Can we use the content of news analytics to predict stock price performance? The ubiquity of data today enables investors at any scale to make better investment decisions. The challenge is ingesting and interpreting the data to determine which data is useful, finding the signal in this sea of information. Two Sigma is passionate about this challenge and is excited to share it with the Kaggle community.

As a scientifically driven investment manager, Two Sigma has been applying technology and data science to financial forecasts for over 17 years. Their pioneering advances in big data, AI, and machine learning have pushed the investment industry forward. Now, they're eager to engage with Kagglers in this continuing pursuit of innovation.

By analyzing news data to predict stock prices, Kagglers have a unique opportunity to advance the state of research in understanding the predictive power of the news. This power, if harnessed, could help predict financial outcomes and generate significant economic impact all over the world.

Data for this competition comes from the following sources:

- Market data provided by Intrinio.
- News data provided by Thomson Reuters. Copyright ©, Thomson Reuters, 2017. All Rights Reserved. Use, duplication, or sale of this service, or data contained herein, except as described in the Competition Rules, is strictly prohibited.

Data Description

In this competition, you will be predicting future stock price returns based on two sources of data:

1. Market data (2007 to present) provided by Intrinio - contains financial market information such as opening price, closing price, trading volume, calculated returns, etc.
2. News data (2007 to present) Source: Thomson Reuters - contains information about news articles/alerts published about assets, such as article details, sentiment, and other commentary.

   Each asset is identified by an `assetCode` (note that a single company may have multiple `assetCodes`). Depending on what you wish to do, you may use the `assetCode`, `assetName`, or `time` as a way to join the market data to news data.

Since this is a Kernels-only, time-based competition, you will not interact directly with the data files as you would in a standard Kaggle competition. You should refer to the submission instructions for details on how to fetch data and make predictions. As noted in the instructions, you will encounter synthetic future data within competition data. This is included to simulate the volume, timeline, and the computational burden that real future data will introduce.

The custom python module also makes it simple to understand what steps are necessary to participate, telling you which assetsCodes to forecast at what time and, by extenstion, which days are market trading days. During stage one, the leaderboard will show performance on a historical period from 2017-01-01 to 2018-07-31. During stage two, Kaggle will re-run participants' selected Kernels on approximately six months of future data.

The data is stored and retrieved as Pandas dataframes in the Kernels environment. Columns types are optimized to minimize space in memory.

Market data

The data includes a subset of US-listed instruments. The set of included instruments changes daily and is determined based on the amount traded and the availability of information. This means that there may be instruments that enter and leave this subset of data. There may therefore be gaps in the data provided, and this does not necessarily imply that that data does not exist (those rows are likely not included due to the selection criteria).

The marketdata contains a variety of returns calculated over different timespans. All of the returns in this set of marketdata have these properties:

- Returns are always calculated either open-to-open (from the opening time of one trading day to the open of another) or close-to-close (from the closing time of one trading day to the open of another).
- Returns are either raw, meaning that the data is not adjusted against any benchmark, or market-residualized (Mktres), meaning that the movement of the market as a whole has been accounted for, leaving only movements inherent to the instrument.
- Returns can be calculated over any arbitrary interval. Provided here are 1 day and 10 day horizons.
- Returns are tagged with 'Prev' if they are backwards looking in time, or 'Next' if forwards looking.

Within the marketdata, you will find the following columns:

- `time`(datetime64[ns, UTC]) - the current time (in marketdata, all rows are taken at 22:00 UTC)

- `assetCode`(object) - a unique id of an asset
- `assetName`(category) - the name that corresponds to a group of `assetCodes`. These may be "Unknown" if the corresponding `assetCode` does not have any rows in the news data.
- `universe`(float64) - a boolean indicating whether or not the instrument on that day will be included in scoring. This value is not provided outside of the training data time period. The trading universe on a given date is the set of instruments that are avilable for trading (the scoring function will not consider instruments that are not in the trading universe). The trading universe changes daily.
- `volume`(float64) - trading volume in shares for the day
- `close`(float64) - the close price for the day (not adjusted for splits or dividends)
- `open`(float64) - the open price for the day (not adjusted for splits or dividends)
- `returnsClosePrevRaw1`(float64) - see returns explanation above
- `returnsOpenPrevRaw1`(float64) - see returns explanation above
- `returnsClosePrevMktres1`(float64) - see returns explanation above
- `returnsOpenPrevMktres1`(float64) - see returns explanation above
- `returnsClosePrevRaw10`(float64) - see returns explanation above
- `returnsOpenPrevRaw10`(float64) - see returns explanation above
- `returnsClosePrevMktres10`(float64) - see returns explanation above
- `returnsOpenPrevMktres10`(float64) - see returns explanation above
- `returnsOpenNextMktres10`**(float64) - 10 day, market-residualized return. This is the target variable used in competition scoring. The market data has been filtered such that `returnsOpenNextMktres10` is always not null.**

News data

The news data contains information at both the news article level and asset level (in other words, the table is intentionally not normalized).

- `time`(datetime64[ns, UTC]) - UTC timestamp showing when the data was available on the feed (second precision)
- `sourceTimestamp`(datetime64[ns, UTC]) - UTC timestamp of this news item when it was created
- `firstCreated`(datetime64[ns, UTC]) - UTC timestamp for the first version of the item
- `sourceId`(object) - an Id for each news item
- `headline`(object) - the item's headline
- `urgency`(int8) - differentiates story types (1: alert, 3: article)
- `takeSequence`(int16) - the take sequence number of the news item, starting at 1. For a given story, alerts and articles have separate sequences.

- `provider`(category) - identifier for the organization which provided the news item (e.g. RTRS for Reuters News, BSW for Business Wire)
- `subjects`(category) - topic codes and company identifiers that relate to this news item. Topic codes describe the news item's subject matter. These can cover asset classes, geographies, events, industries/sectors, and other types.
- `audiences`(category) - identifies which desktop news product(s) the news item belongs to. They are typically tailored to specific audiences. (e.g. "M" for Money International News Service and "FB" for French General News Service)
- `bodySize`(int32) - the size of the current version of the story body in characters
- `companyCount`(int8) - the number of companies explicitly listed in the news item in the subjects field
- `headlineTag`(object) - the Thomson Reuters headline tag for the news item
- `marketCommentary`(bool) - boolean indicator that the item is discussing general market conditions, such as "After the Bell" summaries
- `sentenceCount`(int16) - the total number of sentences in the news item. Can be used in conjunction with firstMentionSentence to determine the relative position of the first mention in the item.
- `wordCount`(int32) - the total number of lexical tokens (words and punctuation) in the news item
- `assetCodes`(category) - list of assets mentioned in the item
- `assetName`(category) - name of the asset
- `firstMentionSentence`(int16) - the first sentence, starting with the headline, in which the scored asset is mentioned.
  - 1: headline
  - 2: first sentence of the story body
  - 3: second sentence of the body, etc
  - 0: the asset being scored was not found in the news item's headline or body text. As a result, the entire news item's text (headline + body) will be used to determine the sentiment score.
- `relevance`(float32) - a decimal number indicating the relevance of the news item to the asset. It ranges from 0 to 1. If the asset is mentioned in the headline, the relevance is set to 1. When the item is an alert (urgency == 1), relevance should be gauged by firstMentionSentence instead.
- `sentimentClass`(int8) - indicates the predominant sentiment class for this news item with respect to the asset. The indicated class is the one with the highest probability.
- `sentimentNegative`(float32) - probability that the sentiment of the news item was negative for the asset
- `sentimentNeutral`(float32) - probability that the sentiment of the news item was neutral for the asset

- `sentimentPositive`(float32) - probability that the sentiment of the news item was positive for the asset
- `sentimentWordCount`(int32) - the number of lexical tokens in the sections of the item text that are deemed relevant to the asset. This can be used in conjunction with wordCount to determine the proportion of the news item discussing the asset.
- `noveltyCount12H`(int16) - The 12 hour novelty of the content within a news item on a particular asset. It is calculated by comparing it with the asset-specific text over a cache of previous news items that contain the asset.
- `noveltyCount24H`(int16) - same as above, but for 24 hours
- `noveltyCount3D`(int16) - same as above, but for 3 days
- `noveltyCount5D`(int16) - same as above, but for 5 days
- `noveltyCount7D`(int16) - same as above, but for 7 days
- `volumeCounts12H`(int16) - the 12 hour volume of news for each asset. A cache of previous news items is maintained and the number of news items that mention the asset within each of five historical periods is calculated.
- `volumeCounts24H`(int16) - same as above, but for 24 hours
- `volumeCounts3D`(int16) - same as above, but for 3 days
- `volumeCounts5D`(int16) - same as above, but for 5 days
- `volumeCounts7D`(int16) - same as above, but for 7 days

Market data provided by Intrinio.