate API access, I was able to get intricate stats on the target companies official Twitter accounts, as well as a parody account that was still up and a company's response tweet.

EXPLORATORY DATA ANALYSIS

TWITTER API LIMITATIONS

- Twitter API will not be useful in analyzing information on tweets posted by Parody Accounts – or the accounts themselves
- Data on reply counts, quote tweets and impressions are unavailable – which is unfortunate – because as Twitter has grown over the years the API seems to have not added features to keep up with new developments

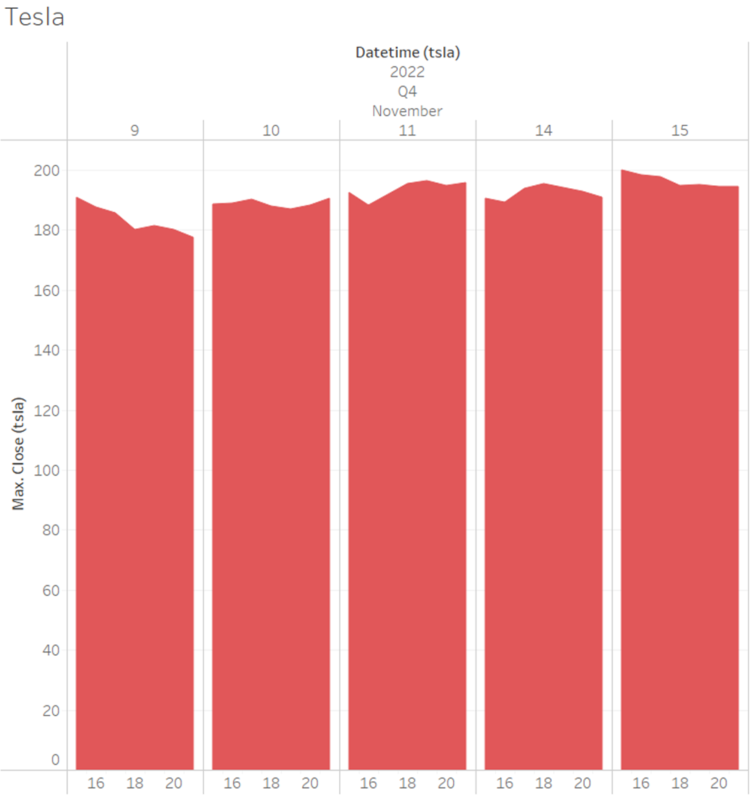
YFINANCE DATA

- There were stock splits in the data downloaded from the Yahoo Finance website, but none took place during the week that I'm focusing on. I dropped the column for the week of hourly data but might have to account for it with the yearly data.
- Dividends for the relatively short time period that I am focused on are irrelevant for what I am aiming to examine here

TABLEAU VISUALIZATIONS

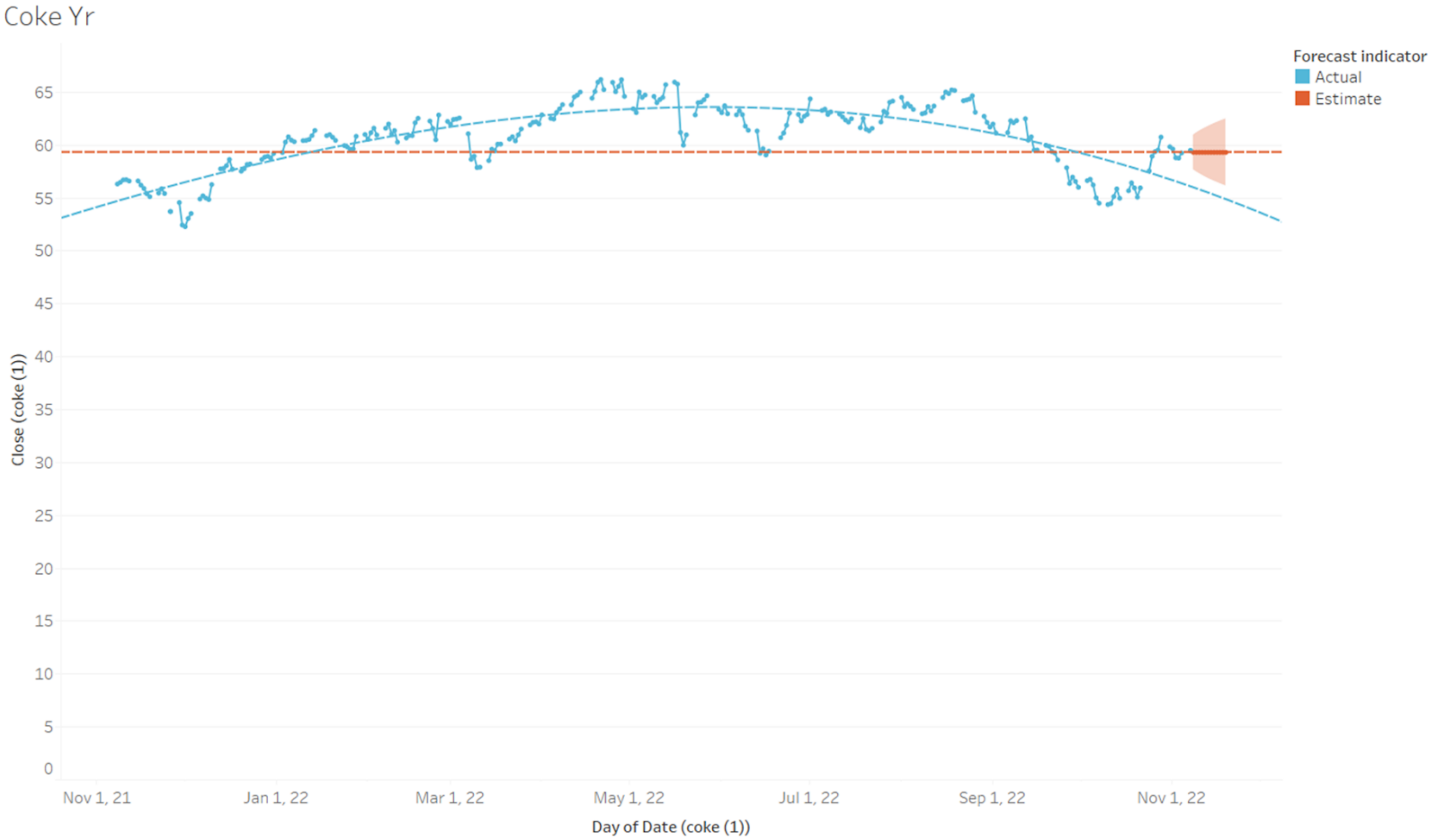
THE VISUALIZATIONS

The trend of sum of Close (tsla (1)) (actual & forecast) for Date (tsla (1)) Day. Color shows details about Forecast indicator.

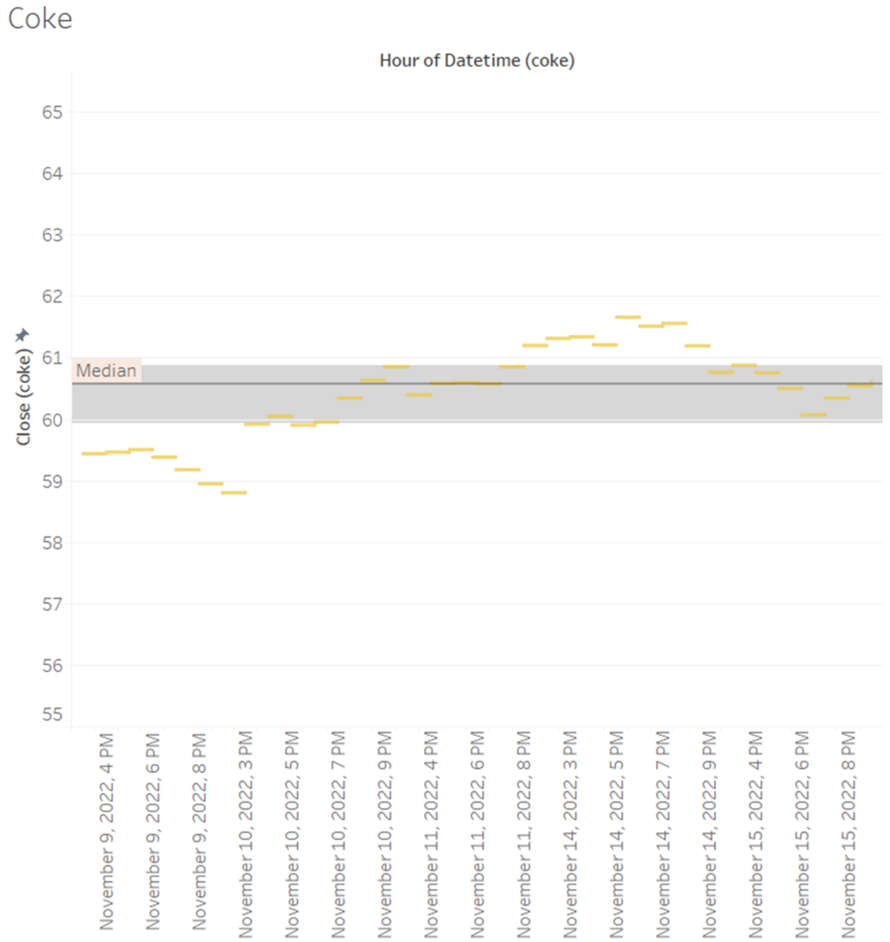


Maximum of Close (tsla) for each Datetime (tsla) Hour broken down by Datetime (tsla) Year, Datetime (tsla) Quarter, Datetime (tsla) Month and Datetime (tsla) Day.

THE VISUALIZATIONS

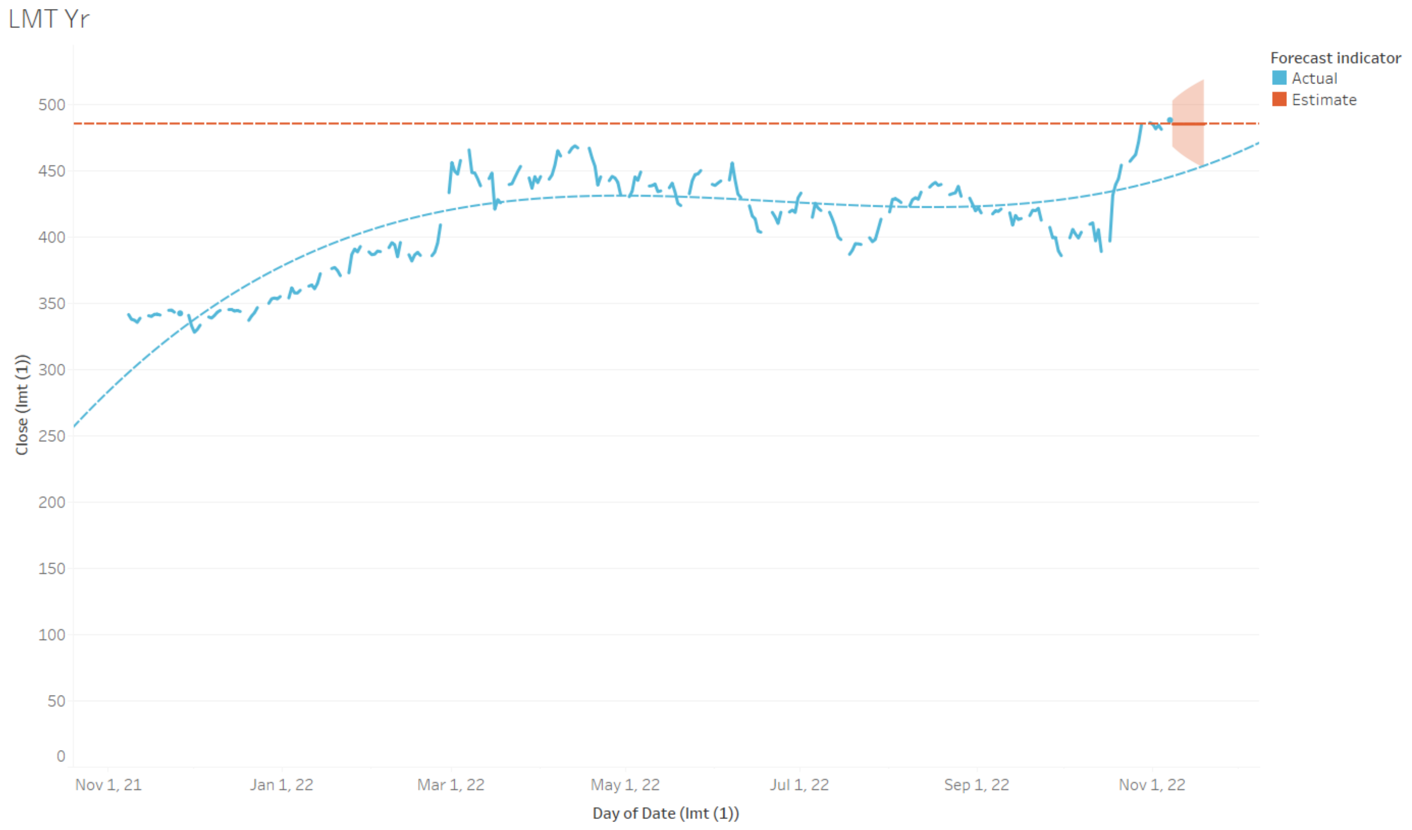


The trend of sum of Close (coke (1)) (actual & forecast) for Date (coke (1)) Day. Color shows details about Forecast indicator.

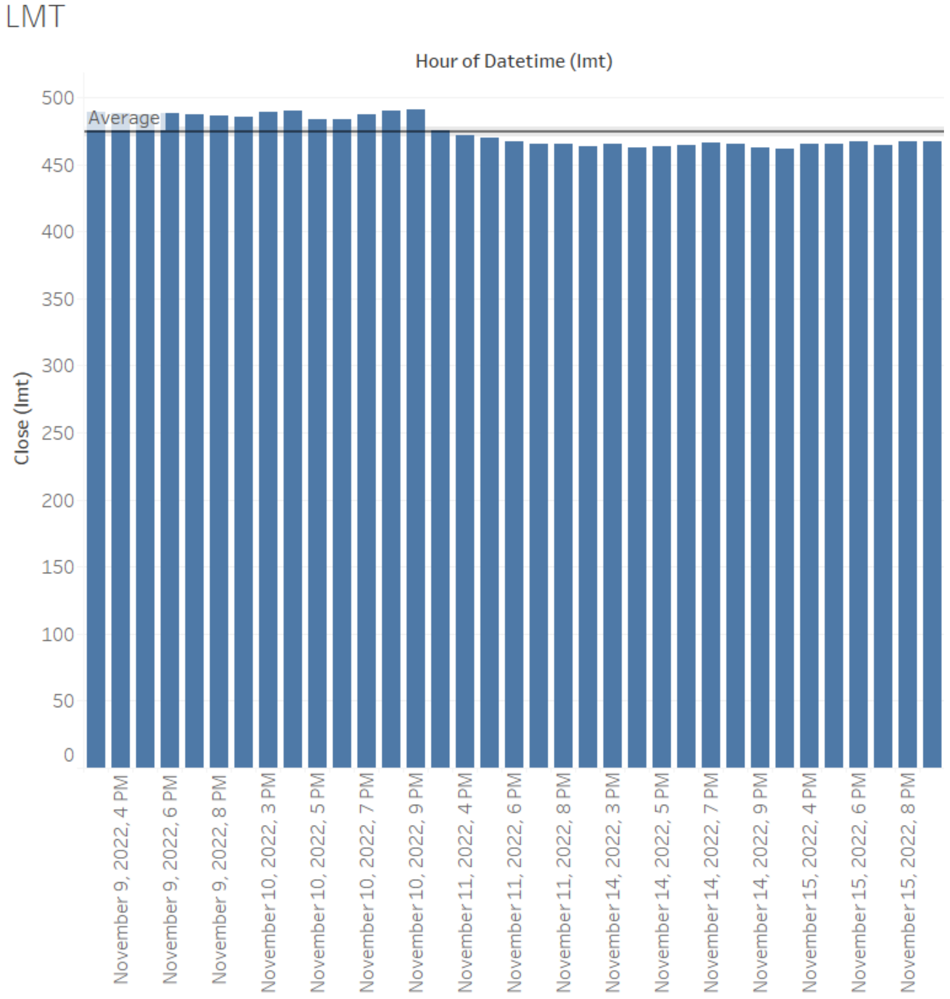


The trend of sum of Close (coke) for Datetime (coke) Hour.

THE VISUALIZATIONS

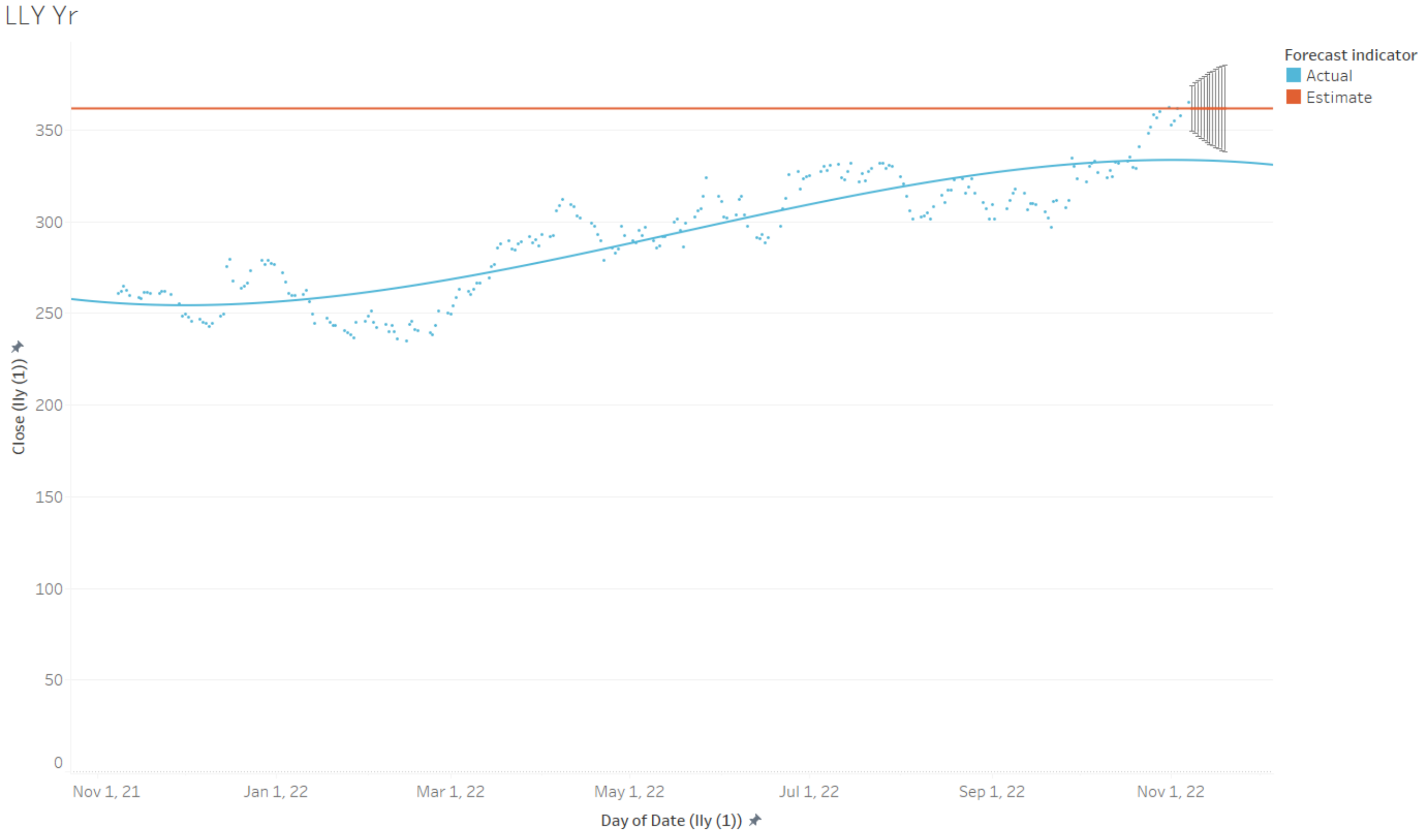


The trend of sum of Close (lmt (1)) (actual & forecast) for Date (lmt (1)) Day. Color shows details about Forecast indicator.

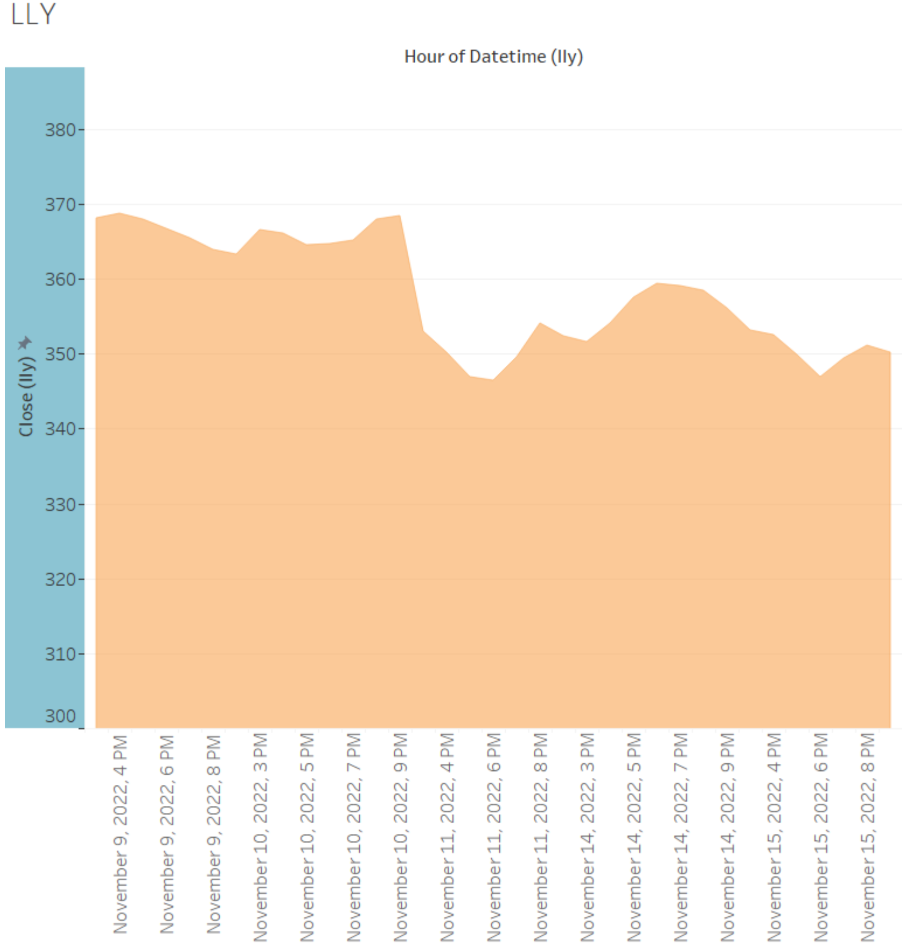


Sum of Close (lmt) for each Datetime (lmt) Hour.

THE VISUALIZATIONS

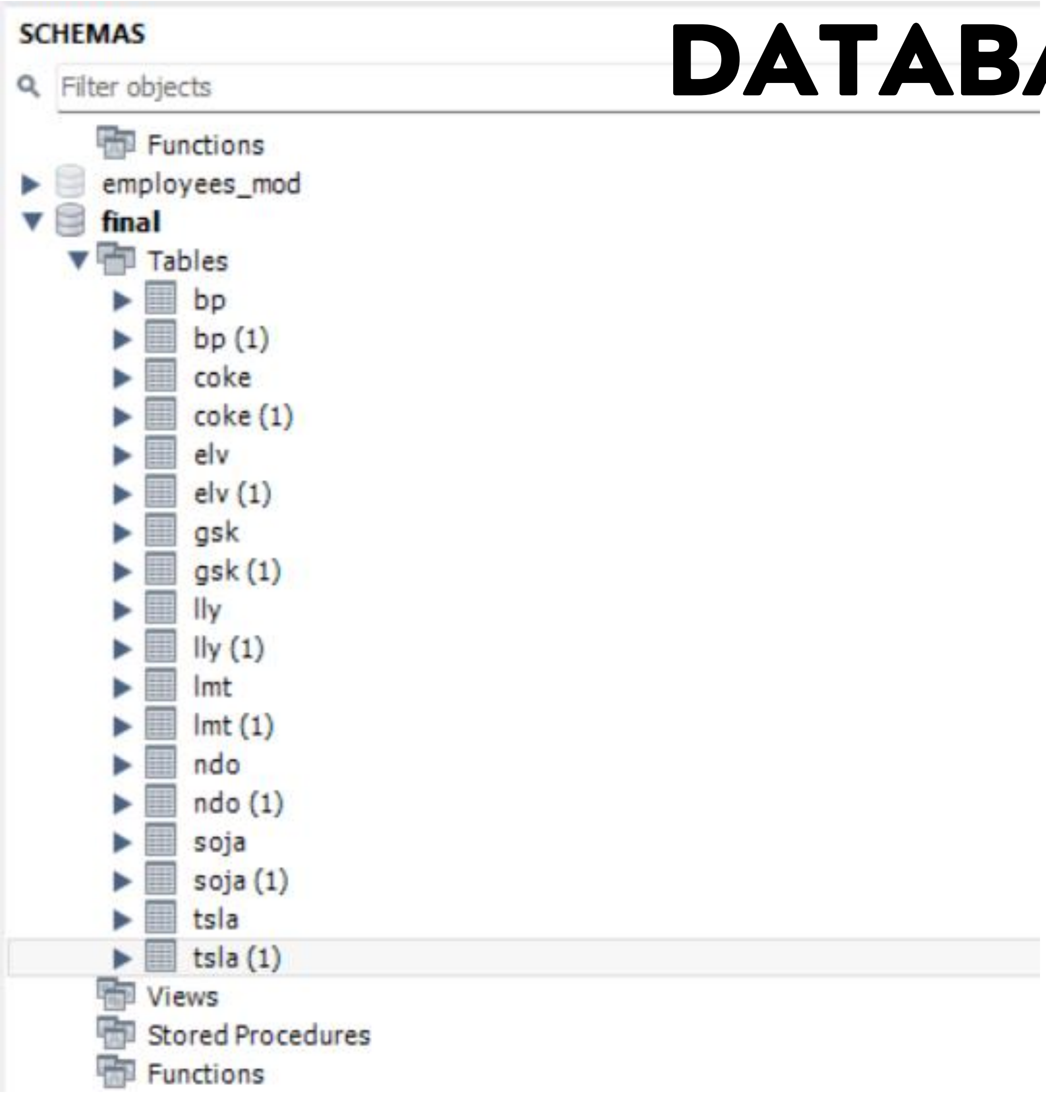


The plot of sum of Close (lly (1)) (actual & forecast) for Date (lly (1)) Day. Color shows details about Forecast indicator.



Sum of Close (lly) for each Datetime (lly) Hour.

DATABASE SELECTION



SQL (MySQL)

I chose to use MySQL to host and manage my database, because I wanted a Local Host that I could then easily connect to Tableau. Since I had previous experience importing CSV files into SQL using the Table Data import wizard, the interface was practical for my use needs. From there, I would just need to do some simple queries to verify that I have the necessary data, imported correctly from my CSV files.

- Relationship database
- Easy to conceive and set up
- Most common
- Easily reused

SQL BENEFITS

Highly Scalable, MySQL can handle almost any amount of data

With top-ranked speeds, it's simple to syncronize with Tableau

MySQL server has been thoroughly tested to prevent memory leaks

DATA INSERTION AND QUERIES

Importing my data with MySQL

IMPORT WIZARD

Everything was in clean CSV files, so by using MySQL's Table Data Import Wizard, I was able to generate nine tables for each of my nine stocks

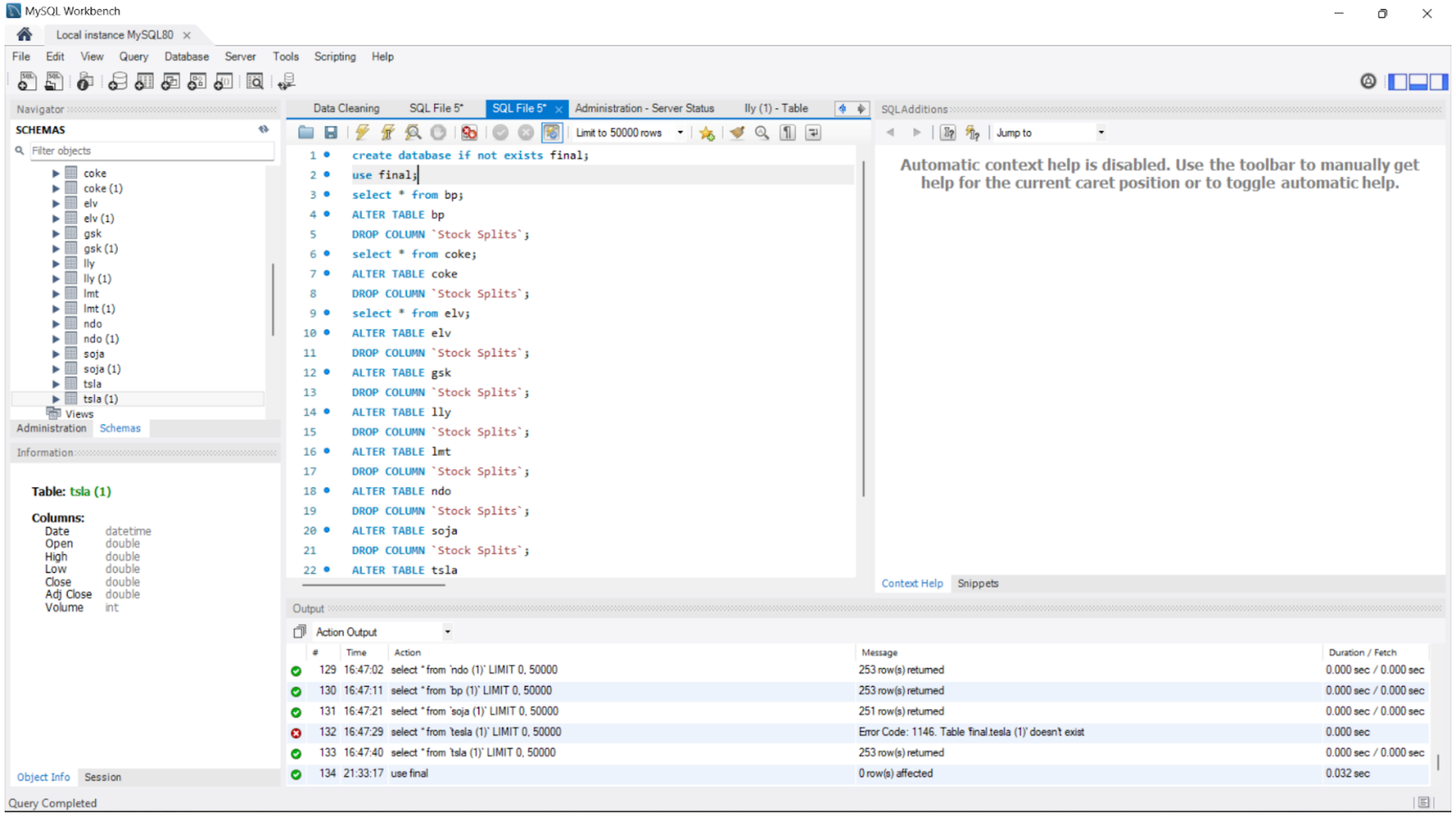
DROP COLUMNS

From there, I inserted the data in the correct formats, and proceeded to drop the "Stock Splits" column from each of the nine respective tables

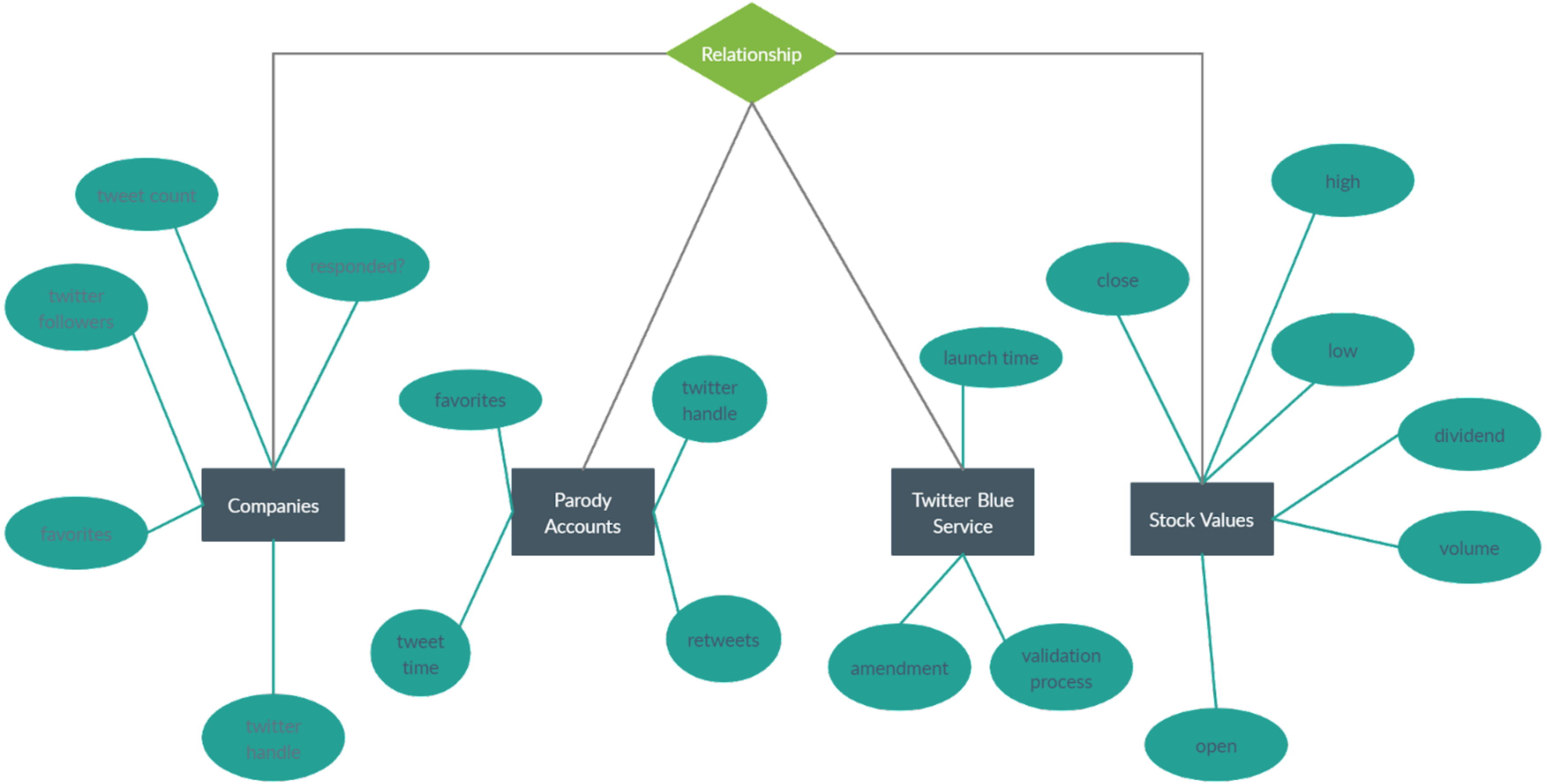
INTERSECT OPERATOR

To verify there were no repeats, I selected all the dates from each table and assigned them to a variable & checked them using the INTERSECT operator

SQL INSERTION AND QUERIES



ENTITY RELATIONSHIP DIAGRAM



MACHINE LEARNING MODEL

Why I chose Long Short-Term Memory (LSTM) :

THE PERFECT TIME-SERIES FIT

LSTM models are well-suited for analyzing time-series data, such as stock price data, because they can capture long-term dependencies. This allows the model to make more accurate predictions based on past trends and patterns.

WELL-ADAPTED TO VOLAITILITY

LSTM can handle data with a high level of volatility. This is because LSTM models have a built-in mechanism to forget irrelevant information, which allows them to focus on the most relevant/ predictive features in the data.

WITH FREEDOM TO INVENT

LSTM models can learn to impute missing values based on the surrounding data, rather than requiring the data to be complete. This makes LSTM models a robust and flexible choice for analyzing stock data.

MACHINE LEARNING CODE

```
[65] stock_data["Date"]=pd.to_datetime(stock_data.Date,format="%Y-%m-%d")
     stock_data.index=stock_data['Date']

     plt.figure(figsize=(16,8))
     plt.plot(stock_data["Close"],label='Close Price history')

[<matplotlib.lines.Line2D at 0x7fe1f7ae43d0>]
```

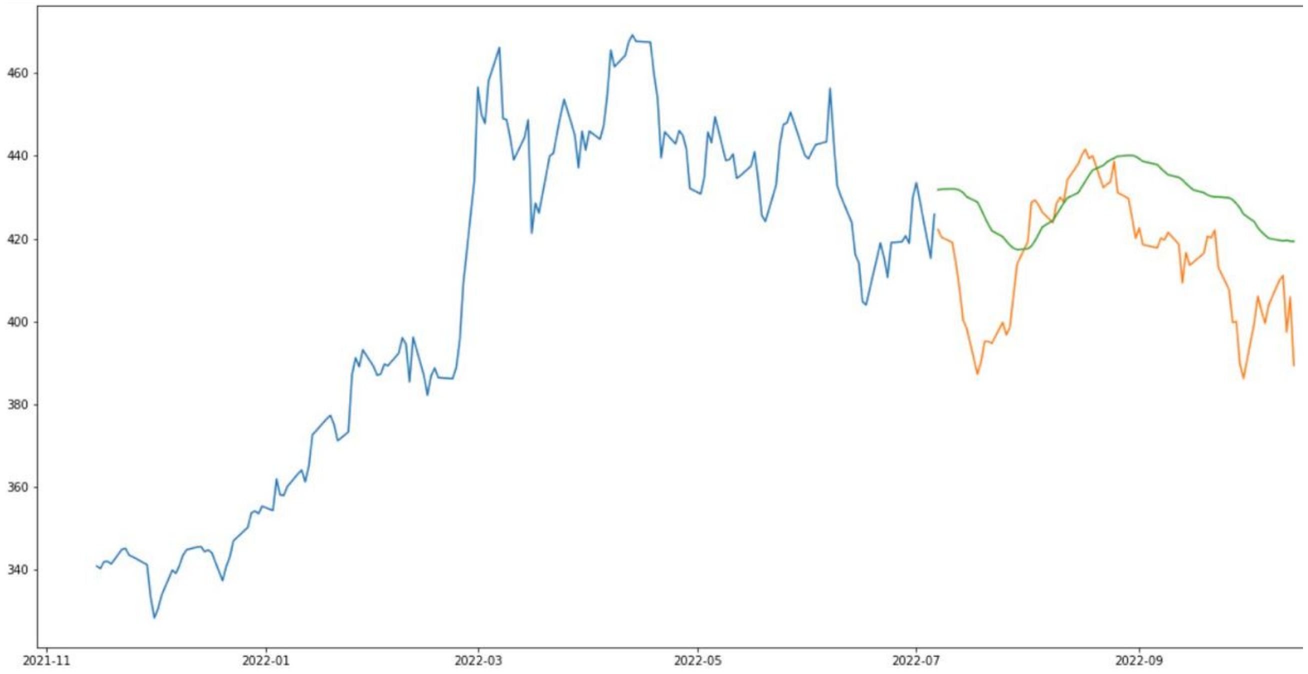

```
[67] data=stock_data.sort_index(ascending=True,axis=0)
     new_dataset=pd.DataFrame(index=range(0,len(stock_data)),columns=['Date','Close'])

[68] for i in range(0,len(data)):
         new_dataset["Date"][i]=data['Date'][i]
         new_dataset["Close"][i]=data["Close"][i]

[69] scaler=MinMaxScaler(feature_range=(0,1))

     train_data=final_dataset[0:160,:]
     valid_data=final_dataset[160:,:]
```

```
[70] new_dataset.index=new_dataset["Date"]
     new_dataset.drop("Date",axis=1,inplace=True)

     final_dataset=new_dataset.values
     scaler=MinMaxScaler(feature_range=(0,1))

[71] scaled_data=scaler.fit_transform(final_dataset)

     x_train_data,y_train_data=[],[]

     for i in range(60,len(train_data)):
         x_train_data.append(scaled_data[i-60:i,0])
         y_train_data.append(scaled_data[i,0])

     x_train_data,y_train_data=np.array(x_train_data),np.array(y_train_data)

     x_train_data=np.reshape(x_train_data,(x_train_data.shape[0],x_train_data.shape[1],1))

[72] lstm_model=Sequential()
     lstm_model.add(LSTM(units=50,return_sequences=True,input_shape=(x_train_data.shape[1],1)))
     lstm_model.add(LSTM(units=50))
     lstm_model.add(Dense(1))

     inputs_data=new_dataset[len(new_dataset)-len(valid_data)-60:].values
     inputs_data=inputs_data.reshape(-1,1)
     inputs_data=scaler.transform(inputs_data)

     lstm_model.compile(loss='mean_squared_error',optimizer='adam')
     lstm_model.fit(x_train_data,y_train_data,epochs=1,batch_size=1,verbose=2)

     100/100 - 5s - loss: 0.0380 - 5s/epoch - 50ms/step
     <keras.callbacks.History at 0x7fe1fca01f40>

[73] X_test=[]
     for i in range(60,inputs_data.shape[0]):
         X_test.append(inputs_data[i-60:i,0])
     X_test=np.array(X_test)

     X_test=np.reshape(X_test,(X_test.shape[0],X_test.shape[1],1))
     predicted_closing_price=lstm_model.predict(X_test)
     predicted_closing_price=scaler.inverse_transform(predicted_closing_price)

     3/3 [==============================] - 2s 33ms/step
```

MACHINE LEARNING CODE

```
train_data=new_dataset[:160]
valid_data=new_dataset[160:]
valid_data['Predictions']=predicted_closing_price
plt.plot(train_data["Close"])
plt.plot(valid_data[['Close',"Predictions"]])
```

```
<ipython-input-77-e0fd45caab64>:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  valid_data['Predictions']=predicted_closing_price
[<matplotlib.lines.Line2D at 0x7fe1f5a5c910>,
 <matplotlib.lines.Line2D at 0x7fe1f5a5ca90>]
```



COMPETING APPROACHES

Linear Regression

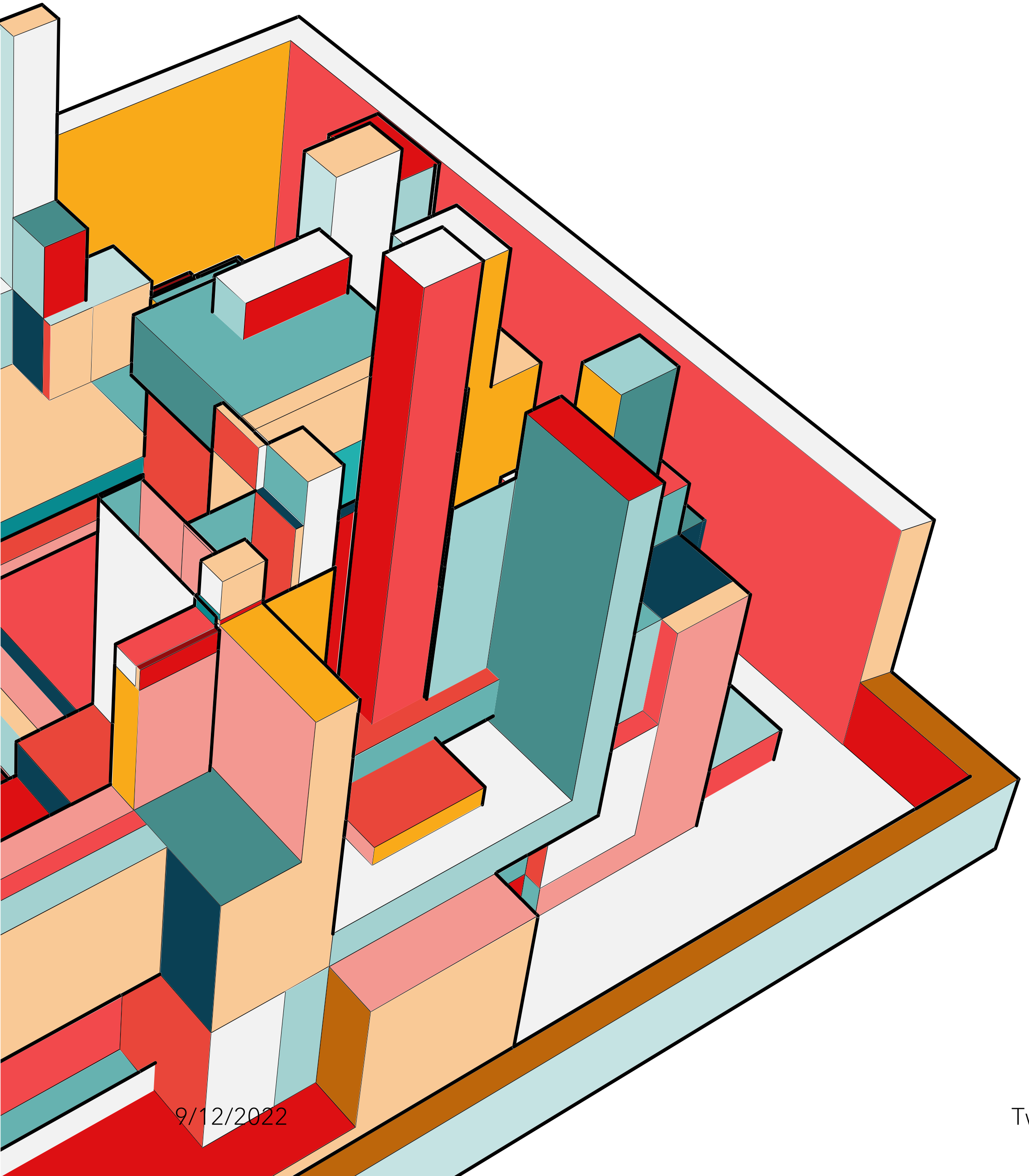
Compared to a linear regression model, an LSTM model is better suited for analyzing stock data because it can capture non-linear relationships in the data. This is important because stock prices are influenced by a wide range of factors, many of which may not have a simple linear relationship with the stock price

Decision Tree Model

Compared to a decision tree model, an LSTM model is better suited for analyzing stock data because it can handle data with a high level of volatility. Decision tree models are sensitive to changes in the data and may not be able to effectively handle the rapid changes that can occur in stock markets

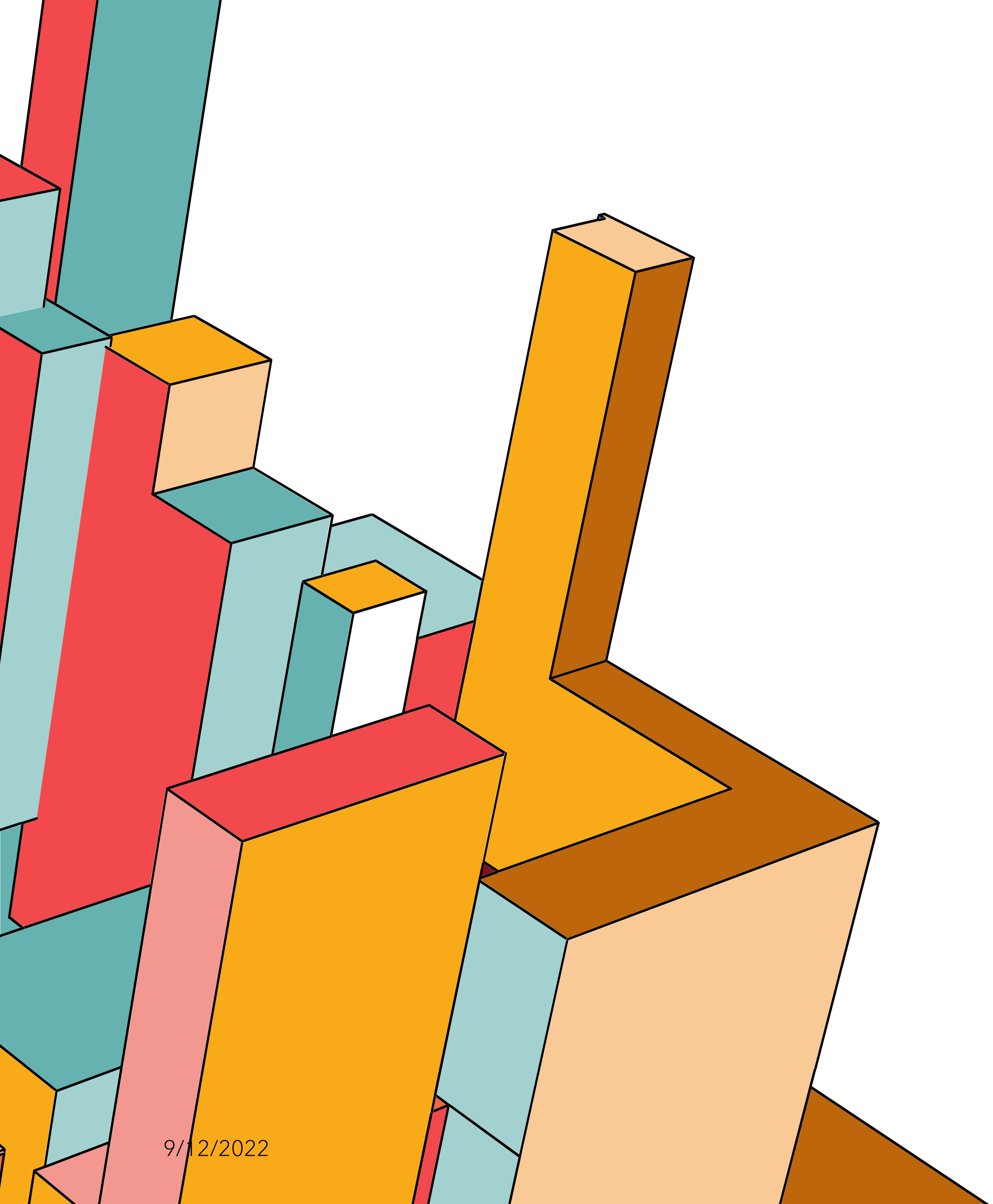
Random Forest Model

Compared to a random forest model, an LSTM model is better suited for analyzing stock data because it can capture long-term dependencies in the data. Random forest models are typically used for classification tasks and may not be able to effectively capture the complex patterns and trends that can occur in stock price data over long periods of time



IMPROVEMENT IDEAS

- Tying stock market changes to reach of parody Tweets
- Using advanced NLP techniques to analyze the content of Twitter posts
- Extracting more nuanced insights about market sentiment and trends
- Incorporating additional data sources into the analysis
- Using news articles, company earnings reports, and economic indicators to provide a more comprehensive view of the factors that may be influencing stock prices.



SUMMARY

- The stock market is a complex and dynamic system that is influenced by a wide range of factors, including macroeconomic conditions, market sentiment, and the performance of individual companies.
- Parody tweets are likely to have a very small and transitory impact on stock prices, if any, compared to these other factors.
- The main takeaway that I've gotten was the scope of impact these tweets were said to have had on the stock market – headlines like "Billions of dollars lost! How Twitter Blue troubled investors" – was clearly sensationalized by the media.
- None of the stocks I analyzed saw unexpected change in valuation, even if they did see millions/ billions of dollars taken off their market caps.

CLOSING STATEMENT

THANK YOU

Evan Strait

0767388598

evanstrait@live.com

https://github.com/evanstrait