Working with Outliers and Time Series Shocks

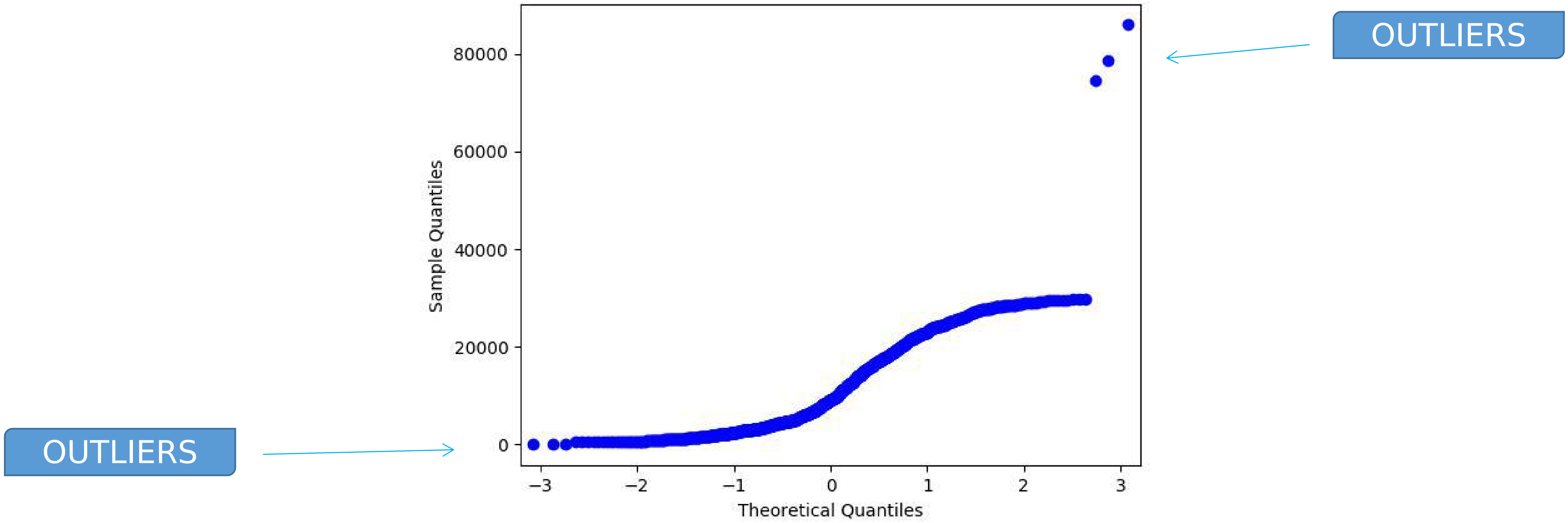
Michael Grogan

About Me

- Data Science Consultant and Educator based in Ireland (michaeljgrogan.com).
- Specialist in statistics and machine learning using Python and R
- Examples of my projects:
  - Image recognition and text classification with Keras
  - Financial time series analysis
  - ML and social media analysis

Outliers

Feel free to ask over the Slido app using **#bigdata2018** :)



Example

- **Consider the following numbers: 5, 8, 10, 10, 15, 20**

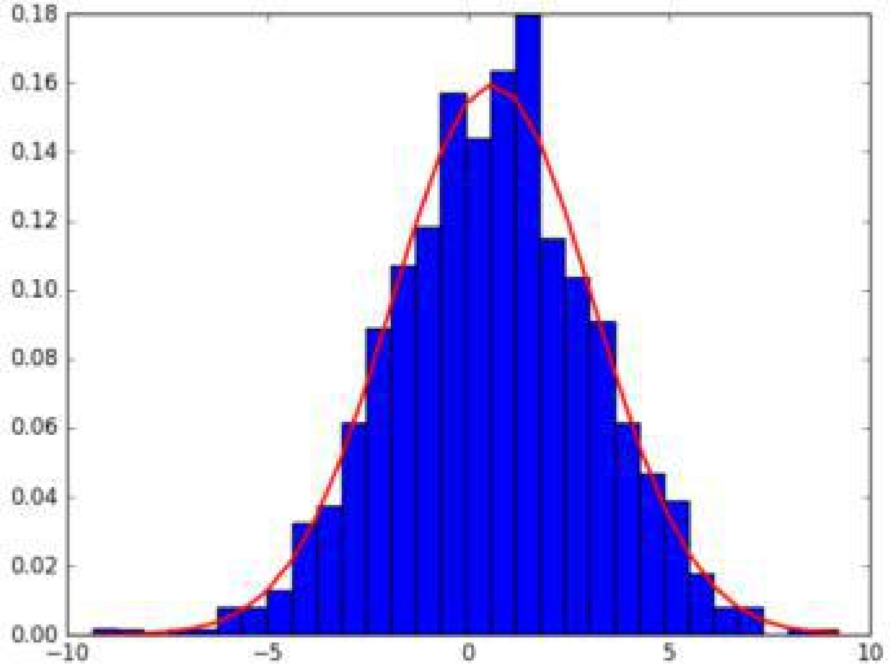
- Mean = 11.33
- Standard Deviation = 5.35

- **Now, we add a number to the array: 5, 8, 10, 10, 15, 20, 1000**

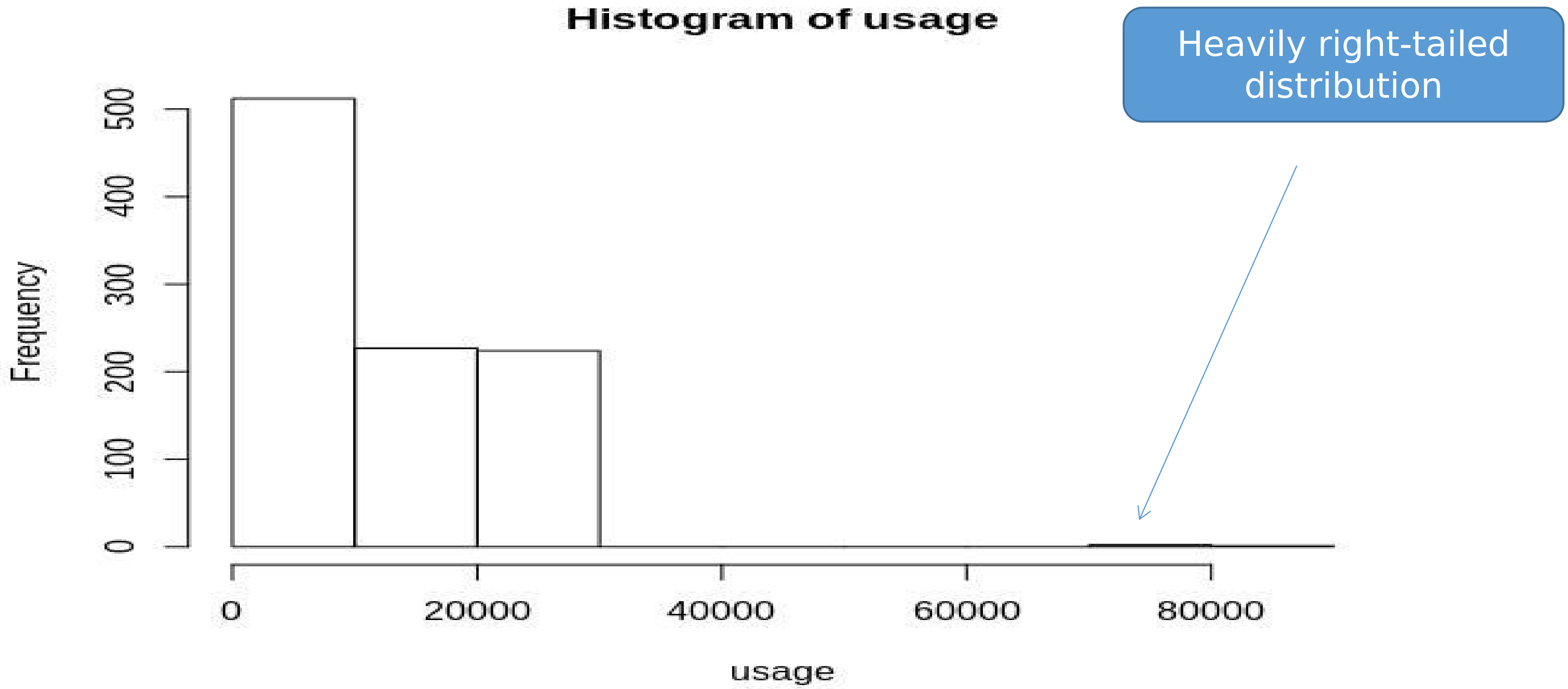
- Mean = 152.57
- Standard Deviation = 373.71

What is a normal distribution?

1. Mean = Median = Mode
2. Follows the Empirical Rule (68 - 95 - 99.7)
3. Shape of bell curve is symmetric



A skewed distribution



Heavily right-tailed distribution

Is a normal distribution assumption important?

1. Technically speaking, we do not NEED a normal distribution for regression analysis.
2. We simply require a BLUE Estimator (Best Linear Unbiased Estimator).
3. However, outliers can significantly change the shape of our distribution, and hence our overall results.

Effects of Outliers

1. Skewing of mean and standard deviation
2. Could significantly affect significance readings when generating regression analysis
3. Can give us false readings on the magnitude of correlations

Dilemma of outliers

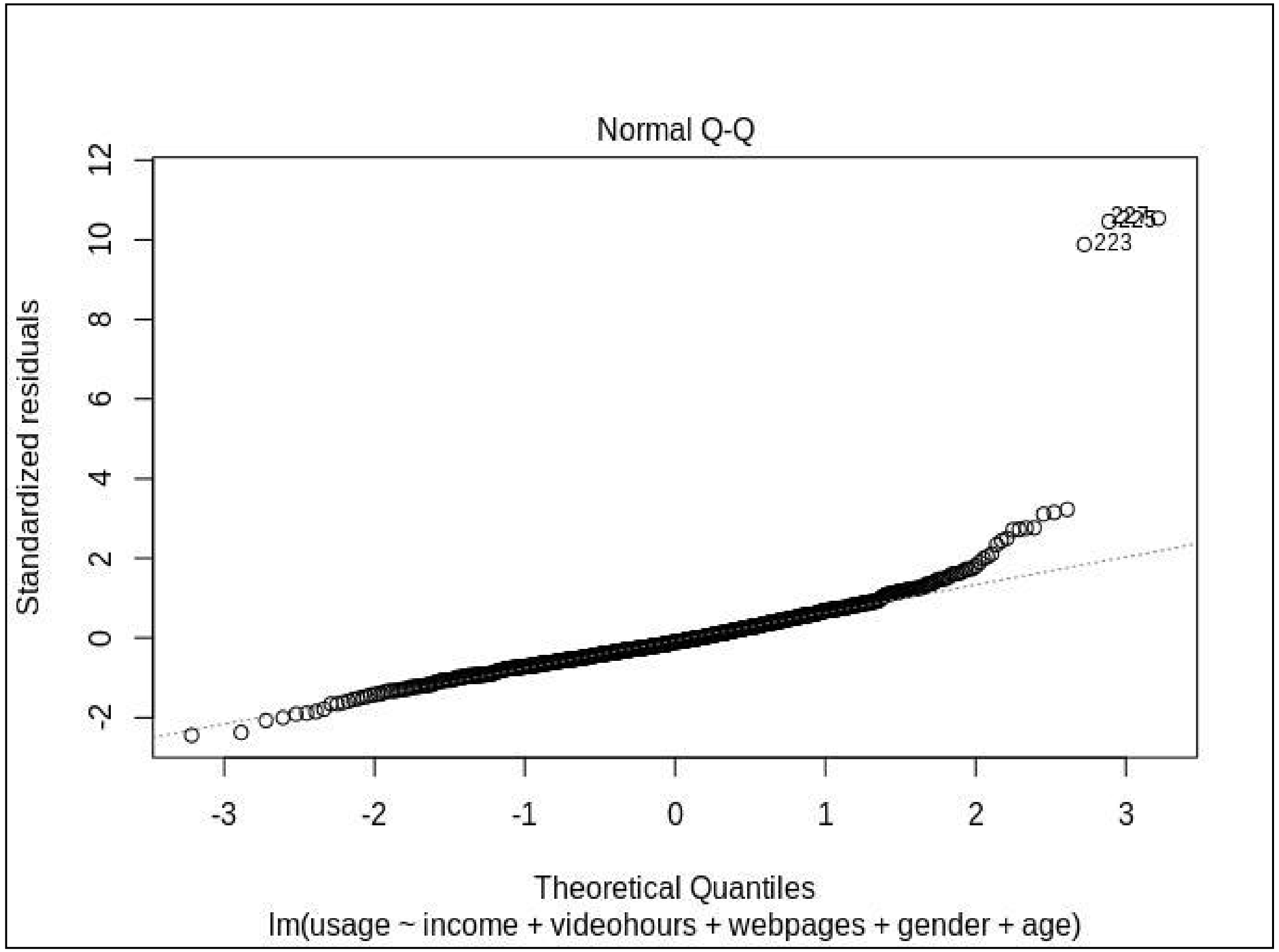
- A common consideration when dealing with outliers is whether to:
    - Remove the outliers
    - Normalize all data
    - Keep the outliers
- All of these scenarios risk significantly skewing the regression results.
- What if there was another solution?

How do we deal with outliers?

- A key method for dealing with outliers is through the use of a **weighting mechanism**.
- This is one whereby weights of observations are adjusted so as to append less weight to extreme values.
- The two types of weighted regressions we will examine are:
  - Huber regression
  - Bisquare regression

OLS (Linear) Regression

- The purpose of an OLS regression is to minimise the sum of squared residuals.
- Of course, this is much more difficult if significant outliers are present.
- Least squares are too sensitive to extreme values.



Robust Regression

- Outliers with extreme values are measured in terms of **leverage**, i.e. if the difference between a variable and its mean is extreme, then it has high leverage.
- Specifically, a robust regression adjusts the weights to give less emphasis on extreme values.
- This process is known as **Iteratively Reweighted Least Squares (IRLS)**.
- Robustness is defined as insensitivity to small deviations from the mean.

M-Estimation

- Generalisation of the maximum likelihood estimation.
- Allows for an estimation that is not as sensitive as least squares to unusual data.

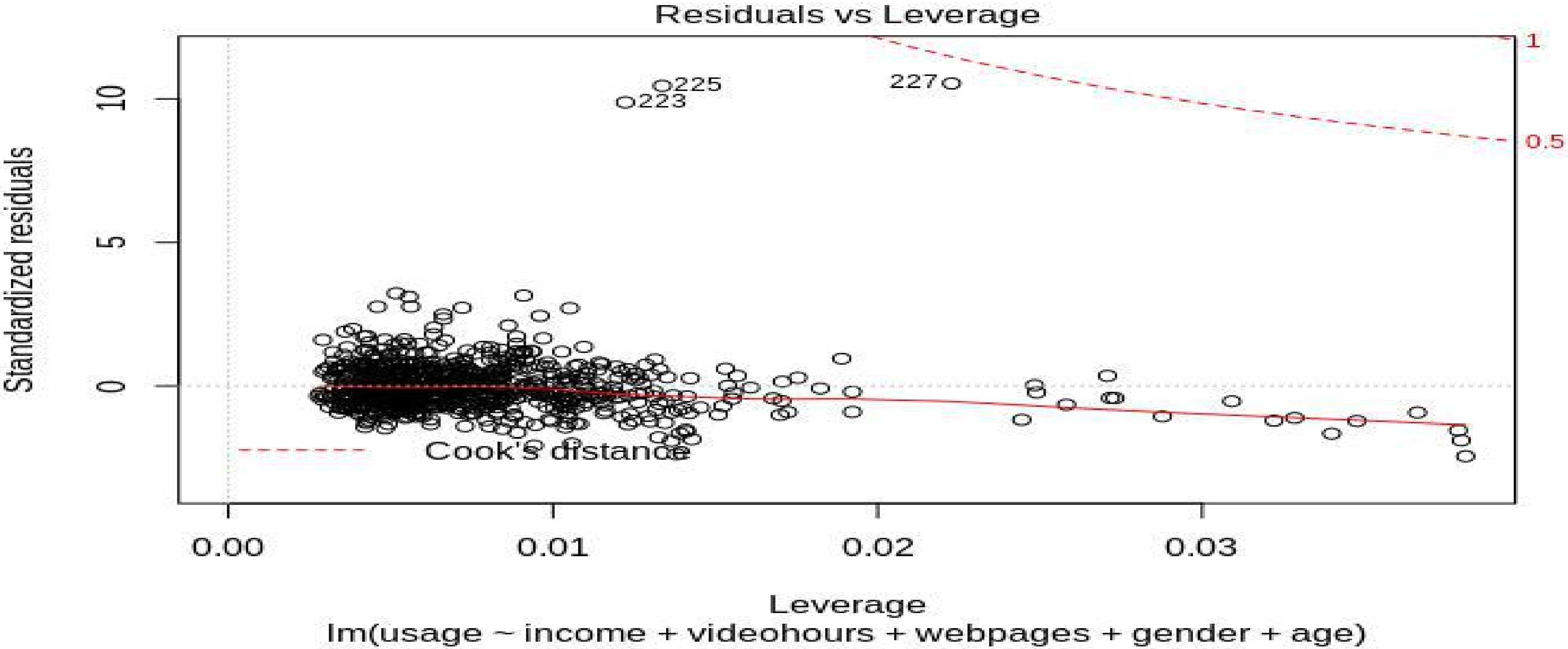
$$\sum_{i=1}^{n} \rho(\mathrm{r_i})$$

- The purpose of the M-Estimator is to minimise rho, as related to the likelihood function for the assumed residual distribution.

Cook's Distance

- How do we know whether an observation can actually be classed as an outlier?
- One useful way of doing this is using **Cook's distance**.
- As a rule of thumb, a Cook's distance three times greater than the mean indicates the presence of an outlier and warrants further investigation.

Cook's Distance



Our Case Study

- A hypothetical telecommunications firm is using regression analysis to determine internet consumption in megabytes among potential customers.
- However, the presence of significant outliers in the dataset could skew the results and reduce model accuracy.
- Weighting techniques are used to deal with this issue.

Huber vs. Bisquare Weighting

```
> huber2[1:10, ]
    usage    resid     weight
227 85970 53688.27 0.08414706
225 78774 52411.19 0.08619675
223 74561 50294.80 0.08982477
230 24433 15972.44 0.28283568
422 17843 15711.29 0.28754642
93  27213 15371.04 0.29389690
710 29626 14056.28 0.32139719
571 25973 13950.78 0.32382460
584 23903 13307.63 0.33946565
516 25182 13103.97 0.34473723
```

```
> bisqr2[1:10, ]
    usage    resid       weight
223 74561 50521.41 0.000000e+00
225 78774 52483.16 0.000000e+00
227 85970 54021.03 0.000000e+00
230 24433 16110.28 0.000000e+00
422 17843 15760.19 1.187774e-10
93  27213 15542.68 7.518861e-04
710 29626 14206.90 3.513518e-02
571 25973 14123.85 3.877236e-02
584 23903 13454.75 7.353427e-02
516 25182 13255.17 8.562033e-02
```

Bisquare appending much less weight to values significantly outside the mean.

Accuracy across regression models

- **OLS:** 70.89%
- **Huber:** 76.05%
- **Bisquare:** 77.34%

What if we wish to directly control sensitivity of the model to outliers?

```
class sklearn.linear_model.HuberRegressor(epsilon=1.35, max_iter=100, alpha=0.0001,
warm_start=False, fit_intercept=True, tol=1e-05)
```

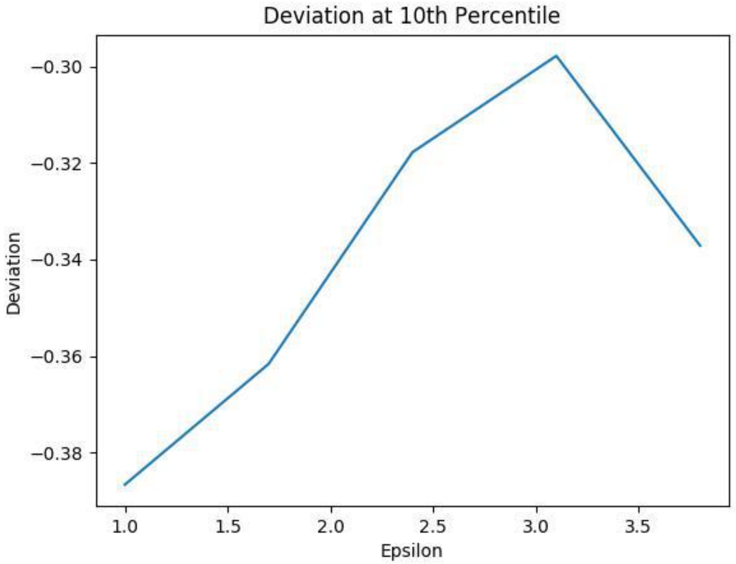
- Using Python, we can set the epsilon value (default = 1.35) in order to set the number of samples that the model should classify as outliers.
- The smaller the epsilon value, the more robust the model is to outliers.
- If an outlier is not particularly extreme, then a higher epsilon value might be more desirable depending on the percentile being examined. Let's look at another example.

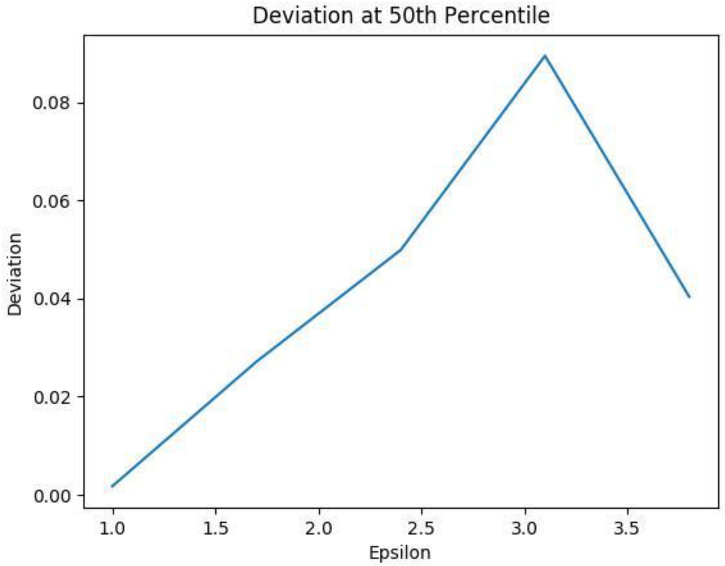
Deviations

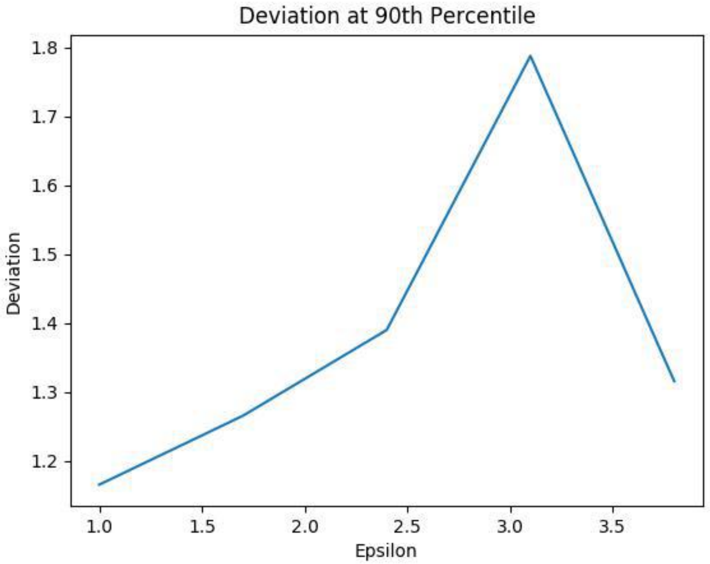
1oth percentile

50th percentile

90th percentile





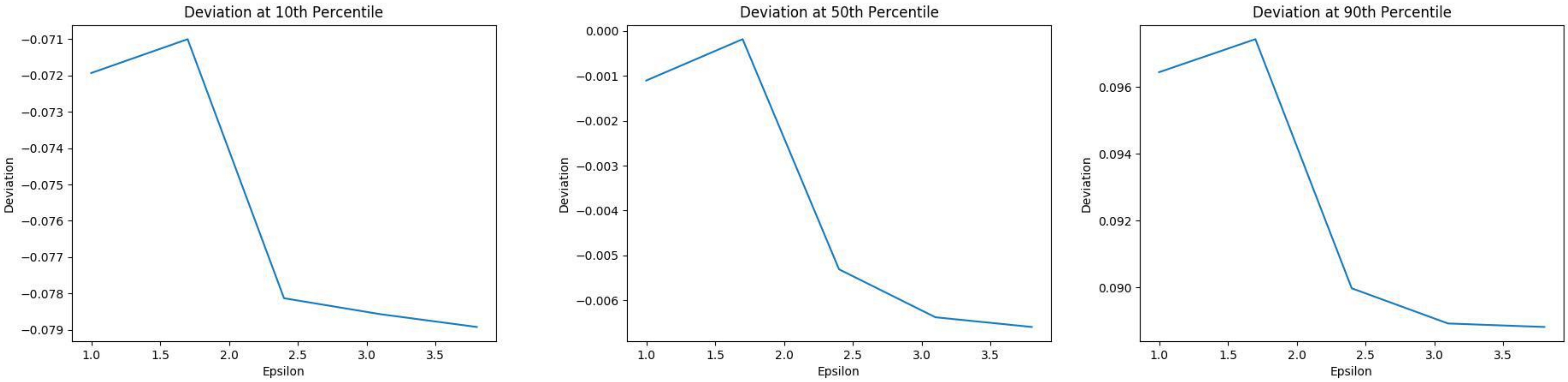


What about deviations in the case of no outliers?

1oth percentile

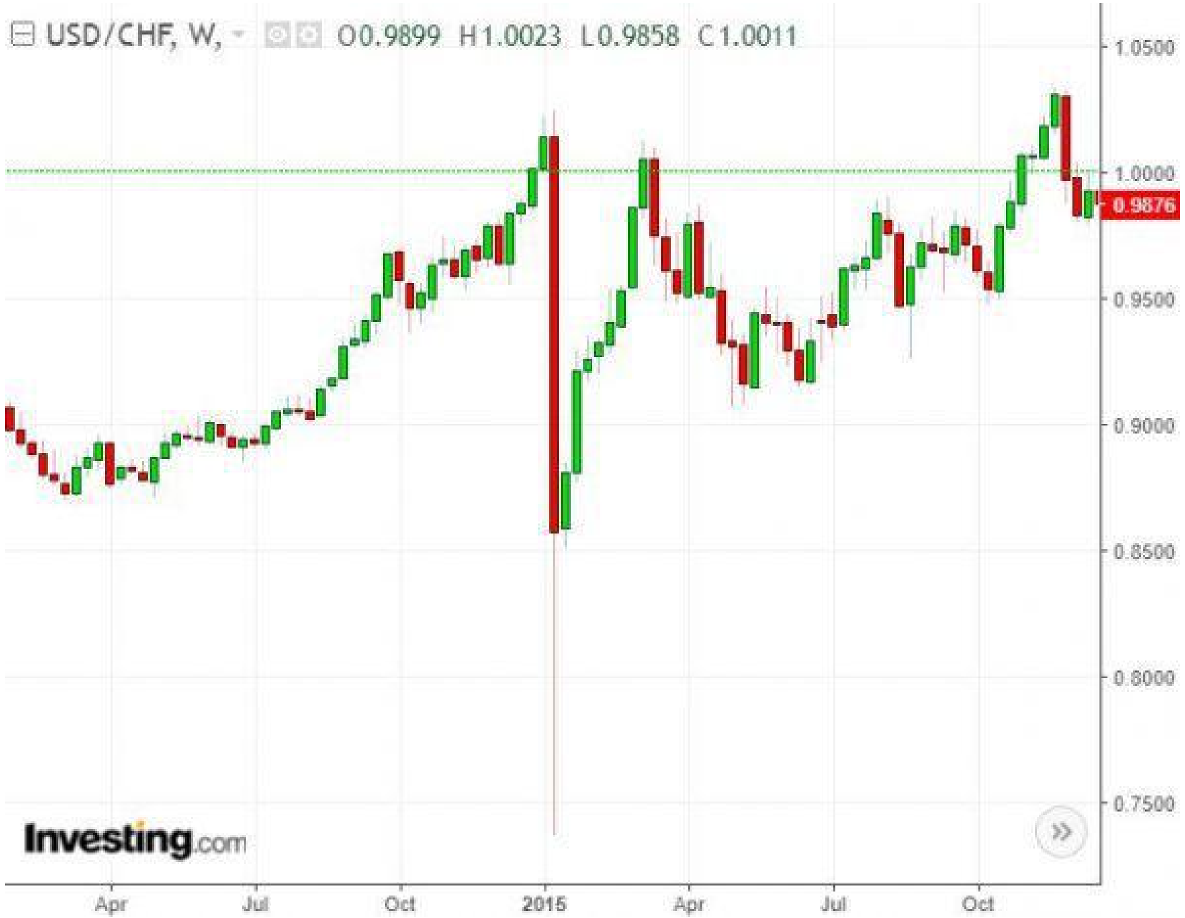
50th percentile

90th percentile



Time Series Shocks

- It is often the case that a time series can deviate from a trend quite rapidly.
- For instance, remember in January 2015 when the dollar plunged against the Swiss franc?



Time Series Shocks

- It is often the case that traditional time series models such as ARIMA fail to account for the effects of the shock sufficiently.
- Therefore, it is necessary to use what is called a **Kalman Filter** in order to adjust for such a shock.

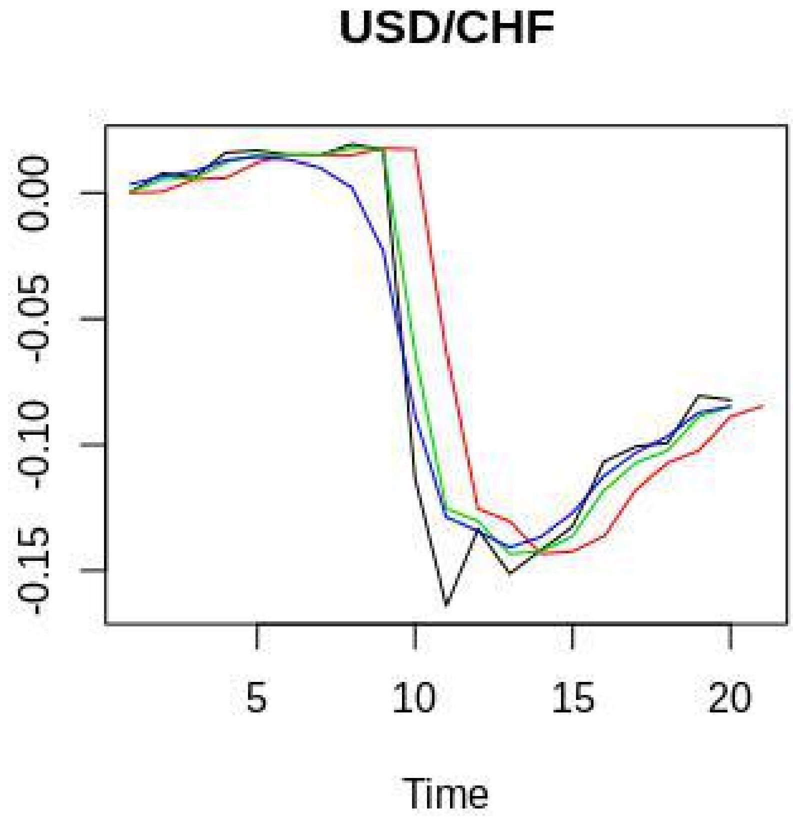
What Is A Kalman Filter?

- The Kalman Filter is a state-space model that adjusts more quickly for shocks to a time series.
- There are three particular variables we are looking to examine:
  - a: One-step-ahead predictions of states
  - att: Filtered estimates of states
  - alphahat: Smoothed estimates of states

USD/CHF

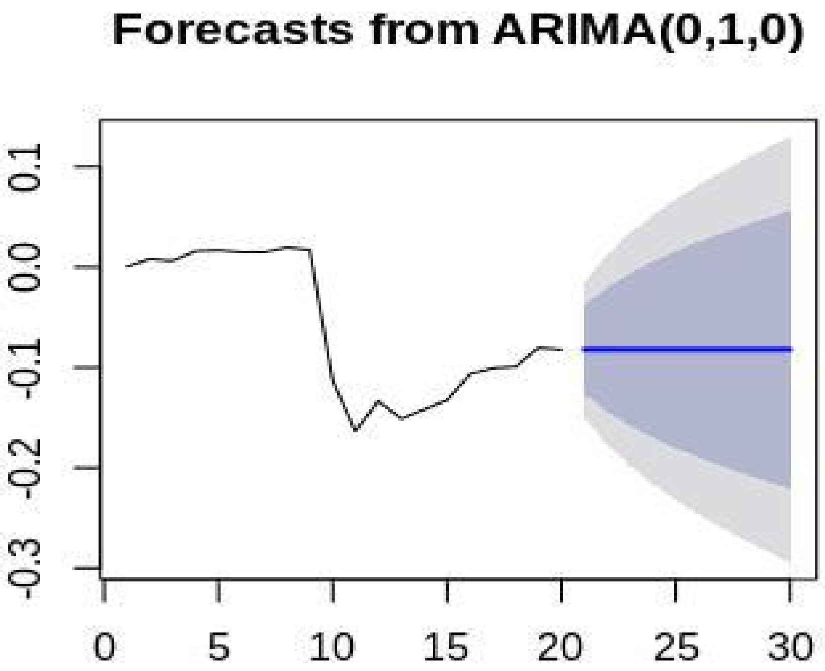
```
logmodel <- SSModel(usdchf ~ SSMtrend(1, Q = 0.01), H = 0.01)
```

- In this instance, we can see that the Kalman Filter is adjusting to the shock immediately.
- Ultimately, the filter is allowing the state to evolve over time.
- Q and H denote the unconstrained time-invariant covariance estimates.

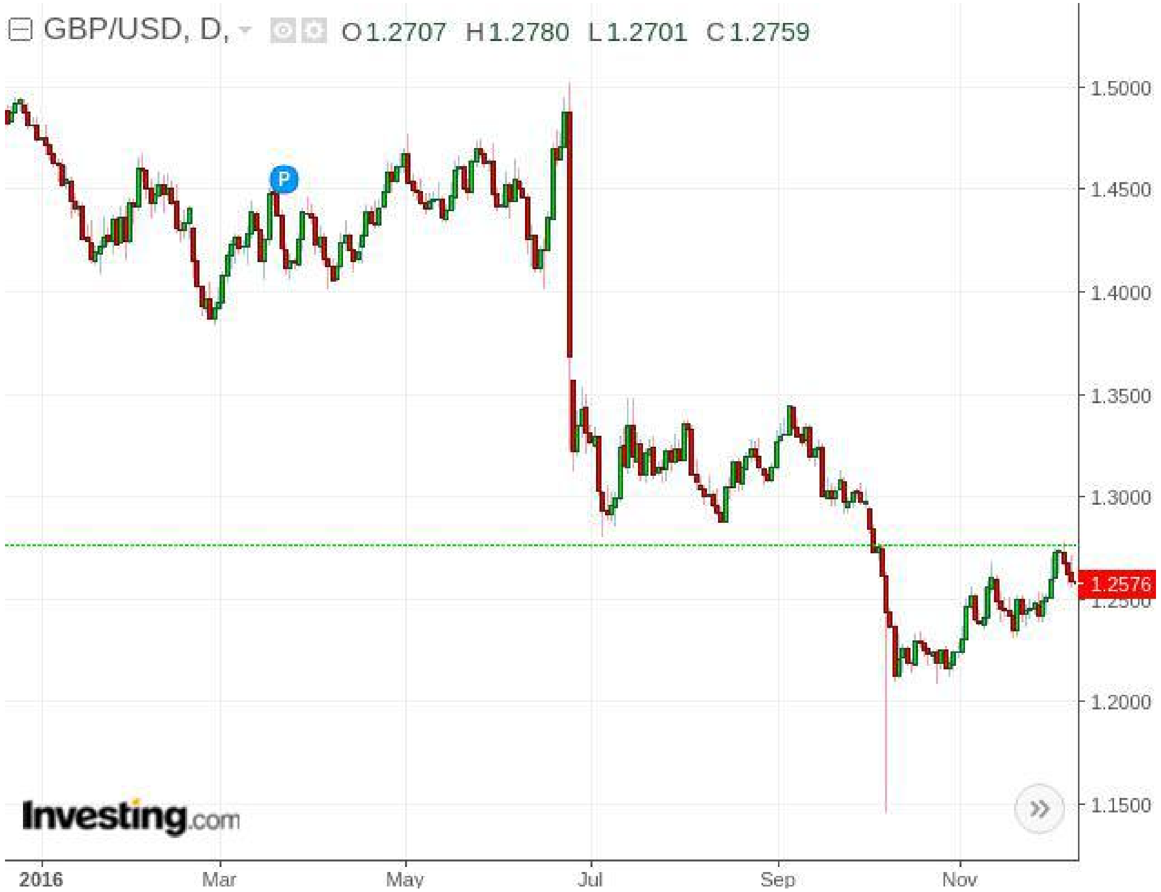


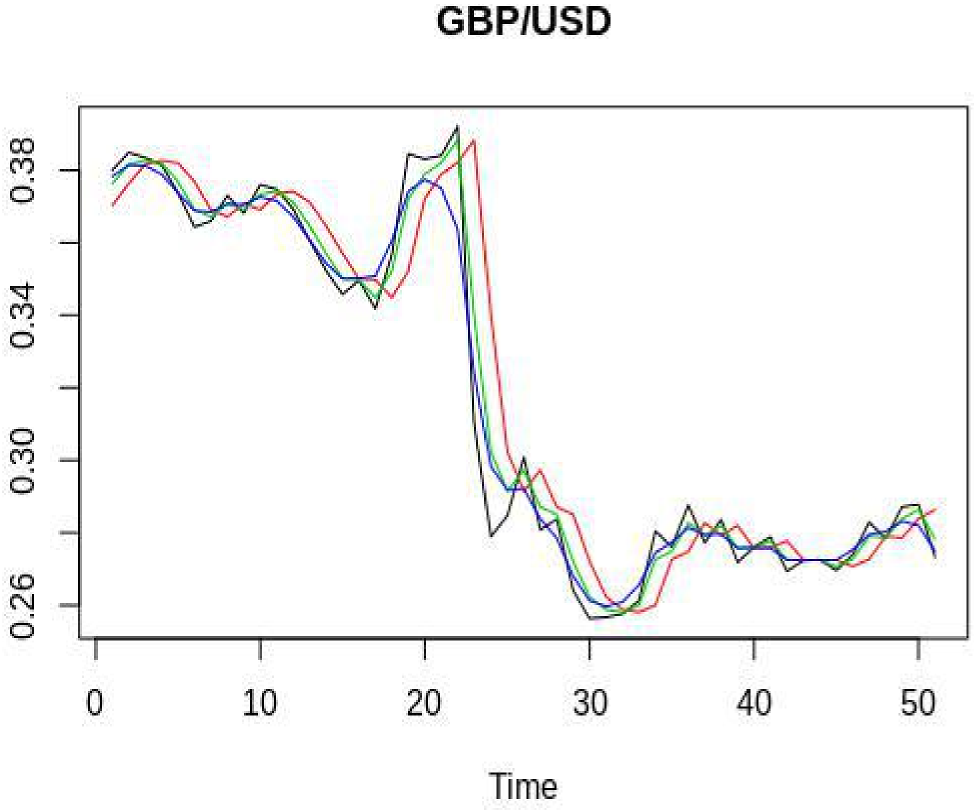
How does ARIMA compare?

- In this particular instance, ARIMA does not do a good job at prediction.
- When confronted with a time series shock, the model forecast simply becomes too wide for any meaningful prediction.



Another example of the Kalman Filter





Kalman Filters are not just for time series!

- The Kalman Filter is ultimately a great model when it comes to modelling **noisy systems.**

- Some examples of these could be:
  - Navigation modelling (e.g. autonomous vehicles)
  - Computer vision applications
  - Prediction of other noisy states (e.g. weather)

Conclusion

- In this presentation, we have seen:

  1. How outliers can hinder effective data analysis
  2. The use of weighting mechanisms in mitigating the effects of outliers
  3. How to screen accuracy of weighted regressions as compared with least squares
  4. Use of the Kalman Filter in adjusting for time series shocks

Questions?

- Happy to take any questions at this point!