Spurious relationships in Twitter data

Data Science in Techno-Socio-Economic Systems

Spring Semester 2016

Claudio Bruderer
cbruderer@phys.ethz.ch

Andrina Nicola
nicolaa@phys.ethz.ch

Anna Weigel
anweigel@phys.ethz.ch

Abstract

One of the main goals in data analysis of any sort, is detecting and quantifying relations between quantities. With the increasing amount of data available to us, this task becomes more challenging, since large datasets are prone to spurious correlations. It is therefore important to identify these spurious correlations in order to be able to distinguish them from causal relationships. In this project search for spurious correlations by querying twitter data for a list of pre-defined keywords and computing the correlation between keyword pairs from the time series of the keyword occurrence frequencies. We compute correlations using two separate methods and find that both methods detect both expected as well as spurious correlations. These results show that for large datasets, we are easily inclined to over-interpret the results and that conventional correlation measures need to be well understood and carefully applied.
Our results are also available online: http://spuriousrelationships.eu.pn/
All scripts (including LaTeX files) used in this analysis are public and can be found here: https://github.com/cbruderer/spuriousrelations/

1 Introduction

We have entered the era of Big Data and the wealth of information available to us entails its own new challenges. Relationships between quantities, whether real or without any

obvious causal link, will naturally arise if the data sets are large enough (for examples see [1]). Even though we might be inclined to interpret apparent links between measurements, a correlation does not imply causation. One needs to find a way to distinguish spurious relationships between measurements from those with a causal link, and quantify the frequency of chance correlations occurring.

Based on a list of pre-defined keywords and Twitter data, we aim to show that relationships between quantities without any obvious causal link can be identified in large data sets. We use a Twitter dataset which contains unfiltered tweets between May 2014 - July 2014 and September 2014 - December 2014. For each day, we determine the number of tweets containing one of our pre-defined keywords. We then measure and quantify the time-correlation functions between the popularity of various keywords. To test our method we recover and quantify expected correlations between synonyms or related keywords. We additionally investigate and discuss the performance of different correlation measures.

2 Data

2.1 Data set

In order to identify relationships between words, we need a suitable data set containing a large amount of text data and covering an extended period of time. Data from all major social media platforms primarily using text as a medium (e.g. Twitter [3], Facebook [4], and Reddit [5]) satisfy these conditions.

We decided to apply our analysis on Twitter data, which has been previously gathered by an ETH research group. This data set was created via the public Twitter stream [6], which downloads 1% of the tweets posted at that moment without applying any further filtering, and covers the years 2012-2014. We analyse a subset of the whole data set, which contains data of the months May, 2014 - July, 2014 and September, 2014 - December, 2014. These data have been converted to Parquet files, thus simplyfing reading in the data.

2.2 Keyword selection

Table 1: Selected keywords

Terrorism / Religion	Refugees	Ebola	Influenza	Science
isis	syria	ebola	influenza	bitcoin
terrorism	refugees	guinea	flu	rosetta
arab	migrants	sierra	birds	comet
spring	africa	leone	swine	higgs
attack	italy	liberia	pig	climate
god	ethiopia	virus		doomsday
christian	asylum	epidemic		maya
allah	unhcr	vaccine		curiosity
islam	immigration	who		sandy
	foreigners			hurricane
Discrimination	**Countries / Cities**	**Technology**	**Everyday life**	**Food / Trends**
black	europe	apple	homework	vegan
white	usa	linux	television	gluten
croweded	philippines	pc	coffee	vegetarian
left	sochi	google	tea	meat
right	olympics	iphone	school	pasta
	geneva	galaxy	work	banana
	boston	watch	teacher	
	london	facebook	sports	
		twitter	jogging	
		whatsapp	marathon	
Malaysian Airlines	**Ukraine / Crimea**	**Politics**	**Family**	
mh17	ukraine	snowden	family	
mh370	crimea	nsa	divorce	
	russia	obama	marriage	
		putin	wedding	
		mandela	holidays	
		nelson		
		pope		
		unemployment		

In our analysis we select one hundred keywords. All these keywords are listed in Table 1. To choose these keywords, we apply different criteria. First, we identify major events

that happened during the period for which we have access to twitter data (May 2014 - July 2014 and September 2014 - December 2014). For these topics, we then select related words, and deliberately including synonyms as well. Besides these 'major events categories' we also choose words related to long-term trends we *a priori* expect to find in the data (e.g. 'Food / Trends', 'Refugees', 'Politics'). Furthermore, we include a group of keywords on 'Everyday life' for which we *a priori* do not expect to see strong trends.

These selection criteria are chosen to address various problems. First, by picking words related to major events localized in time (e.g. Rosetta satellite [7] dispatching Philae lander), we are able to test our pipeline and assert that this *a priori* expected quick rise in mentions on social media platforms is identified in our data as well. Second, by including several synonyms, it is reasonable to expect certain correlations between the keywords, hence allowing us to validate our analysis in a second, independent way. Third, the keywords for which we do not *a priori* expect the number of mentions to display a significant structure over time (e.g. 'Everyday life'), should only weakly correlate with other keywords displaying more structure.

We note however that there are caveats in this type of analysis. Many words have different meanings and could thus potentially correlate or not correlate with other words depending on which meaning is more common. As an example, we include 'who' in our analysis. While it is primarily used as a pronoun, it is also the abbreviation of the World Health Organization [8]. Due to the Ebola outbreak in Western Africa in 2014, we expect this secondary meaning to modulate the number of mentions over time, and potentially display correlations with related words (e.g. 'ebola'). Other examples of words with multiple meanings include 'right' (opposite of wrong; political perspective; direction) and 'left' (past participle of leave; political perspective; direction).

2.3 Data analysis

We use the PYTHON front end of Apache SPARK[9] called PYSPARK to analyse the twitter data. The data set is read in and then filtered to contain only non-empty tweets. As a next step, the timestamp is adjusted to be in a more accessible format, which encodes only the day of the year and the year, and the tweets are split up into individual words and converted to be only lowercase. For all words we compute jointly the number of mentions per day. As a last step, we filter to only recover the statistics for our keywords, where we also take the keywords into account if they are used as a hashtags, and save the results.

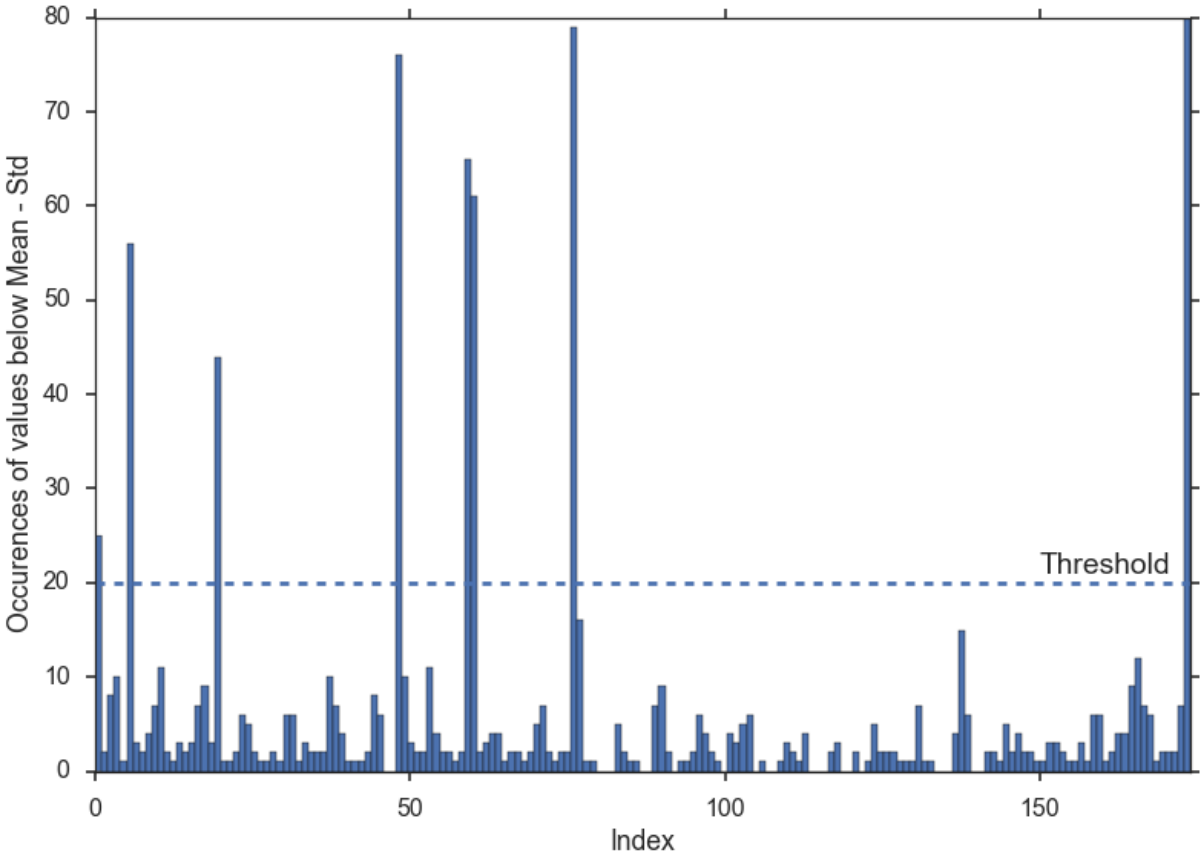


Figure 1: Histogram of the occurences of specific indices, which correspond to individual days, for which the number of mentions of a keyword is below the average minus the standard deviation for this keyword. The threshold is set at 20.

2.4 Data completeness

The streaming of Twitter data was interrupted several times during the period the data were collected. Thus, minutes, hours, or even days are missing. While missing days are trivial to identify, doing so for partly incomplete daily data is more complicated. These days however would induce systematic biases to the analysis, as for all keywords the number of mentions decreases at the same time, thus inducing artificial strong positive correlations.

Without access to statistics on the streaming of the data, only an empirical correction on the reduced data set is possible. We proceed in the following way, we compute the mean and the standard deviation for the number of mentions over time for each keyword. We note the indices, which correspond to individual days, in these time series for which

$$\#\text{Mentions} < \text{Mean} - 1 \cdot \text{Std}, \tag{1}$$

and compare these indices among the keywords. We then set a threshold and consider the

days incomplete, for which the number of occurences of the keyword is below the mean and standard deviation for at least 20 keywords. This process is illustrated in Figure 1.

In order not to bias our analysis by, we remove the occurences of the keywords for these days then from our analysis and treat them as days for which we do not have any tweets (see Section 3.3).

All our analysis codes can be found on this public GitHub repository [10].

3 Time series analysis

The main results of the data reduction explained in the previous section, are time series of the occurrence of each of the 100 keywords in the queried twitter dataset. In order to identify relationships between different keywords, we compare their occurrence frequencies using three different methods based on the Pearson correlation coefficient and k means clustering.

3.1 Correlation between stationary time series

We can determine the amount of linear dependence between two time series X_i and X_j using the correlation coefficient ρ. The correlation $\rho(\Delta t)$ between two time series is a function of the time lag Δt between the two and is defined as [11]

$$\rho_{ij}(\Delta t) = \langle (X_{i,t} - \bar{X}_i)(X_{j,t+\Delta t} - \bar{X}_j) \rangle, \tag{2}$$

where $\langle ... \rangle$ denotes the ensemble average. For $\Delta t = 0$ we recover the usual correlation coefficient, whereas for $\Delta t \neq 0$, $\rho_{ij}(\Delta t)$ quantifies the amount of linear dependence between shifted time series. If the two time series are equal i.e. $i = j$, then we recover the autocorrelation function of the time series, which quantifies the amount of linear dependence in the time series itself. In order to compare the correlation between different time series, it is customary to normalise Eq. 2 by the variance of the two time series i.e. [11]

$$\rho_{ij}(\Delta t) = \frac{\langle (X_{i,t} - \bar{X}_i)(X_{j,t+\Delta t} - \bar{X}_j) \rangle}{\sqrt{\langle (X_{i,t} - \bar{X}_i)(X_{i,t} - \bar{X}_i) \rangle \langle (X_{j,t} - \bar{X}_j)(X_{j,t} - \bar{X}_j) \rangle}}. \tag{3}$$

When performing time series analyses we generally do not have access to the ensemble averages of the series, but we need to estimate the auto and cross correlations from the samples we have at hand. In order to be able to estimate the auto- or cross correlation

using Eq. 2, the ensemble average needs to equal the sample mean i.e. the values of both time series at all times t need to identically distributed random variables. This is equivalent to requiring both time series X_i, X_j to be stationary, which means that their mean μ, variance σ^2 and autocorrelation function $\rho_{ii}(\Delta t)$ do not depend explicitly on time t

$$\mu = \langle X_{i,t} \rangle, \tag{4}$$

$$\sigma^2 = \langle (X_{i,t} - \bar{X})^2 \rangle \tag{5}$$

$$\rho_{ii}(\Delta t) = \frac{\langle (X_{i,t} - \bar{X}_i)(X_{i,t+\Delta t} - \bar{X}_i) \rangle}{\sqrt{\langle (X_{i,t} - \bar{X}_i)(X_{i,t} - \bar{X}_i) \rangle \langle (X_{i,t+\Delta t} - \bar{X}_i)(X_{i,t+\Delta t} - \bar{X}_i) \rangle}}. \tag{6}$$

Only in this case, can we replace the ensemble averages in Equations 2 and 3 with the sample means. This leads to the following estimators for the auto and cross correlation functions [11]:

$$\rho_{ii}(\Delta t) = \frac{\sum_{t=1}^{n-\Delta t} (X_{i,t} - \bar{X}_i)(X_{i,t+\Delta t} - \bar{X}_i)}{\sum_{t=1}^{n} (X_{i,t} - \bar{X}_i)^2} \tag{7}$$

$$\rho_{ij}(\Delta t) = \frac{\sum_{t=1}^{n-\Delta t} (X_{i,t} - \bar{X}_i)(X_{j,t+\Delta t} - \bar{X}_j)}{\sqrt{\sum_{t=1}^{n} (X_{i,t} - \bar{X}_i)^2 \sum_{t=1}^{n} (X_{j,t} - \bar{X}_j)^2}}. \tag{8}$$

3.2 Differencing of non-stationary time series

We do not expect the occurrences of keywords in twitter data to be stationary time series, since the mean occurrence of any keyword will for example depend on world events, its popularity and also on the number of people using twitter. We can roughly assess if any time series is stationary by visually inspecting the time series plot and also by computing its autocorrelation function $\rho(\Delta t)$. Non stationary time series are generally characterised by clear patterns in the autocorrelation functions because consecutive values are strongly correlated through the common trend.

In order to be able to quantify the correlation between non stationary time series we need to make the series stationary prior to evaluating Equations 2 and 3. The most basic form is called differencing, which allows us to transform time series with a slow trend to stationary series. A time series with a trend can be written as [11]

$$X_{i,t} = T_{i,t} + R_{i,t}, \tag{9}$$

where $T_{i,t}$ denotes the time series' trend and $R_{i,t}$ are random fluctuations around this trend. These fluctuations $R_{i,t}$ are generally correlated random variables with mean zero. If the time series varies slowly, we can remove the trend by considering the differenced

time series instead of the initial one. The differenced time series is defined as [11]

$$D_{i,t} = X_{i,t+1} - X_{i,t}. \tag{10}$$

If $T_{i,t}$ is slowly varying, this transformed time series will only capture the random variations around the global trend and will thus approximately be stationary.

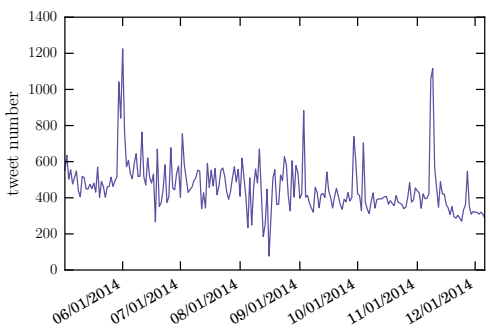
For non stationary time series, we can thus first perform differencing and then compute the auto and cross correlations between the differenced time series. This will encode the amount of linear dependence between changes in one time series from its global trend and changes in the second.

In order to quantify the significance of both the auto- and cross correlations, we need to compare the measured values to the expected values for uncorrelated time series. For uncorrelated time series, which are a sequence of IID variables we expect both for the auto and cross correlations $\rho(\Delta t) = 0$ with a variance of $\sigma^2(\rho(\Delta t)) = \frac{1}{n}$, where n denotes the number of samples used to estimate the correlation [11]. For simplicity, we will call a cross-correlation between two time series significant if its value exceeds $1.96\sigma(\rho(\Delta t))$ (i.e. the 95% confidence limits), even though we do not expect the differenced time series to be perfect IID processes.

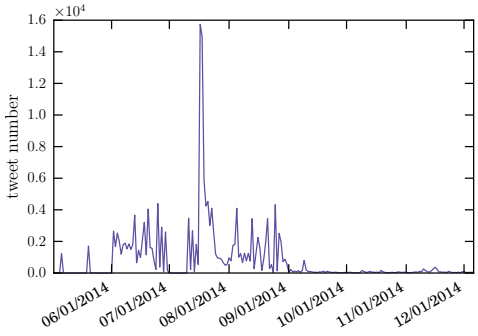
3.3 Missing data

The above considerations assume that there is no missing time series data, which is rarely the case in reality. Several pieces of data are missing in the twitter dataset. First of all we do not have access to data for most of August 2014 and furthermore there are several days missing thoughout. In order to be able to analyse the data, we therefore fill the missing values using random samples drawn from a normal distribution with the mean and variance of the respective time series. This procedure will keep mean and standard deviation constant and will not introduce any correlations between different time series.

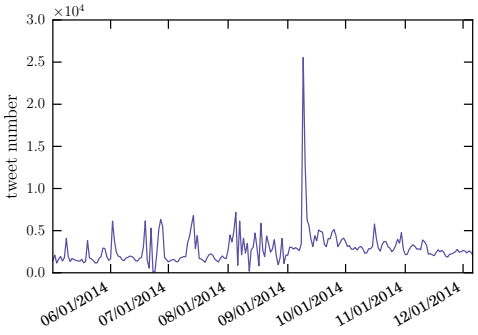
In order to find relations between keywords, we will proceed in two steps: for illustration purposes we will compute the cross correlation between all the time series (filled as explained above) regardless of stationarity for timelags $-10 \leq \Delta t \leq 10$. We will then compute the autocorrelation function of all the series and recompute the correlation coefficients using the differenced time series for non stationary processes. For both methods, we assign the highest correlation coefficient (regardless of timelag Δt) to each pair and compare the results obtained with these two measures.



(a) Occurrence frequency for *banana*.



(b) Occurrence frequency for *MH17*.

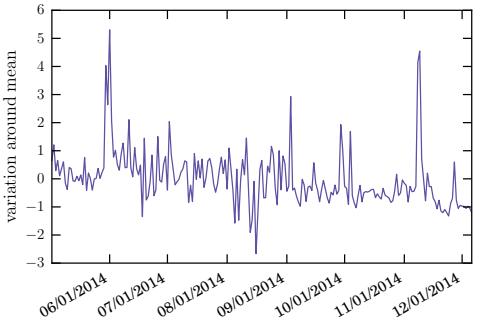


(c) Occurrence frequency for *apple*.

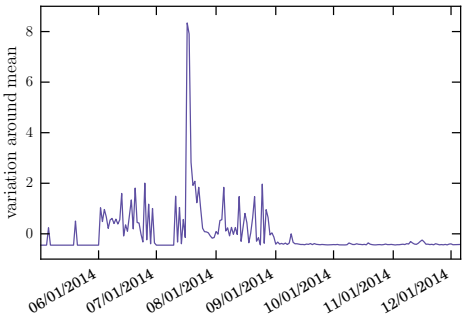
Figure 2: Absolute number of keyword occurrences as a function of time.

4 Results

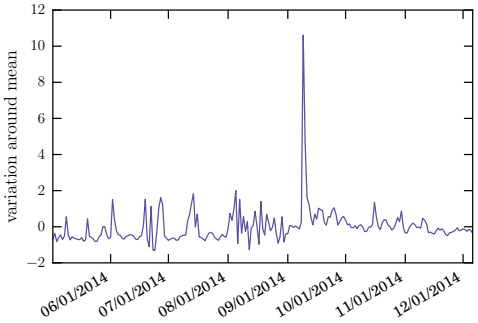
The occurrence of different keywords over time displays different behaviour depending on the keyword. Keywords from the everyday life and family sections for example feature stationary time series with little to no variation. On the other hand, keywords associated to major events show significant trends and structure. Examples for both these cases are shown in Figures 2 and 3. The Figures show the absolute and relative time series for the words *banana* and *MH17* respectively. *MH17* is the abbreviation for the Malaysian airlines flight 17, which was shot down over the Ukraine on July, 17th 2014. We see that the mentions of *MH17* are concentrated in this period while the word banana appears with a constant frequency over the 6 months we have analysed. Another interesting example is the word *apple*, which denotes the fruit as well as the company. We would expect the frequency of the word *apple* to be fairly time-independent, but from the plot we see that the number of mentions features a strong peak in September, 2014. This is probably caused by apple holding its keynote speech during this period of the year. This example shows that many words can have different meanings depending on the context or language which can cause *a priori* unexpected features in the time series.



(a) Occurrence frequency for *banana*.

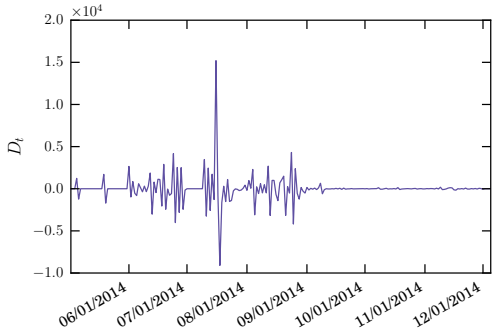


(b) Occurrence frequency for *MH17*.

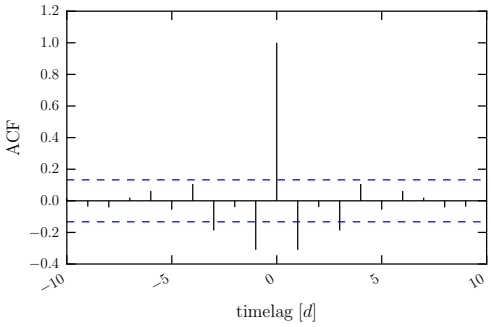


(c) Occurrence frequency for *apple*.

Figure 3: Mean subtracted number of keyword occurrences as a function of time normalised by the standard deviation of the time series.



(a) Differenced time series for *MH17*.



(b) Autocorrelation function for *MH17*.

Figure 4: Differenced time series and autocorrelation function for *MH17*. The dashed lines indicate 95% confidence intervals for IID data.

From Figures 2 and 3, we see that in general, the time series are not stationary. We therefore transform all the time series by differencing, as outlined in Eq. 10. The differenced time series for the keyword *MH17* are shown in Fig. 4 along with its autocorrelation function. We see that the differenced time series is largely stationary, even though *MH17* is one of the most strongly varying keywords in our sample. This is confirmed by the autocorrelation function.

The cross correlation coefficients between keyword pairs for both methods are shown in Fig. bla. As expected, we see a correlation of 1 on the diagonal, corresponding to the auto correlations. The correlation matrix has a block-diagonal form corresponding to the correlations within keyword groups, e.g. keywords within the refugees and terrorism categories. Comparing the two correlation measures, we find that the correlations computed from the differenced time series are more pronounced than the ones computed using the raw data. This is unexpected, since we would have naively expected, the latter method to generally overestimate correlations.

As can be seen from the figure, we are able to retrieve expected correlations, such as correlation between *nelson* and *mandela* or *refugees* and *syria*. However, we also find several unexpected and probably spurious correlations between keywords such as *MH370* and *swine* or *linux* and *allah*. This is probably both due to the simple methods employed to compute correlations as well as that computing correlations between noisy datasets without clear features is generally a difficult problem.

5 Conclusion

In this project we analysed roughly six months of unfiltered twitter data by querying the tweets for a set of 100 predefined keywords. We aimed at recovering correlations between the occurrence frequencies of related keywords and showing that spurious relationships can be detected in large datasets. We used both the time series of the number of tweets per day containing a particular keyword and its differenced counterpart to compute correlation coefficients for pairs of keywords.

While we found large correlation coefficients for *a priori* expected keyword pairs, we also measured a significant number of unexpected and probably spurious correlations.

6 Acknowledgements

This project was part of the lecture *Data Science in Techno-Socio-Economic Systems* given by Dr. Evangelos Pournaras, Prof. Dr. Dirk Helbing and Dr. Izabela Moise at ETH Zurich during the spring semester 2016. We would like to thank Dr. Izabela Moise and Dr. Rok Roskar for the valuable guidance and helpful discussions. We used Apache Spark [12] to query the Twitter data, and Python [13] and bokeh [14] for the data analysis and visualization. The website was built on a html5up [15] template.

References

[1] http://tylervigen.com/spurious-correlations

[2] http://spuriousrelationships.eu.pn/

[3] https://twitter.com/

[4] https://www.facebook.com/

[5] https://www.reddit.com/

[6] https://dev.twitter.com/streaming/public

[7] http://www.esa.int/Our_Activities/Space_Science/Rosetta

[8] http://www.who.int/en/

[9] https://spark.apache.org/

[10] https://github.com/cbruderer/spuriousrelations

[11] Marcel Dettling, Applied Time Series Analysis, SS 2014, Zurich University of Applied Sciences, https://stat.ethz.ch/education/semesters/ss2014/atsa/Scriptum_v140523.pdf

[12] http://spark.apache.org

[13] https://www.python.org

[14] http://bokeh.pydata.org/en/latest/

[15] http://html5up.net