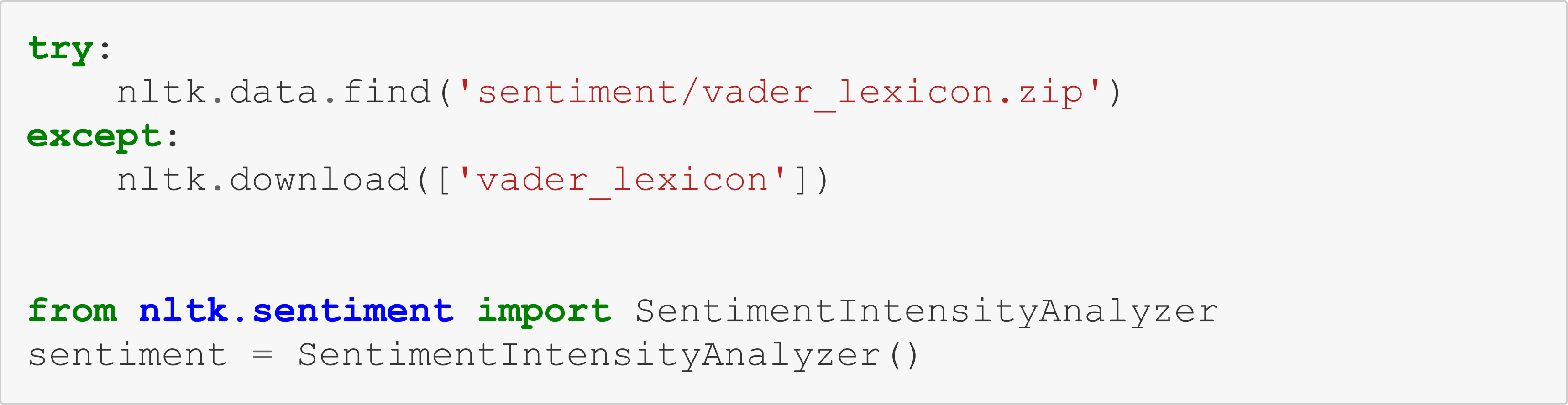
Fundamental Concepts in Data Insight:

Demo: AI & Deep Learning

Fundamentals for a General Audience

The code below loads the "Natural Language Proecssing Toolkit" and downloads the "vader lexicon": a dataset which enables the tookit to perform sentiment analysis.

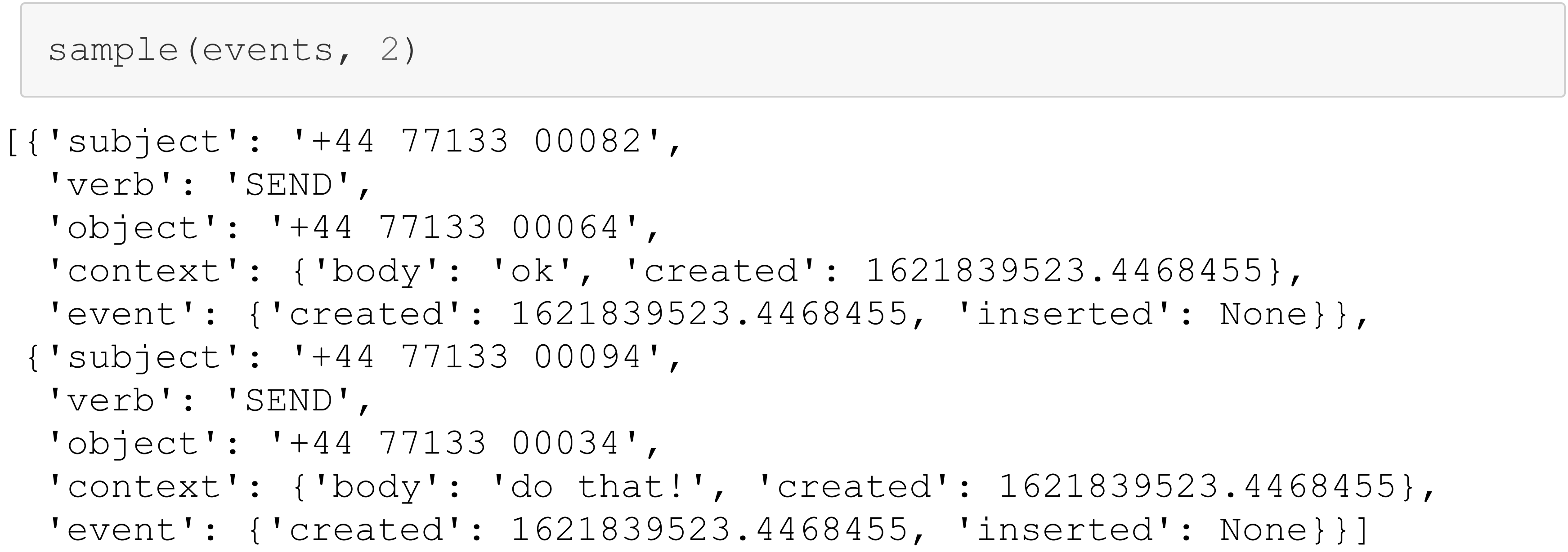


```python
try:
    nltk.data.find('sentiment/vader_lexicon.zip')
except:
    nltk.download(['vader_lexicon'])


from nltk.sentiment import SentimentIntensityAnalyzer
sentiment = SentimentIntensityAnalyzer()
```

What data set are we analysing?

The gang messages log is a simulated set of messages, hypothetically sent between gang members,



```
sample(events, 2)
```

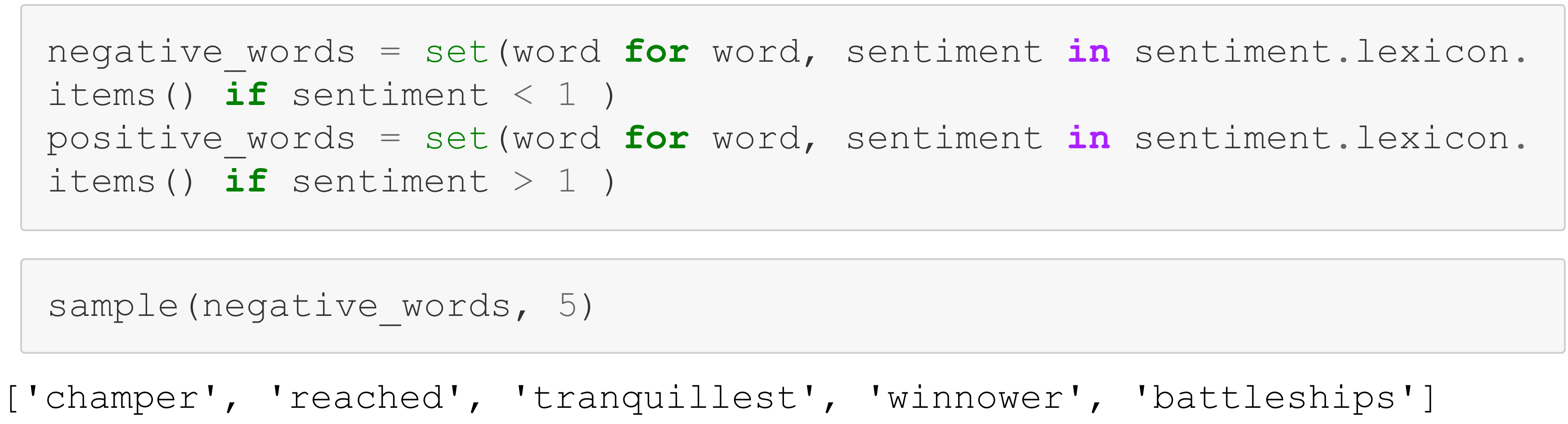
```
[{'subject': '+44 77133 00082',
  'verb': 'SEND',
  'object': '+44 77133 00064',
  'context': {'body': 'ok', 'created': 1621839523.4468455},
  'event': {'created': 1621839523.4468455, 'inserted': None}},
 {'subject': '+44 77133 00094',
  'verb': 'SEND',
  'object': '+44 77133 00034',
  'context': {'body': 'do that!', 'created': 1621839523.4468455},
  'event': {'created': 1621839523.4468455, 'inserted': None}}]
```

What is sentiment analysis?

Sentiment is how positive or negative a fragment of text is; where "positive" is roughly, how "nice" we would find it; and "negative" how "critical".

To produce a sentiment analysis system we need a historical set of words (, phrases, sentences) which have been **labelled by human operators** as positive or negative.

A sample of five such *negative* words are,

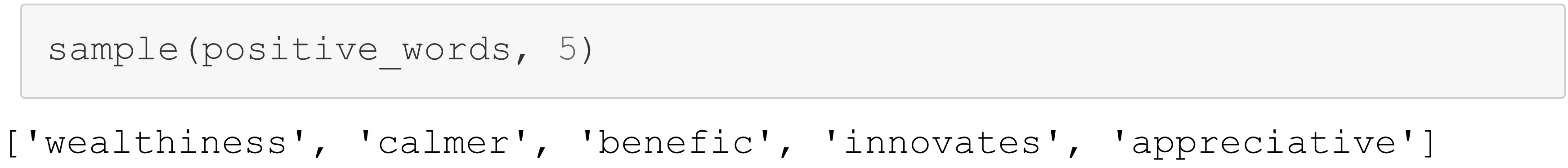


```
negative_words = set(word for word, sentiment in sentiment.lexicon.
items() if sentiment < 1 )
positive_words = set(word for word, sentiment in sentiment.lexicon.
items() if sentiment > 1 )
```

```
sample(negative_words, 5)
```

```
['champer', 'reached', 'tranquillest', 'winnower', 'battleships']
```

...and positive,



```
sample(positive_words, 5)
```

```
['wealthiness', 'calmer', 'benefic', 'innovates', 'appreciative']
```

The sentiment analysis system used here scores each word of some text and aggregates the scores,

```
sentiment.polarity_scores("I am the trustiest peacetime grim outraged president!")
```

```
{'neg': 0.444, 'neu': 0.178, 'pos': 0.379, 'compound': -0.2714}
```

We can see how negative, how positive and how neutral a piece of text is as aggregate score of negative/positive/neutral words. The `compound` score is a net estimate for the sentiment of the whole.

What messages does the dataset contain?

```
sample(set(e['context']['body'] for e in events), 5)
```

```
['ok',
 'where are you?',
 'do it NOW!',
 'do that!',
 "I can't believe we are in this mess!"]
```

What sentiments does the dataset contain?

Let's analyse *all* of these messages for their *sentiment*, with a `cutoff = 0.5`,

```
for event in events:

    score = sentiment.polarity_scores(event['context']['body'])['compound']

    if abs(score) > cutoff:
        print(f"""

        FROM:  {event['subject']}
        TO:    {event['object']}
        SCORE: {score}
        MSG:   {event['context']['body']} """)
```

```
FROM:  +44 77133 00092
TO:    +44 77133 00082
SCORE: -0.7088
MSG:   What the hell is going on!?


FROM:  +44 77133 00064
TO:    +44 77133 00092
SCORE: 0.6239
MSG:   amazing!
```