GTK CYBER

GET ON THE FAST TRACK

Module 7: Anomaly Detection

Agenda for Today

- Overview of Techniques for Anomaly Detection
- Statistical Techniques
- Supervised Techniques
- Unsupervised Techniques

A Problem: Data Validation

A Problem: Data Validation

- Charles was working on a problem of ingesting data into a system
- We knew what good data looked like, and we wanted to detect bad data to prevent corrupted data from being ingested
- The number of columns was large, so we couldn't write rules for each column

Discuss: How you would tackle this problem?

Anomaly Detection is Hard

Anomalous != Bad

"Outliers are not necessarily a bad thing. These are just observations that are not following the same pattern as the other ones. But it can be the case that an outlier is very interesting. For example, if in a biological experiment, a rat is not dead whereas all others are, then it would be very interesting to understand why. This could lead to new scientific discoveries. So, it is important to detect outliers."

– Pierre Lafaye de Micheaux, Author and Statistician

Anomaly = Outlier

What is an outlier?

An **outlier** or **anomaly** is a data point which differs from the rest of the observations in a dataset.

Anomaly Detection != Novelty Detection

Anomaly Detection != Novelty Detection

- Anomaly detection involves learning from data that has both inliers and outliers.

- Novelty detection involves learning starting with a dataset that does not have outliers

Categories of Anomaly Detection

- Forecasting (Supervised Learning)
- Statistical Metrics
- Unsupervised Techniques
- Density Based Methods
- Goodness of fit tests

Python Ecosystem

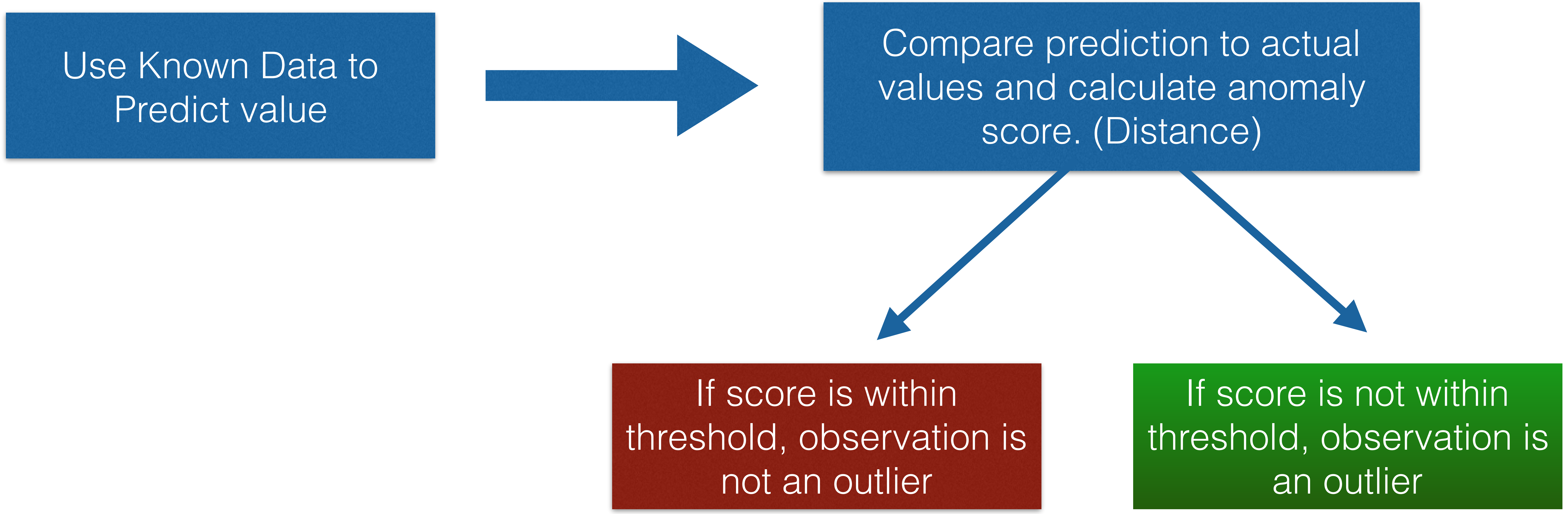
- **Pandas/SciPy/NumPy:** Pandas, SciPy and NumPy can be used for some of the simpler statistical calculations

- **PyOD:** A relatively new module for outlier detection that contains many sophisticated techniques. (**https://pyod.readthedocs.io/en/latest/**)

- **PyFlux:** PyFlux is a module for time series analysis and prediction. (**https://pyflux.readthedocs.io/en/latest/index.html**)

- **Scikit-Learn:** Scikit-learn actually has several algorithms include the OneClassSVM and IsolationForest

Anomaly Detection Process



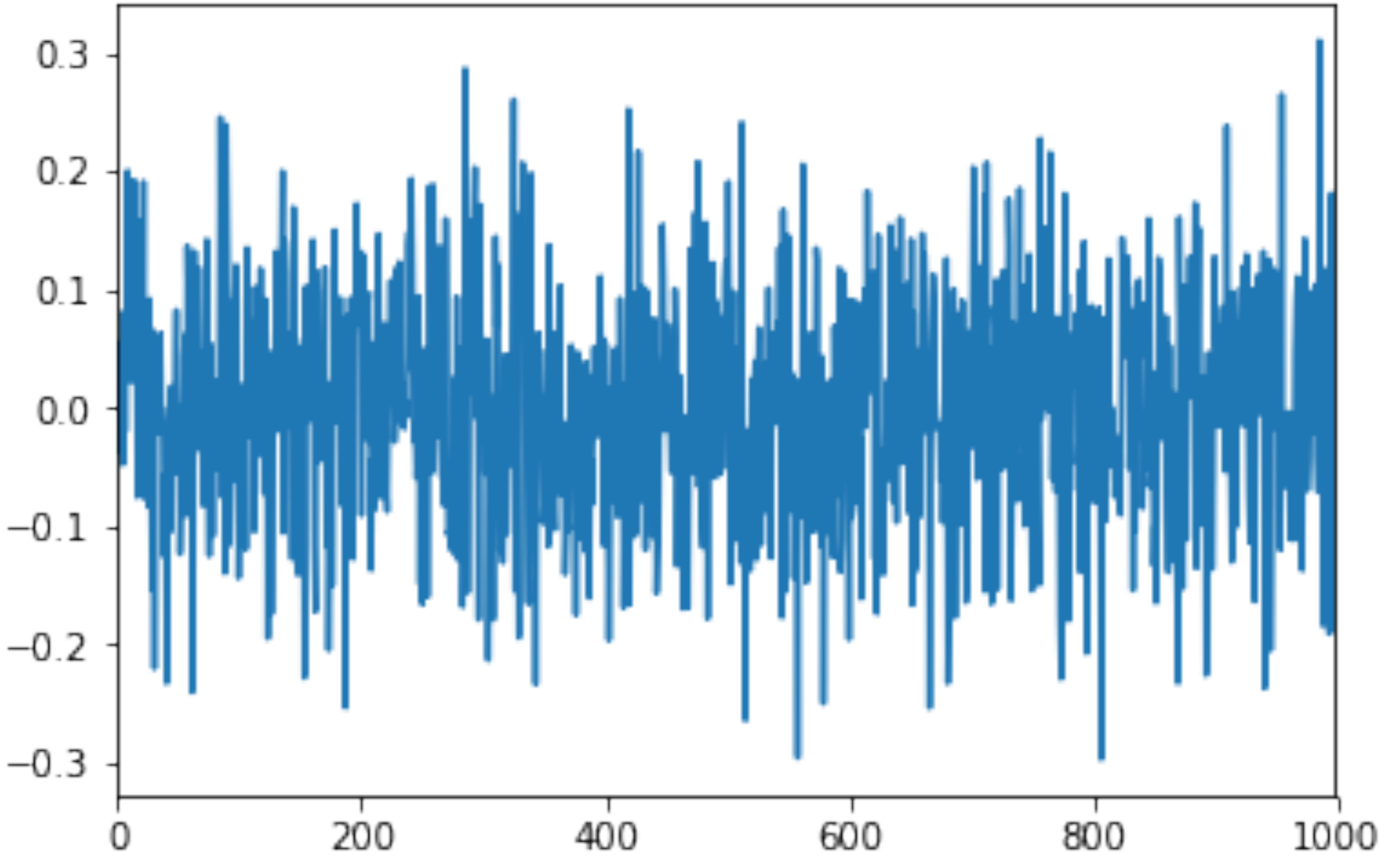
Before you start… there are two questions:

Is the dataset univariate or multivariate?

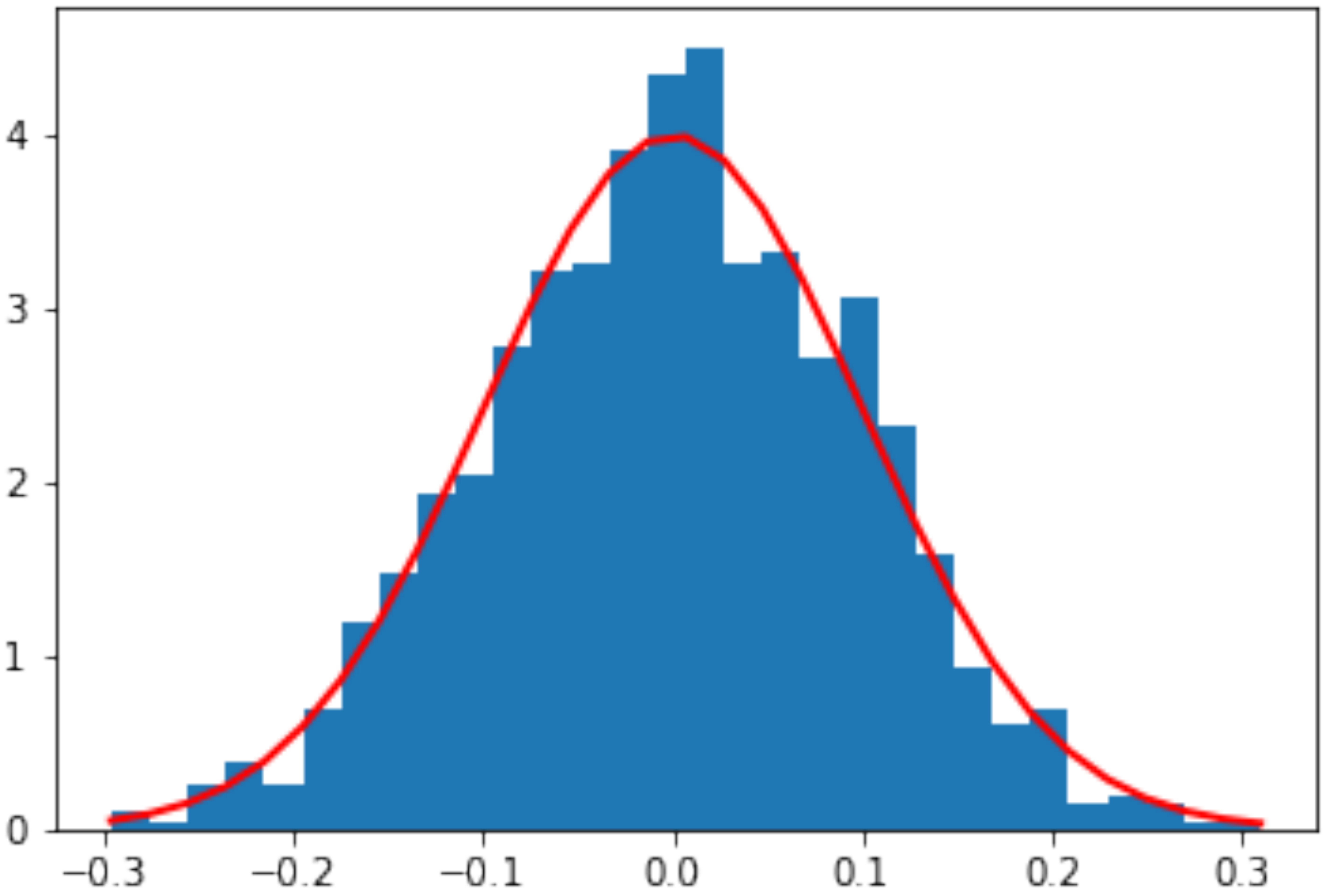
What is the expected distribution of my data, if any?

Simple Statistics

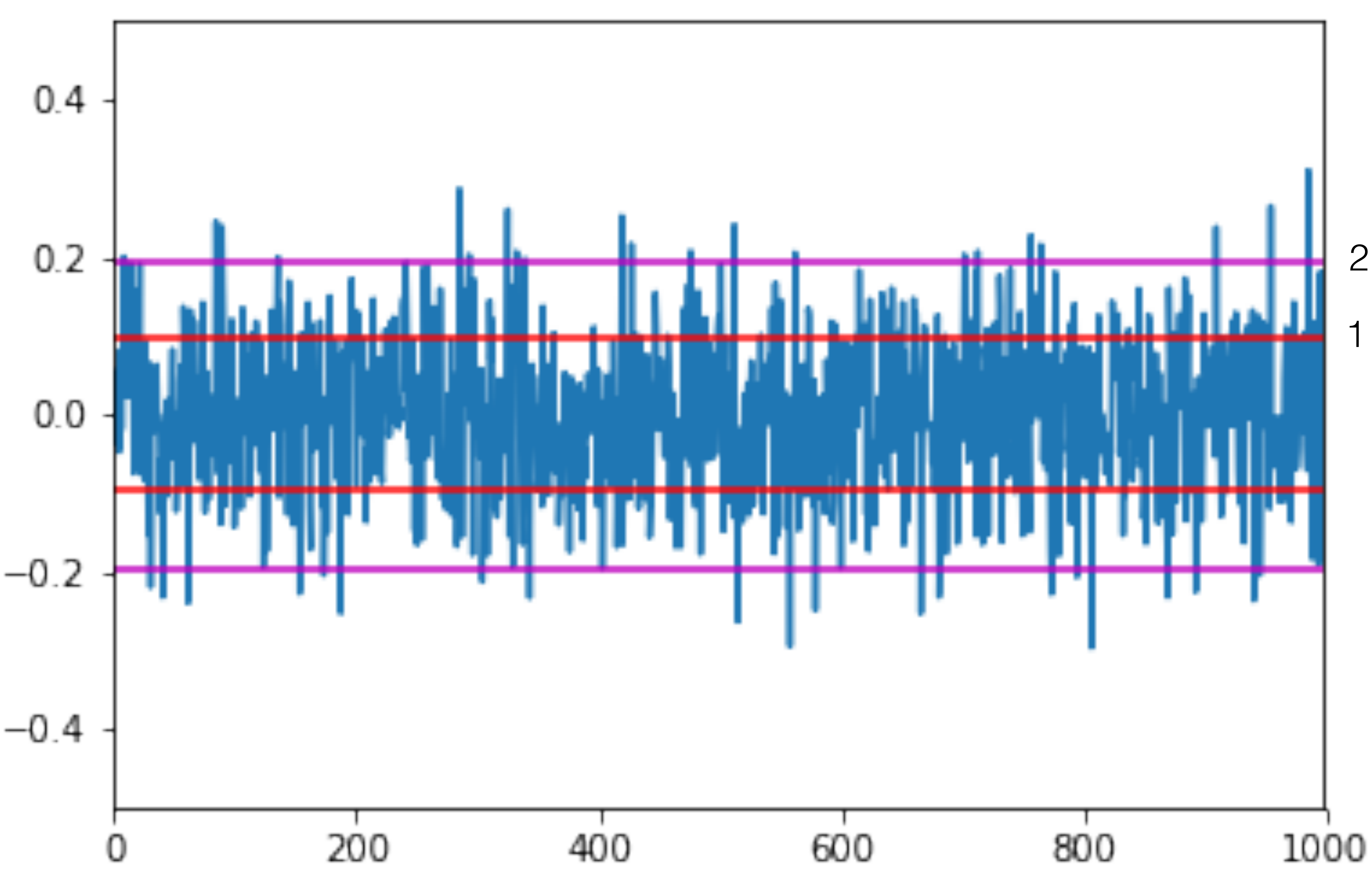
Simple Statistics



Simple Statistics



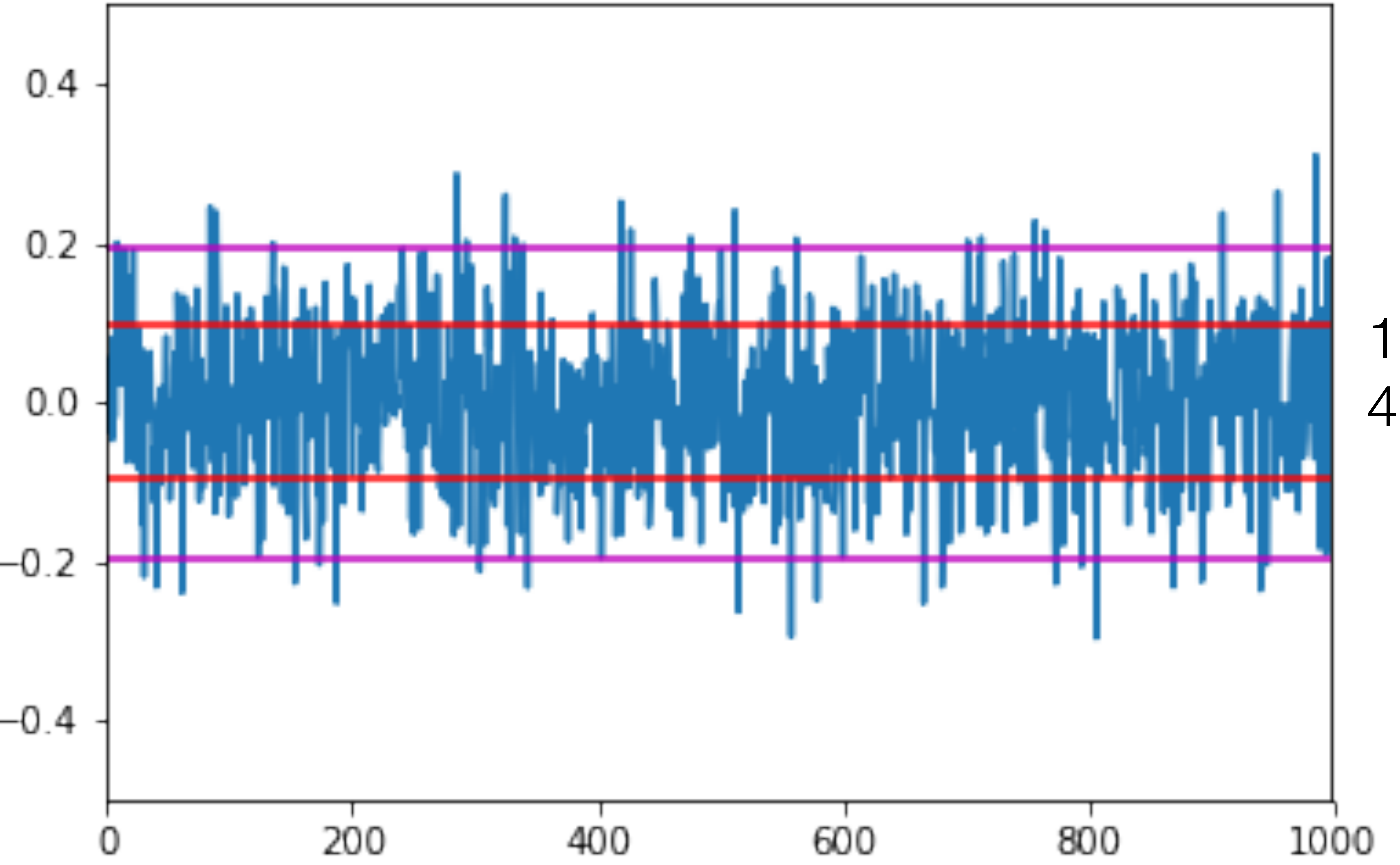
Simple Statistics



2 Standard Deviations

1 Standard Deviation

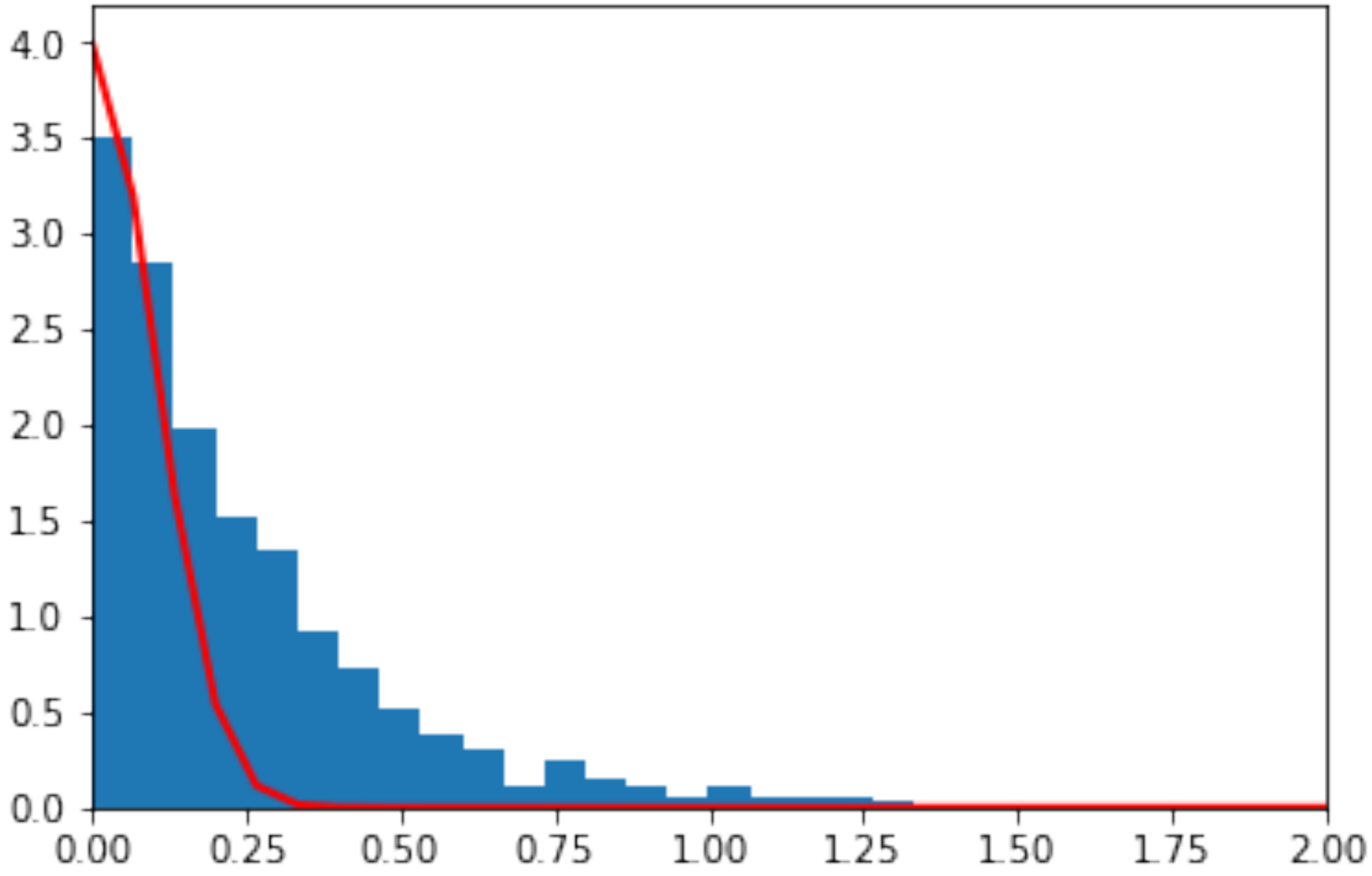
Simple Statistics



167 are outside of 1 deviation
44 are outside of 2 deviations

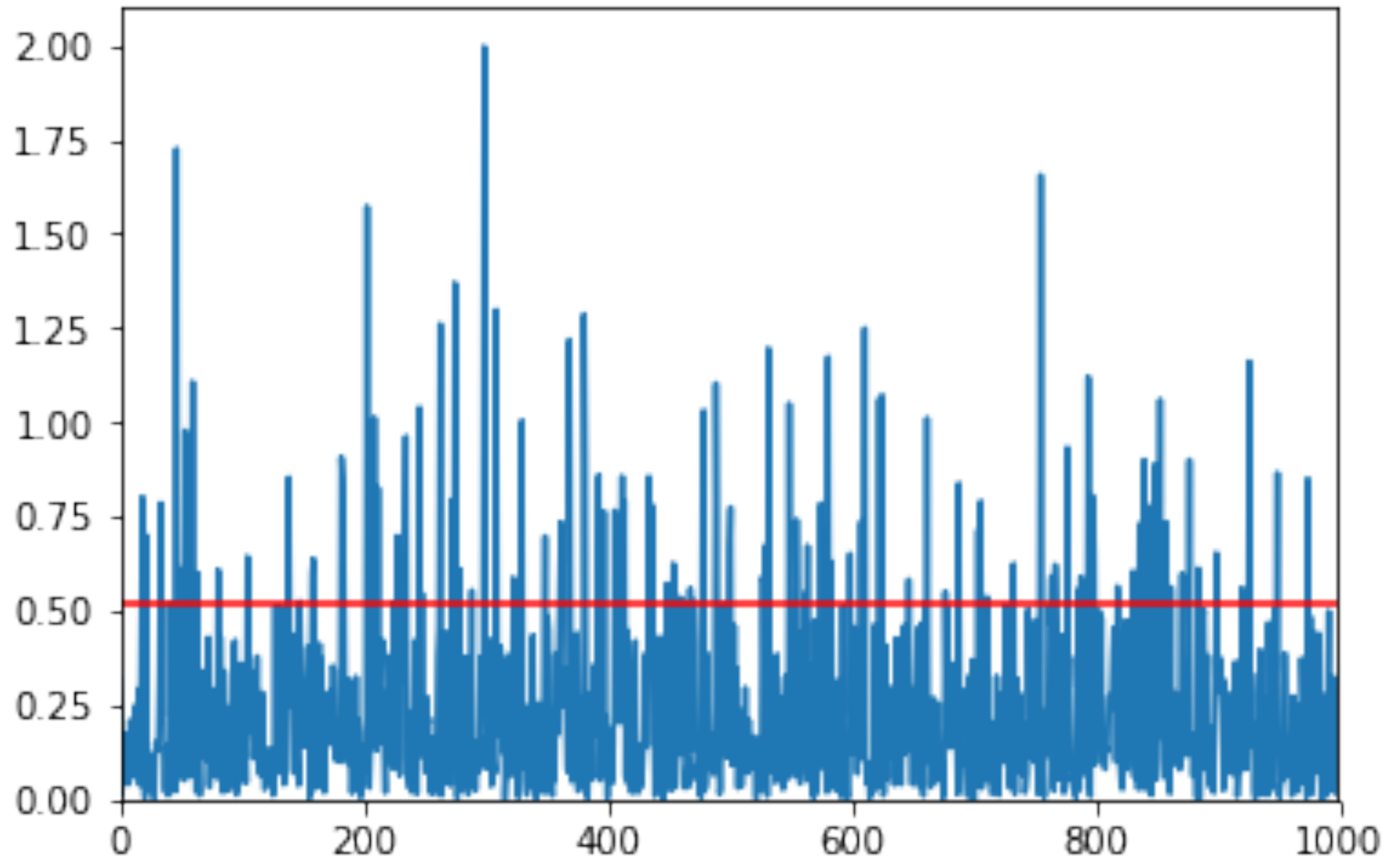
Simple Statistics

What if your data isn't distributed normally?



Simple Statistics

What if your data isn't distributed normally?



123 Outliers

Mean Absolute Deviation

$$\frac{\Sigma\,|x - \bar{x}|}{n}$$

mad = sum(data_point - column_mean) / n

- Mean Absolution Deviation (MAD) is defined as the average of the absolute deviations from a central point.

```
data = pd.Series(<data>)

mad = data.mad()
```

https://en.wikipedia.org/wiki/Average_absolute_deviation

Z-Score measures how many standard deviations an observation is from the mean.

Z-Score measures how many standard deviations an observation is from the mean.

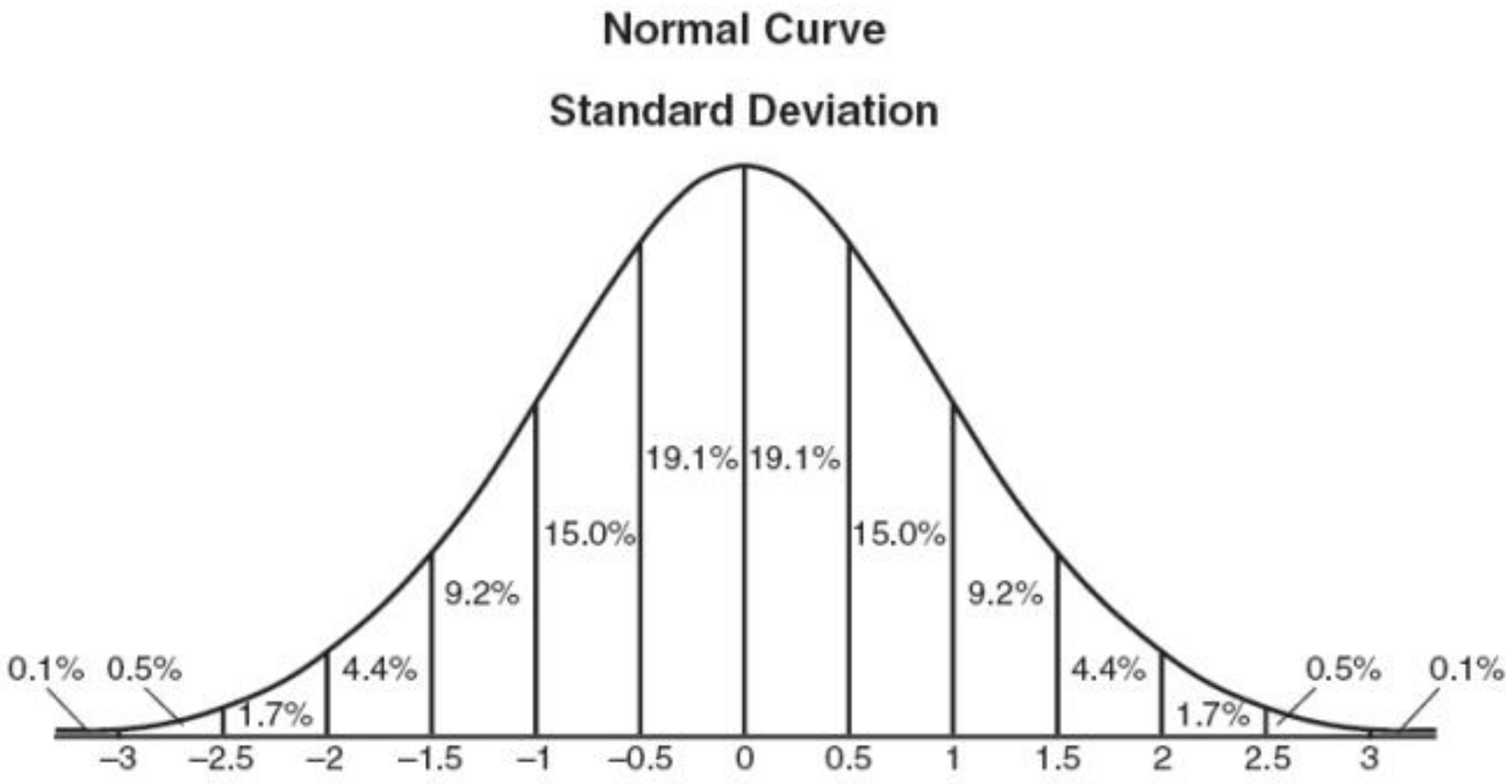
$$z = \frac{x - \mu}{\sigma}$$

```
z_score = (data_point - column_mean) / standard_deviation
```

```
#Use SciPy
from scipy.stats import zscore
df.apply(zscore)

#Use Pandas (Only works for all numeric cols)
df_zscore = (df - df.mean())/df.std()
```

Z-Score



https://towardsdatascience.com/a-brief-overview-of-outlier-detection-techniques-1e0b2c19e561

Z-Score pros:

- It is a very effective method if you can describe the values in the feature space with a gaussian distribution. (Parametric)

- The implementation is very easy using pandas and scipy.stats libraries.

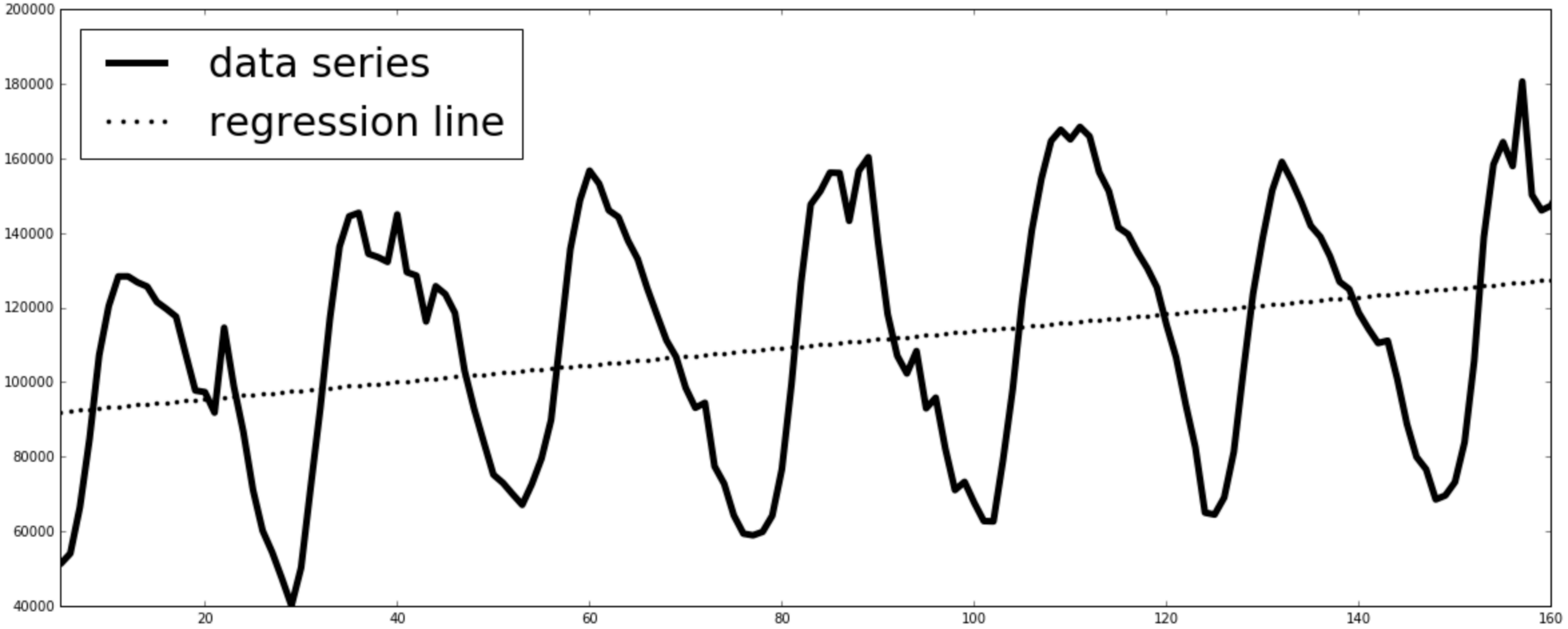
Z-Score cons:

- It is only convenient to use in a low dimensional feature space, in a small to medium sized dataset.

- Is not recommended when distributions can not be assumed to be parametric.

https://towardsdatascience.com/a-brief-overview-of-outlier-detection-techniques-1e0b2c19e561

What if your data looks like this?



https://github.com/oreilly-mlsec/book-resources/blob/master/chapter3/figures/3-01-diurnal-season-trend.png

Forecasting

Forecasting

Forecasting is a **supervised learning technique** which uses past observations to predict future events.

Forecasting is suited for single dimensional, time series data.

Terms

- **Trends:** Long term direction of changes in the data

- **Seasons:** Periodic repetitions of patterns in data

- **Cycles:** General changes in the data that have pattern similarities but vary in periodicity

ARIMA

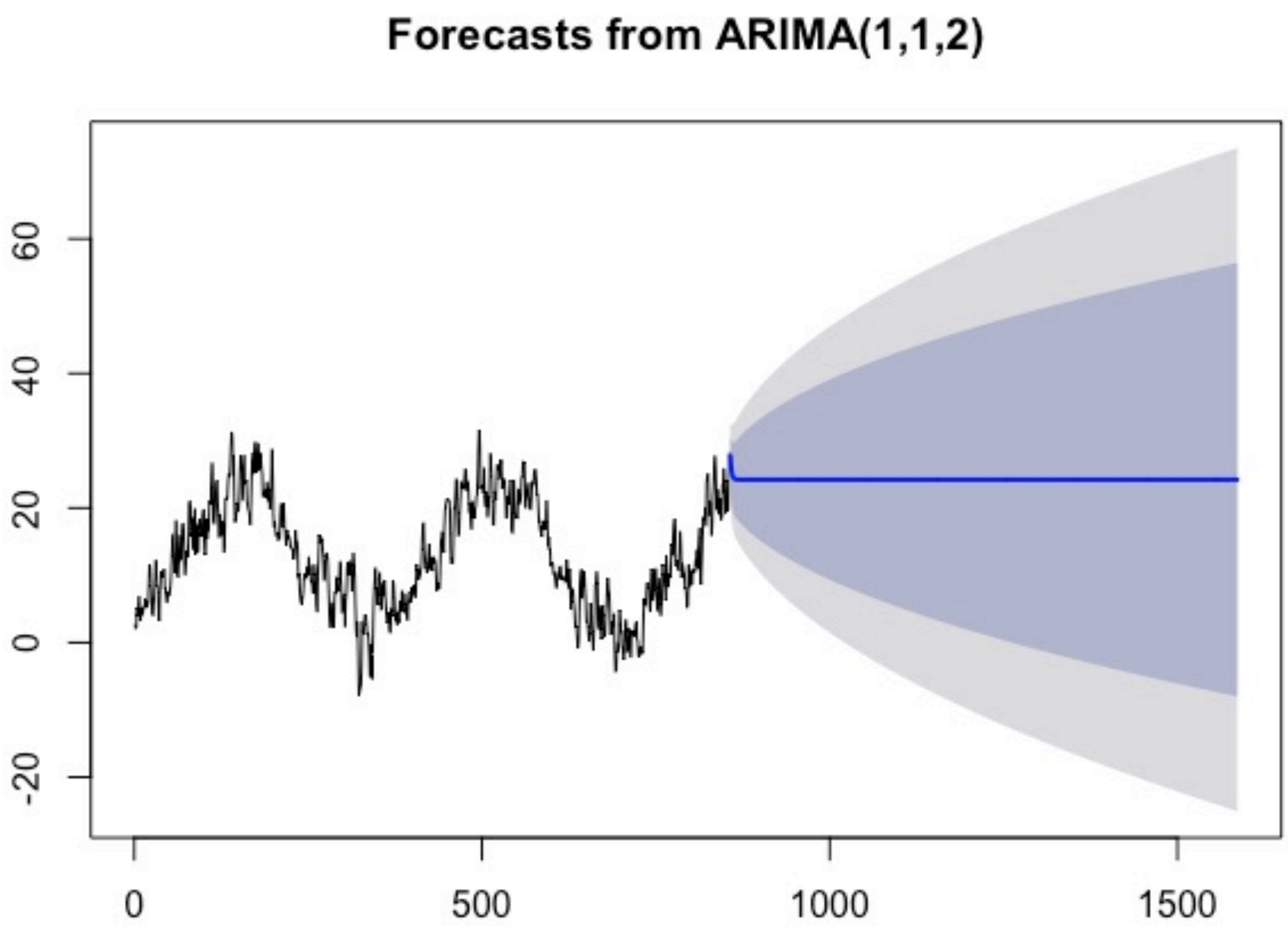
Auto

Regressive

Integrated

Moving

Average



ARIMA

- **AR**: *Autoregression*. A model that uses the dependent relationship between an observation and some number of lagged observations.

- **I**: *Integrated*. The use of differencing of raw observations (e.g. subtracting an observation from an observation at the previous time step) in order to make the time series stationary.

- **MA**: *Moving Average*. A model that uses the dependency between an observation and a residual error from a moving average model applied to lagged observations.

ARIMA

The parameters of the ARIMA model are defined as follows:

- **p**: The number of lag observations included in the model, also called the lag order. (ar)
- **d**: The number of times that the raw observations are differenced, also called the degree of differencing. (integ)
- **q**: The size of the moving average window, also called the order of moving average. (ma)

ARIMA

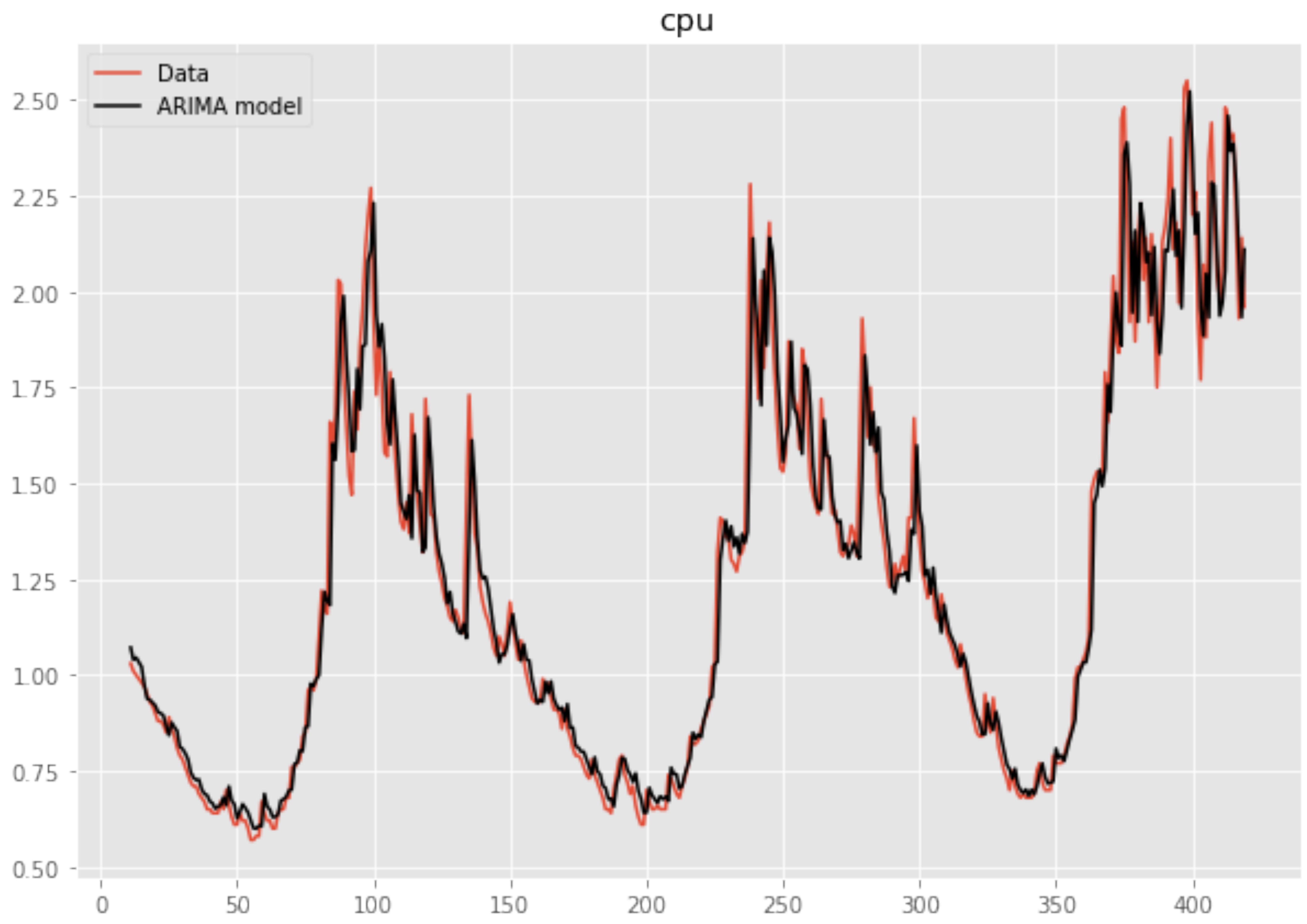
```
import pandas as pd
import pyflux as pf

# Get data & build model
model = pf.ARIMA( data = data_train, ar=11, ma=11, integ=0, target=<target_col>)
model.fit("M-H")

#Draws the original data and the projected data
model.plot_fit()
```

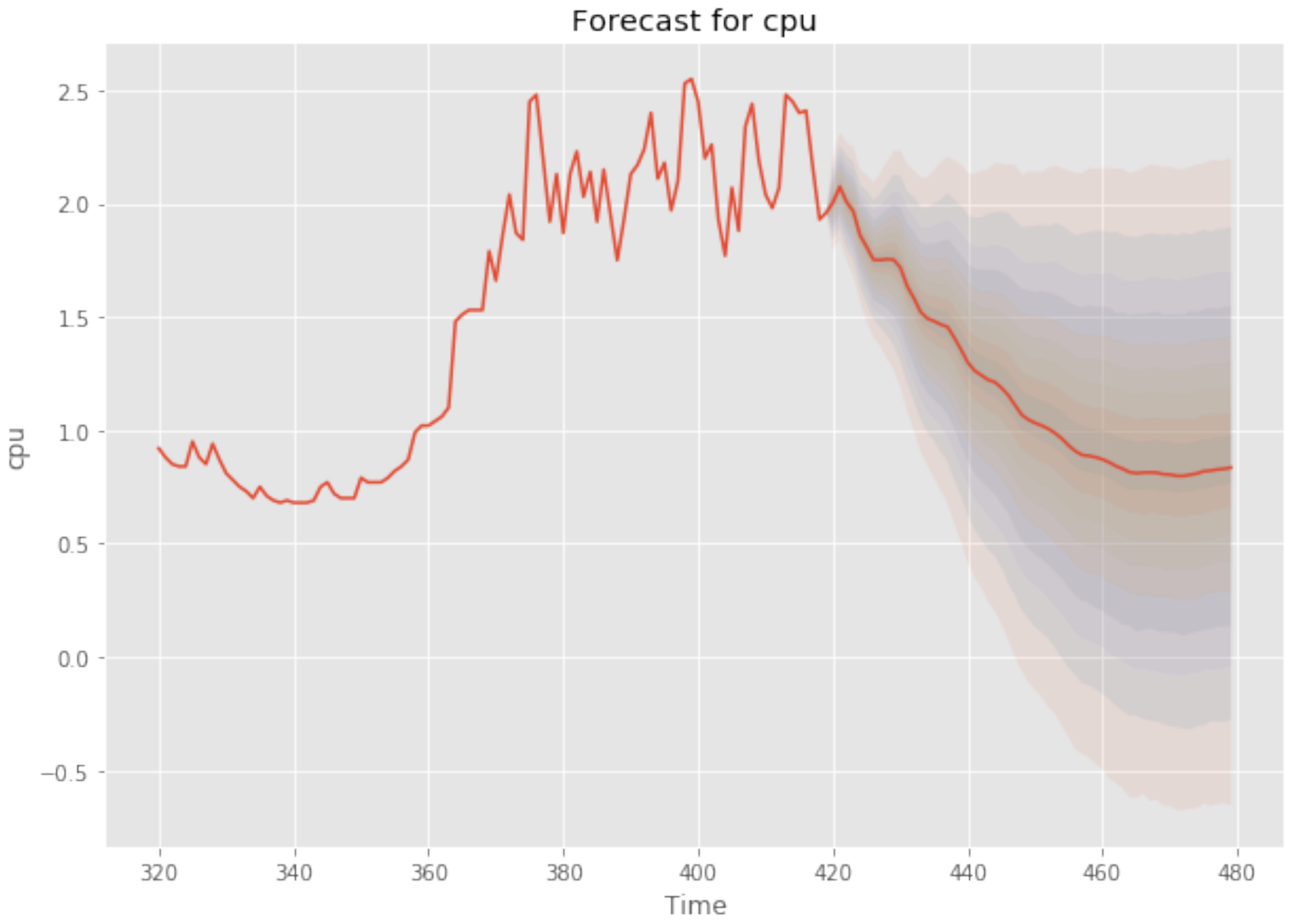
ARIMA

`model.plot_fit()`

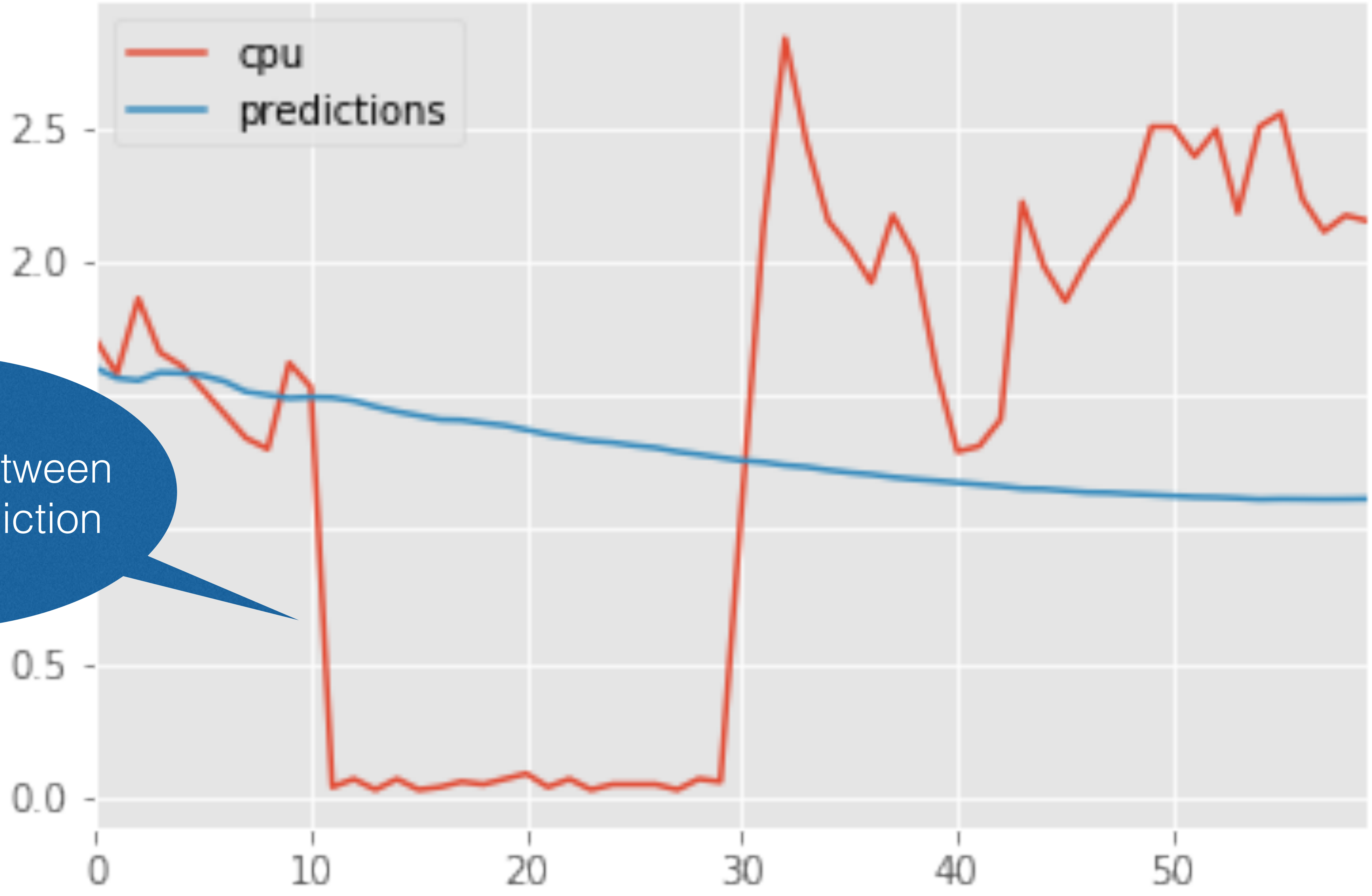


ARIMA

```
model.plot_predict(h=60, past_values=100)
```



ARIMA



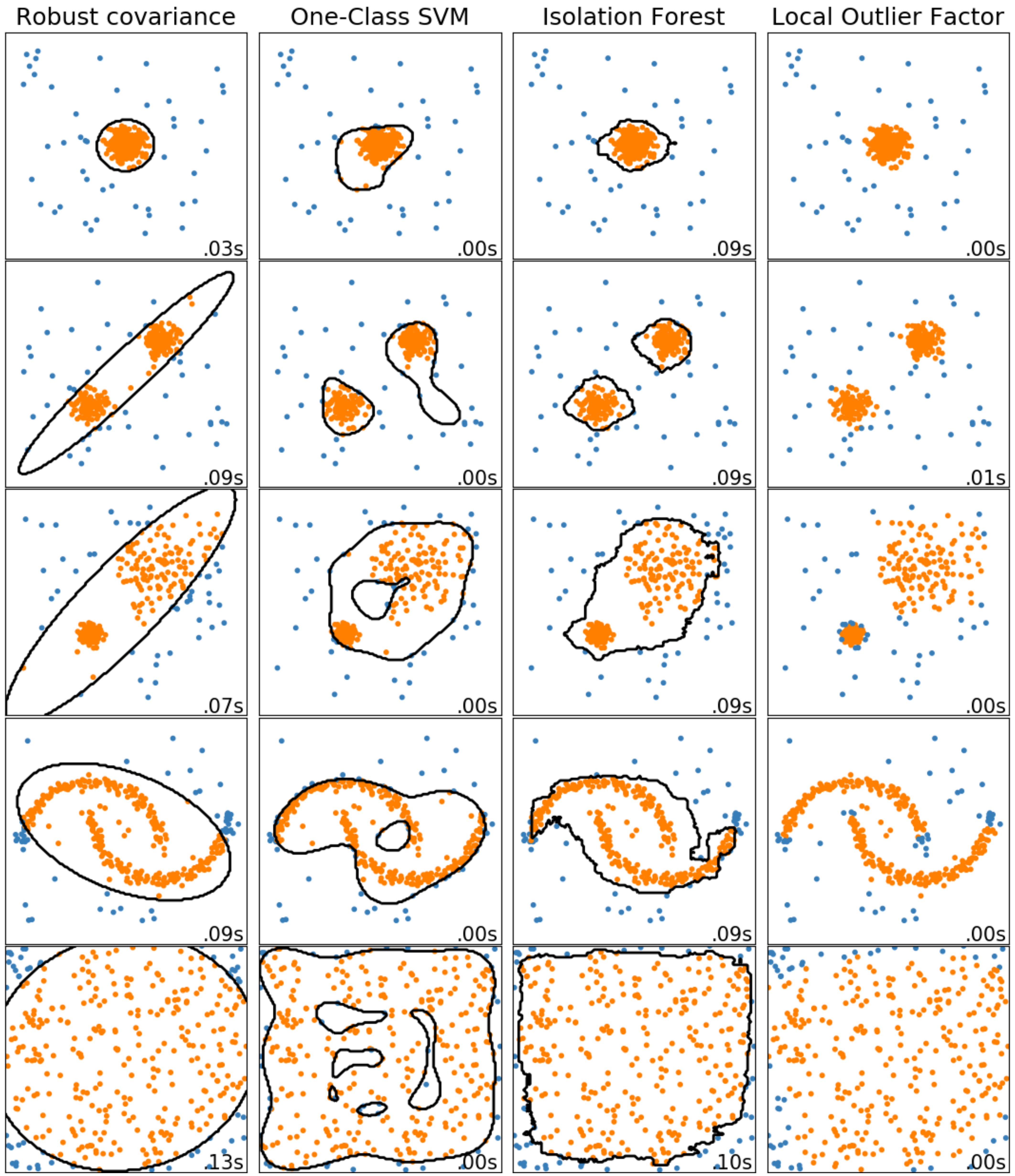
Large delta between data and prediction

ARIMA

- SARIMA: Seasonal ARIMA can be used for data with seasonally
- ARIMAX: Another variation of ARIMA, commonly used for economic forecasting.

Questions?

Unsupervised Learning Techniques for Anomaly Detection



gtkcyber.com

One Class SVM Implementation

- The one class SVM can be used for novelty detection by creating one big class.

- If the observation is a member of the class, it is not an outlier

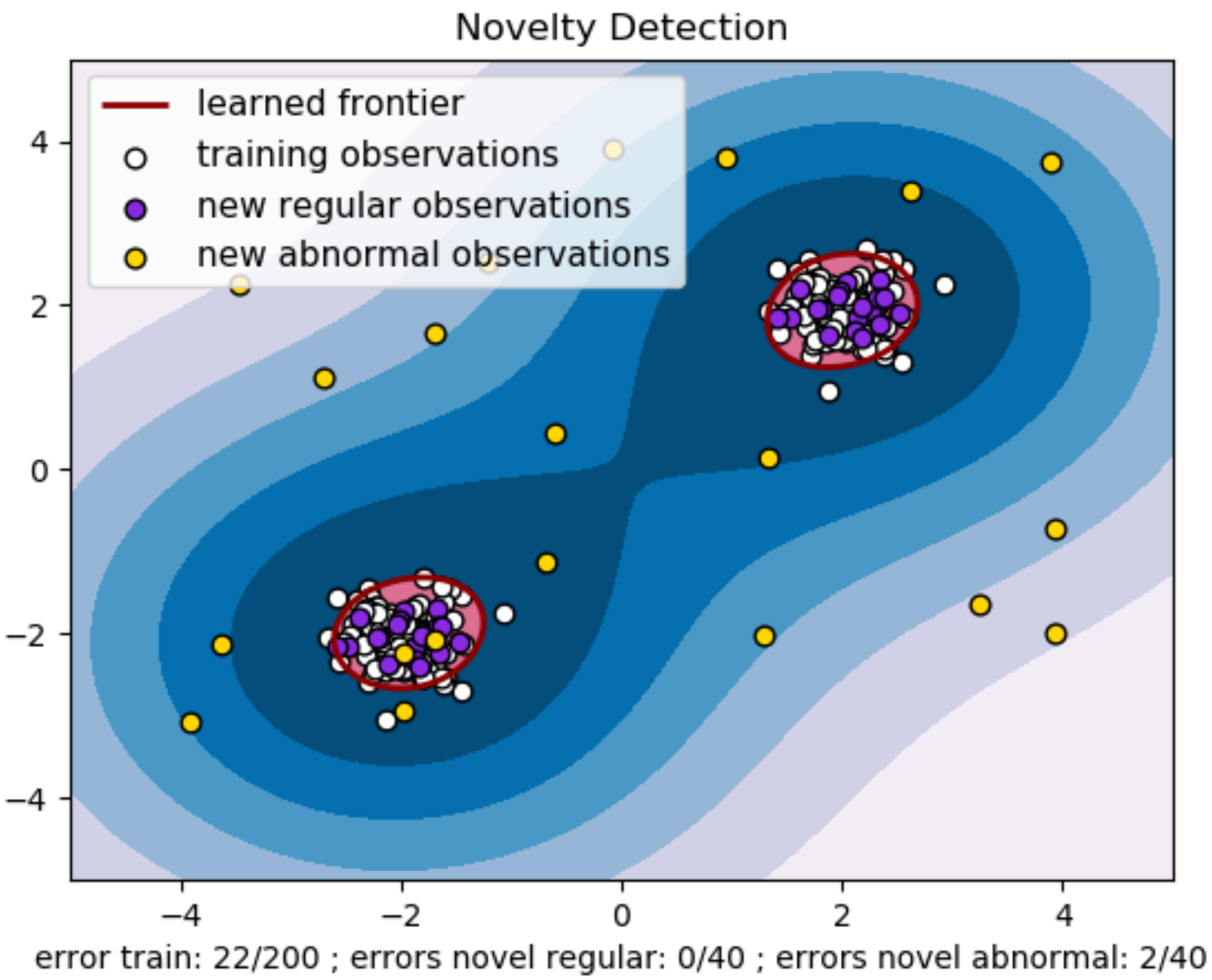
One Class SVM Implementation

```
from sklearn import svm

# fit the model
clf = svm.OneClassSVM(nu=0.1, kernel="rbf", gamma=0.1)
clf.fit(X_train)
y_pred_train = clf.predict(X_train)
```

https://scikit-learn.org/stable/auto_examples/svm/plot_oneclass.html

One Class SVM Implementation



https://scikit-learn.org/stable/auto_examples/svm/plot_oneclass.html

One Class SVM Pros

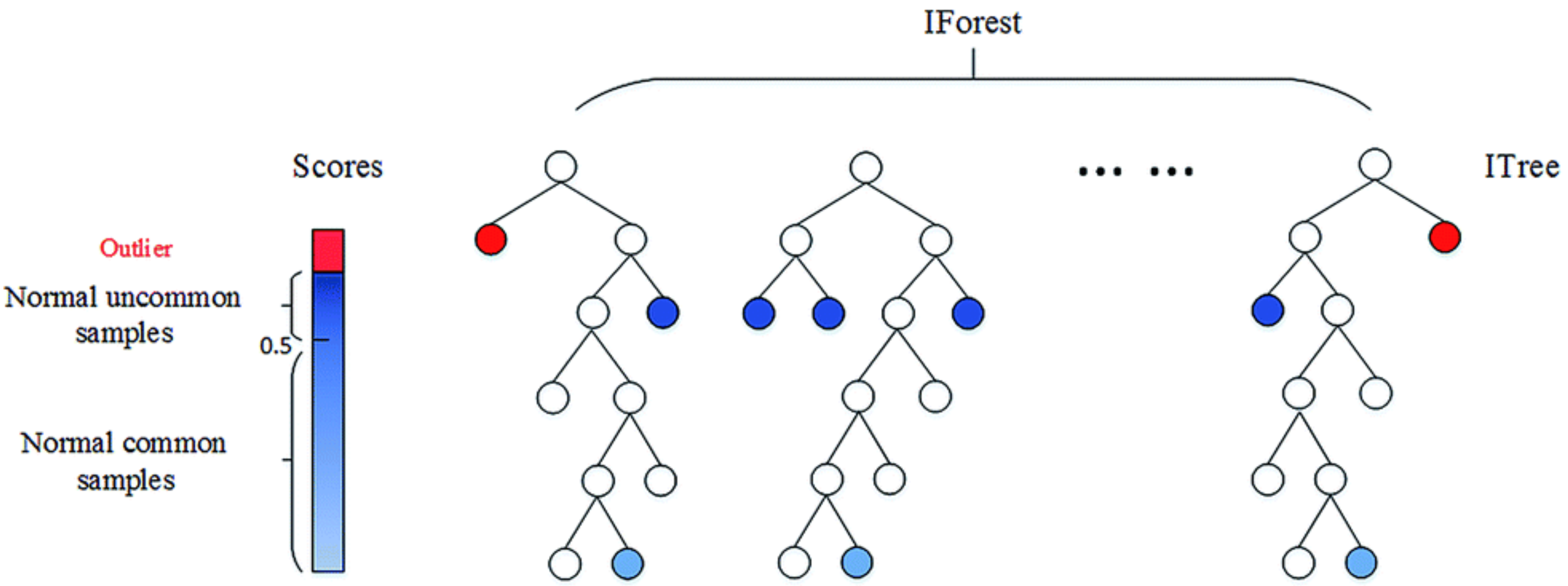
- Useful for high dimensional data sets
- Does not require any assumptions about the distribution of the data

One Class SVM Cons

- Sensitive to outliers
- Best used for novelty detection rather than anomaly detection

Isolation forest's basic principle is that outliers are few and far from the rest of the observations.

https://towardsdatascience.com/a-brief-overview-of-outlier-detection-techniques-1e0b2c19e561



https://towardsdatascience.com/a-brief-overview-of-outlier-detection-techniques-1e0b2c19e561

To Build an Isolation Forest

- Randomly pick a feature and a random split value between the max and min.

- Repeat for all observations in the training set

- A prediction is made by traversing the tree and measuring the path length. Shorter paths are likely to be outliers.

https://towardsdatascience.com/a-brief-overview-of-outlier-detection-techniques-1e0b2c19e561

Isolation Forest Advantages

- There is no need of scaling the values in the feature space.
- It is an effective method when value distributions can not be assumed.
- It has few parameters, this makes this method fairly robust and easy to optimize.
- Scikit-Learn's implementation is easy to use and the documentation is superb.

https://towardsdatascience.com/a-brief-overview-of-outlier-detection-techniques-1e0b2c19e561

Isolation Forest Disadvantages

- Visualizing results is complicated.
- If not correctly optimized, training time can be very long and computationally expensive.

https://towardsdatascience.com/a-brief-overview-of-outlier-detection-techniques-1e0b2c19e561

Isolation Forest Implementation

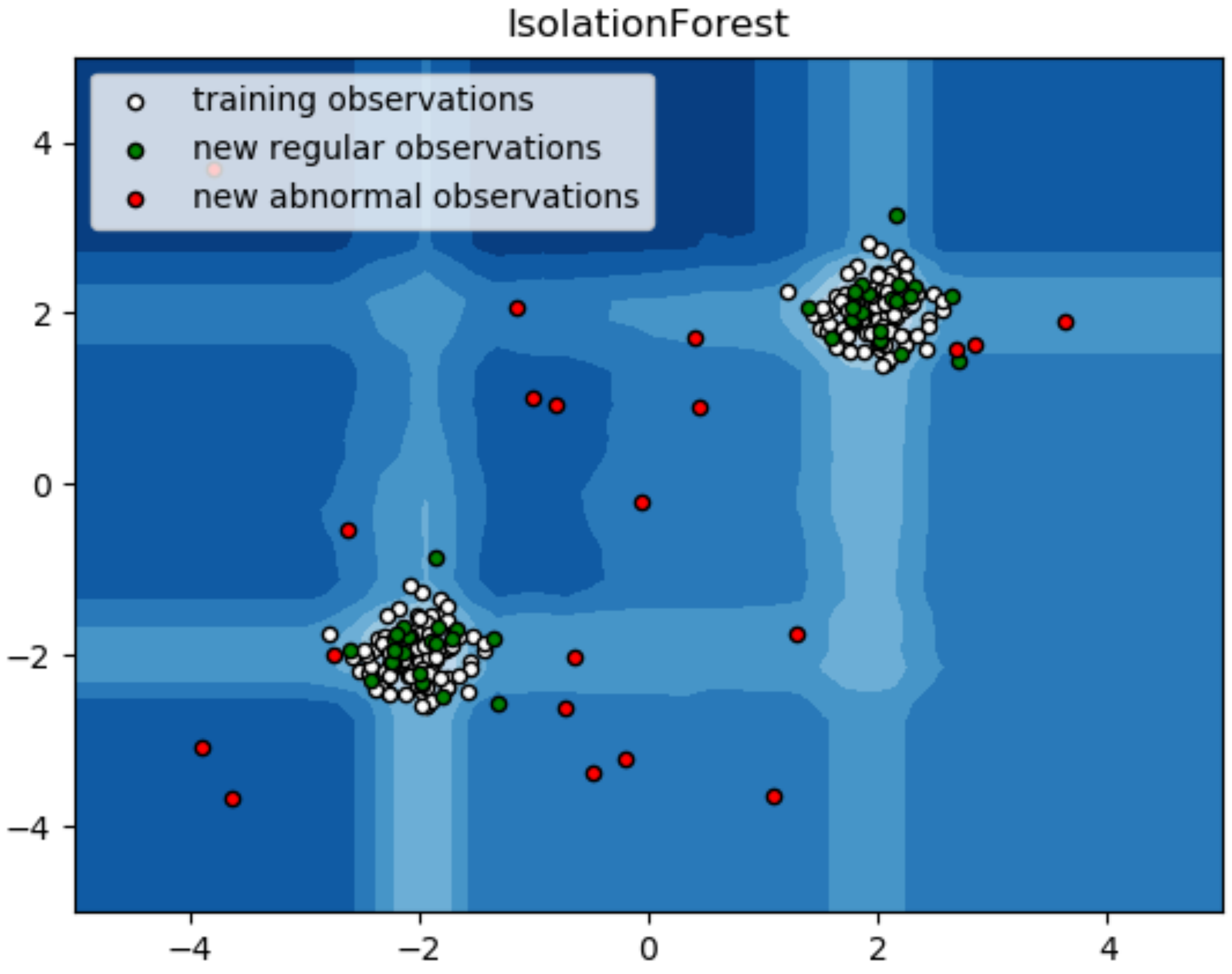
- Scikit Learn Documentation: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.IsolationForest.html

```
from sklearn.ensemble import IsolationForest

# fit the model
clf = IsolationForest(behavior='new', max_samples=100,
                      random_state=rng, contamination='auto')
clf.fit(X_train)
y_pred_train = clf.predict(X_train)
y_pred_test = clf.predict(X_test)
y_pred_outliers = clf.predict(X_outliers)
```

https://towardsdatascience.com/a-brief-overview-of-outlier-detection-techniques-1e0b2c19e561

Isolation Forest Implementation



https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.IsolationForest.html

Back to our data validation problem...

What if you could measure how similar a string is to another?

You can…

String Distance Functions

- **Cosine Similarity**: Converts strings to vectors and compares the angles of the two vectors.
- **Jaccard Distance:** Measures the dissimilarity between two sets.
- **Jaro Distance:** A similarity algorithm indicating the percentage of matched characters between two character sequences. The Jaro measure is the weighted sum of percentage of matched characters from each file and transposed characters.
- **Levenshtein Distance:** The number of changes needed to change one sequence into another, where each change is a single character modification (deletion, insertion, substitution)

Questions?

In Class Exercise

Complete Worksheet 7: Anomaly Detection.