Detecting temporal dependence in databases

Joaquin Cuomo, Sanket Sanjeev Mehrotra, Hajar Houmayouni, Indrakshi Ray
Department of Computer Science
Colorado State University
{*jcuomo, smehrotra, hhajar, iray*}*@colostate.edu*

***Abstract*—This paper presents a theoretical basis for automatically determining the presence of temporal information in a database given no prior knowledge about its attributes or properties. To do so we first explore different properties of temporal sequences and databases that can influence the analysis. The analysis is based on the Ljung-Box test for autocorrelation and we used it to create a table of proposed metrics to evaluate the temporality of a database.**

***Index Terms*—Database Management Systems, Statistics, Temporality detection, Autocorrelation.**

I. Introduction

A common problem across data processing is the presence of outliers. An outlier is a data point or set of points that is unexpected and whose value differs significantly from other observations in the dataset. The most common way to handle outliers is to remove them when cleaning the data, as they can cause errors during the analysis phase. To detect those outliers, constraints in the data must be learnt and knowing information about how the data behave can help to do so. For example, knowing that there is a temporal relation within data points that allows us to exploit that information to learn constraints in the data that otherwise it would not be possible. Many of the techniques to detect outliers in data with temporal dependence imply modeling the time-series to make predictions. Fault detection approaches for sequence data may involve both stochastic and machine learning (ML) techniques. The stochastic techniques are modeling the data using AR (Autoregressive) models, MA (Moving Average) models, and any of its variants such as SARIMA (Seasonal AR Integrated MA) models. Whereas an ML approach may use LSTMs (Long Short Term Memory Networks), SVMs (Support Vector Machines), MLPs (Multi Layer Perceptrons) or ANNs (Artificial Neural Networks).

This work is an effort to enhance the tool developed in “ADQuaTe: An Automated Data Quality Test Approach for Constraint Discovery and Fault Detection“ by H. Homayouni, S. Ghosh and I. Ray (2019) [10]. The purpose of this tool is to discover constraints and faults in databases in an unsupervised fashion. However, these detections only apply to non-sequence data. The next step in the development of this tool is to add a second algorithm to detect outliers exploiting temporal relations in the data. In this paper we focus in determine which of the outlier detection techniques (temporal or non-temporal) should be apply and we based that decision on whether the database present or not temporal dependence within its attributes.

A. Problem description

We aim in this paper to analyze how temporal dependence on data can be found under the assumption of not having prior information on any aspect of the database. In order to do this, we first answered what implies having temporal dependence in the context of the outlier detection techniques that are going to be used. This requires us to first review time-series analysis.

B. Related Work

Time-series analysis has been a field under study for decades and has applications ranging from stock market prediction to digital signal processing. It has been extensively studied in statistics [4], econometrics [13] and in the field of communications [1]. Nevertheless, we did not find any papers related to the detection of temporal dependence in databases. The vast research in the area is about modeling time-series for either prediction or optimization purposes, and despite there is work on what is called temporal data mining [11] it is finding patterns on already assumed temporal sequences. The reason for lack of bibliography is mainly because we are solving a general question that is normally not required to be answered as it is known or presumed beforehand, but it is necessary in the context of fully automating ADQuaTe.

C. Contribution

Our contributions are: (1) setting a theoretical background on what exactly is needed to be searched for in the data in order to be successfully classified and that way choose the right outlier detection algorithm. (2) Also, the ground basis of an algorithm to detect temporality in databases. (3) Finally, we proposed metrics that can be useful to enhance the accuracy of the algorithm.

II. Methods

A. Temporal series background

A time series is a sequence of observations equally spaced and ordered by time. Normally, these observations are not independent within each other because the order is important. This non-independence means that there is a temporal dependence implying that future values are influenced by past values. The classical approach to analyze temporal series is to consider them as a combination of four components. This combination could be additive or multiplicative:

$$X_t = Trend + Seasonal + Cyclical + Irregular \tag{1}$$

$$X_t = Trend \cdot Seasonal \cdot Cyclical \cdot Irregular \tag{2}$$

Secular trends describes the general tendency of the data for a long period, seasonal variations describes the periodic fluctuation within cycles, cyclical component describes longer periodic fluctuations, and irregular variation describes those small changes that are unpredictable.

One of the most important properties of a time-series for its analysis is the stationarity. A time series is said to be stationary if its statistical properties do not change over time, i.e. if it has constant mean and variance, and covariance is independent of time.

Finally, autocorrelation is a measure of the similarity of the observations at certain lag. i.e. the correlation of the series with a delayed copy of itself. It gives critical information on whether future values can be predicted knowing the past ones. For example, stock market prediction is very hard to solve because it resembles more to a random walk than to a autocorrelated time series. Fitting the time-series is a major step in finding outliers in it, and regression analysis uses autocorrelation to do so as it provides information such as seasonality and the order of the fitting model [6]. Therefore, autocorrelation is going to be the most important metric to determine if a database has or does not have temporal dependence. There might be cases where the database consists of samples from a temporal sequence but no autocorrelation is detected. However, if there is no autocorrelation no information can be easily withdrawn based on that assumption and most likely no outlier would be detected.

A potential flaw in using the autocorrelation as the main metric is that it can be detected even when the data is not temporal. A typical example would be cross-sectional data, which is normally data collected at a single point in time and is not unusual to be autocorrelated as there might be a correlation in space [22]. For example, houses from the same neighborhood might have similar electricity rates compare to others from different neighborhoods. Nevertheless, we mentioned it as a potential flaw because we believe that in many cases the temporal outlier detection can also work on autocorrelated cross-sectional data. Hence, depending on what properties of the data the outlier detection algorithms harness, the presence of autocorrelation could be necessary but not sufficient.

B. Testing Autocorrelation

Most of the literature on testing for autocorrelation in time-series is based on evaluating the fitness of an autoregressive model. This is done testing for autocorrelation in the residuals of the model and therefore it implies more conditions and restrictions compared to our case in which we do not have any previous knowledge on our data. We will not go over these differences and each particular case, but it is important to be aware that when a model is tested there are further considerations to be taken into account [14]. The most popular are Ljung-Box [12], Box-Pierce [3] and others like Wald–Wolfowitz, Breusch–Godfrey [18], Daniel-Peña [16] and Monte-Carlo [7] which overcome some of the limitations of the first two but are more focused on time-series model's residuals. Both Ljung-Box and Box-Pierce methods are portmanteau tests in which the hypothesis to be tested is well defined but not the alternative hypothesis. Therefore, it allows testing the autocorrelation of a time series at multiple lags at the same time. The null hypothesis is that the data is independently distributed while the alternative hypothesis is that the data exhibits serial correlation up to any lag. The distribution of the tests approximates asymptotically to a χ^2 and the rejection of the null hypothesis when the p-value is less than the corresponding value for the desired confidence interval will tell us that there is autocorrelation in our data. Ljung-Box is a modification of Box-Pierce and it approximates better to a χ^2 [12]. The formula is:

$$Q(m) = n(n+2) \cdot \sum_{k=1}^{m} \frac{r_k^2}{n-k} \tag{3}$$

$$Q > X^2(1-\alpha, h) \tag{4}$$

where n is the number of samples, m is the maximum lag to test for autocorrelation, and r is the autocorrelation.

The degree of freedom of the χ^2, when there is no other knowledge about the data, should be equal to the number of lags up to where the autocorrelation is being tested. The choice of lag is hard to pick when no information about the data is known. The higher the lag the lower the performance of the test. Also, the lag should be a fraction of the sequence length. For example, Stata implementation uses the rule of h=min(n/2,40) [5], while Box et al. suggest h=20 [21], and Ruey S. Tsay suggests h=ln(n) [20] warning that when seasonal behavior is expected this needs to be taken into consideration and lag values at multiples of the seasonality are more important. Escanciano and Lobato presented a portmanteau test that automatically chooses the lag [8].

1) Limitations:

a) Small sample size: When the sample size is small no statistic test will have enough significance. Depending on the method used this number might change. The Central Limit Theorem shows that a minimum sample size of 30 would work for approximating the sample means to a standard distribution [9], which is the base to many of the techniques applied here. Specific to autoregressive analysis, Box et al. said that 50 would be the minimum sample size to work with ARIMA models [2].

b) Unevenly-sampled data: To estimate the autocorrelation both variance and averages are computed, which are both normalized to the sample length. Therefore, if the uneven sampling is due to missing data points and the sample size is large enough, the limit should converge to the same values as if all data points were present. However, if there is no pattern in the sampling rate different methods should be used to calculate the autocorrelation indirectly. The Wiener–Khinchin theorem states that the Fourier Transform of the autocorrelation of a stationary random process is the power spectral density, therefore the autocorrelation can be estimated from it. Rhefeld et al. compared two of the most common methods, the Lomb-Scargle and the Gaussian kernel [17].

c) Missing values: There are many methods to overcome missing values in time-series data and specific to Ljung-Box test [19]. But, under the assumption that we do not have prior information on the data set we would not be able to decide if one of these methods should be used.

d) Cross-sectional data: Autocorrelation is a necessary condition to exploit temporal data information but it is not a sufficient condition to determine if the data is temporal. There could be datasets with correlations that are not temporal related but spatial for example [22].

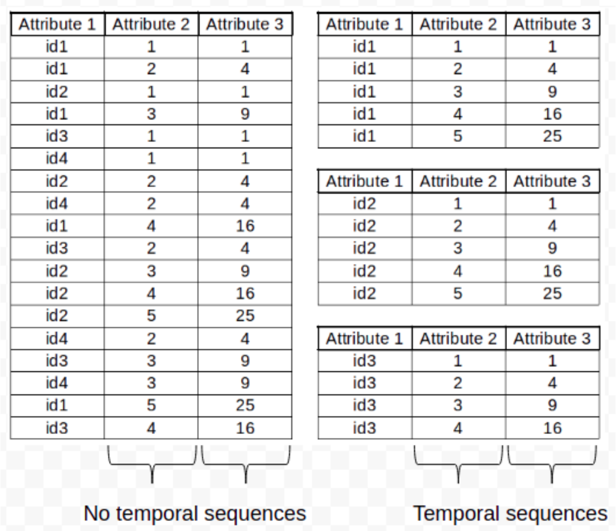
e) Non-stationarity: Non-stationary time-series are those with varying statistical properties over time, therefore the autocorrelation cannot be calculated using the mean and the variance, and thus, it needs to be estimated. Similar methods, as those mentioned before using the frequency spectrum could be used as well as the wavelet transform [15].

C. Study Cases

We will focus our work in three types of databases.

- No temporal dependence: Given a dataset with no temporal information no autocorrelation is expected.
- A continuous evenly-sampled time-ordered database: Given a dataset that corresponds to a continuous-time window we can detect the temporal dependence by computing the autocorrelation of each attribute as a whole.
- Temporal dependence within filtering attribute: Given a dataset that contains blocks of continuous-time windows we can detect the temporal dependence by computing the autocorrelation of each attribute within each block. Here, finding the proper grouping attribute is the crux of the problem.

Figure 1 shows an example on how a database might have hidden temporal dependencies that are uncovered



Attribute 1	Attribute 2	Attribute 3
id1	1	1
id1	2	4
id2	1	1
id1	3	9
id3	1	1
id4	1	1
id2	2	4
id4	2	4
id1	4	16
id3	2	4
id2	3	9
id2	4	16
id2	5	25
id4	2	4
id3	3	9
id4	3	9
id1	5	25
id3	4	16

Attribute 1	Attribute 2	Attribute 3
id1	1	1
id1	2	4
id1	3	9
id1	4	16
id1	5	25

Attribute 1	Attribute 2	Attribute 3
id2	1	1
id2	2	4
id2	3	9
id2	4	16
id2	5	25

Attribute 1	Attribute 2	Attribute 3
id3	1	1
id3	2	4
id3	3	9
id3	4	16

No temporal sequences
Temporal sequences

Fig. 1. Example of sub time-sequences by grouping attributes. Left shows the entire database. Right shows the decomposition by grouping by Attribute 1.

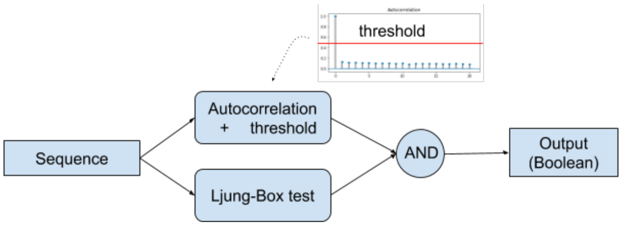


Fig. 2. Diagram of proposed algorithm.

once the proper attribute is used to filter the database. On the left the whole database does not exhibit any autocorrelation for any of the attributes. On the other hand, we can see in the right that after filtering by Attribute 1 in both Attribute 2 and Attribute 3 there are temporal series.

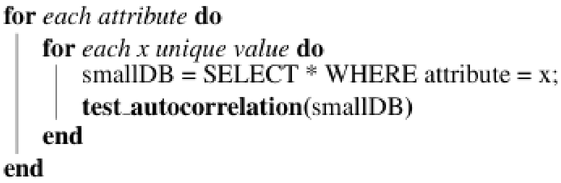
D. Algorithm

To test the theory discussed above we developed a simple algorithm, shown if Figure 2, that evaluates the autocorrelation of a database.

The core part of the algorithm consists of analysing a single sequence and determine if it has autocorrelation. For that, we do a Ljung-Box test and compute the autocorrelation function passed through a threshold. Then, if both tests are positive we conclude that there exists a significant autocorrelation, determined by a threshold, and that it is statistically significant.

The next part iterates over all the numeric attributes of the database and returns those attributes with autocorrelation and their corresponding value.

At this point, we introduce a test that considers that there might be temporal sequences when a particular attribute is grouped by common values. We simply do a brute force approach iterating over each attribute and testing for autocorrelation for each unique value in that attribute.



```
for each attribute do
    for each x unique value do
        smallDB = SELECT * WHERE attribute = x;
        test_autocorrelation(smallDB)
    end
end
```

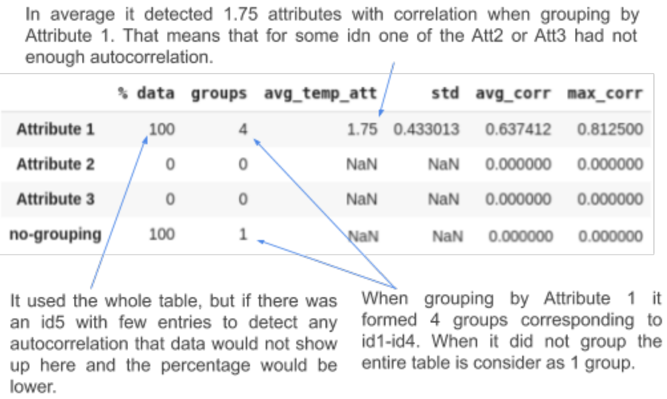
The output consists of a table showing statistics of filtering the database by each attribute. Therefore, the rows are the attributes of the database and the columns have the information shown in Table 1. Also, a row consisting of no-filtering-by-any-attribute is added.

TABLE I
METRICS

Name	Description
% data	Percentage of data used for the test
groups	Count of groups used for the test (the entries without autocorrelation detected are not counted)
avg_temp_att	Average of attributes with autocorrelation detected
std	Standard deviation of avg_temp_att
avg_corr	Average autocorrelation of every attribute with autocorrelation detected
max_corr	Maximum autocorrelation of every attribute with autocorrelation detected

Using the same example of the datatable in Figure 1 we show in Figure 3 how the resultant table would look like.

Then, we analyze the values from the table created to determine if indeed filtering by some attribute generates temporal sequences. The major metric is the average amount of temporal sequences detected weighted by the standard deviation of that measure to discard any spurious attribute value that might have given many more temporal sequences than the rest of the values for that same attribute. The more attributes found with autocorrelation the better that attribute is ranked to be used as the filtering attribute. Also, the amount of unique values in the grouping attribute is useful to discard extreme cases, such as when an attribute has a unique



In average it detected 1.75 attributes with correlation when grouping by Attribute 1. That means that for some idn one of the Att2 or Att3 had not enough autocorrelation.

	% data	groups	avg_temp_att	std	avg_corr	max_corr
Attribute 1	100	4	1.75	0.433013	0.637412	0.812500
Attribute 2	0	0	NaN	NaN	0.000000	0.000000
Attribute 3	0	0	NaN	NaN	0.000000	0.000000
no-grouping	100	1	NaN	NaN	0.000000	0.000000

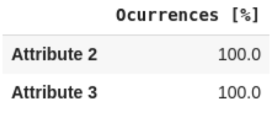
It used the whole table, but if there was an id5 with few entries to detect any autocorrelation that data would not show up here and the percentage would be lower.

When grouping by Attribute 1 it formed 4 groups corresponding to id1-id4. When it did not group the entire table is consider as 1 group.

Fig. 3. Example of an analysis of the metrics.

value in the entire database. Additionally, the average of the autocorrelation of each of the segments is evaluated and the highest value of the sum of autocorrelations of each segment attribute are taken into account.

Once the grouping attribute is chosen we can obtain those attributes with temporal dependence. As before, we create a result table shown in Figure 4 with rows being the potential attributes with temporal dependence, and the only column a score, being the percentage of grouping-attribute values for which autocorrelation was detected for that attribute. It then only remains to calculate a threshold to choose in which ones to run the outlier detection algorithm. The threshold is calculated as one standard deviation from the mean.



	Ocurrences [%]
Attribute 2	100.0
Attribute 3	100.0

Fig. 4. Example of result table for determining which attributes present temporal dependence.

III. Experiments

We conducted different experiments to show how the metrics we chose can help to determine if there is temporal dependence on the databases. As our method relies on the autocorrelation of sequences the final decision of whether it should be treated as a temporal sequences under the scope of finding outliers should be done depending on the outlier detection algorithm. Each of the experiments are performed in order to exemplify different scenarios and the assessment on the method proposed is done qualitatively over the metrics proposed.

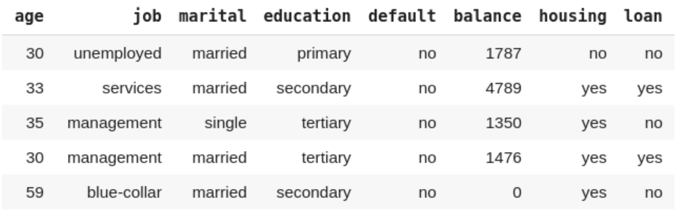
All the databases used here are publicly available and were picked to exemplify various scenarios. The link to each of them can be found in the Appendix.

We ran the algorithm for the three types of databases mentioned in the Study Cases section. For the last case, where we tested on databases we know there is a grouping attribute that can help in detecting the temporal sequences, we tried three different databases. This is done because of the difficulty of this case and to explore the usefulness of metrics we chose to assess the temporal dependence.

IV. Results

A. No temporal dependence in database

The “Bank Marketing Data Set” is a marketing campaign of a bank based on phone calls. Despite the fact that there could be more than one contact to the same client there is no identification of the subject nor the amount is enough to consider it a temporal sequence.



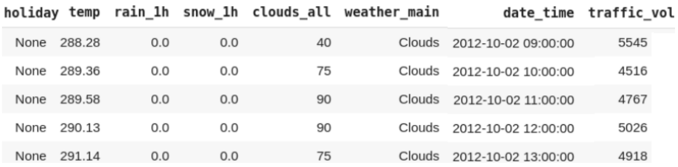
age	job	marital	education	default	balance	housing	loan
30	unemployed	married	primary	no	1787	no	no
33	services	married	secondary	no	4789	yes	yes
35	management	single	tertiary	no	1350	yes	no
30	management	married	tertiary	no	1476	yes	yes
59	blue-collar	married	secondary	no	0	yes	no

Fig. 5. Bank Marketing database

No autocorrelation was found and therefore no result table is constructed. See Appendix for extensive analysis.

B. No-grouping temporal database

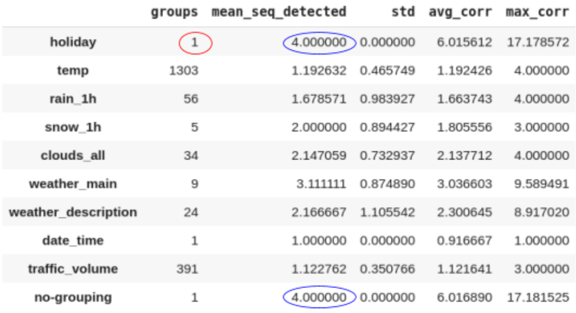
The “Metro Interstate Traffic Volume Data Set” database is the hourly traffic volume in a highway. Also, weather features and holidays are included. Therefore, there is a temporal sequence evenly-sampled every hour for various of the attributes.



holiday	temp	rain_1h	snow_1h	clouds_all	weather_main	date_time	traffic_vol
None	288.28	0.0	0.0	40	Clouds	2012-10-02 09:00:00	5545
None	289.36	0.0	0.0	75	Clouds	2012-10-02 10:00:00	4516
None	289.58	0.0	0.0	90	Clouds	2012-10-02 11:00:00	4767
None	290.13	0.0	0.0	90	Clouds	2012-10-02 12:00:00	5026
None	291.14	0.0	0.0	75	Clouds	2012-10-02 13:00:00	4918

Fig. 6. Metro Interstate Traffic database

The results in Figure 7 shows that no-grouping is the best option, which is correct as the ”date_time” attribute is in monotonically ascending order and the measurements correspond to the same sensors. We can notice that it also marked “holiday” as a grouping attribute but looking closely that is because there is only one unique value in that attribute meaning that it is exactly the same as no grouping at all.



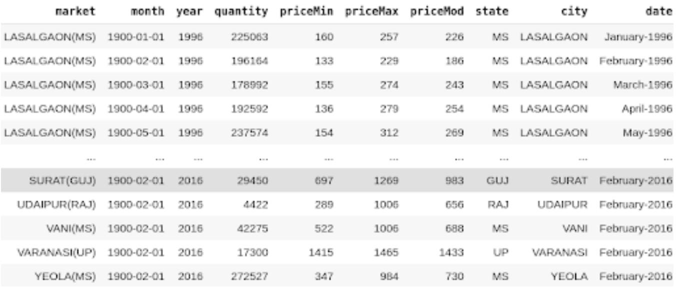
	groups	mean_seq_detected	std	avg_corr	max_corr
holiday	1	4.000000	0.000000	6.015612	17.178572
temp	1303	1.192632	0.465749	1.192426	4.000000
rain_1h	56	1.678571	0.983927	1.663743	4.000000
snow_1h	5	2.000000	0.894427	1.805556	3.000000
clouds_all	34	2.147059	0.732937	2.137712	4.000000
weather_main	9	3.111111	0.874890	3.036603	9.589491
weather_description	24	2.166667	1.105542	2.300645	8.917020
date_time	1	1.000000	0.000000	0.916667	1.000000
traffic_volume	391	1.122762	0.350766	1.121641	3.000000
no-grouping	1	4.000000	0.000000	6.016890	17.181525

Fig. 7. Metro Interstate Traffic analysis

See Appendix for extensive analysis.

C. Temporal dependence within grouping attribute

The “Market Arrivals” dataset consist of market values of different cities in India on a monthly basis for 20 years.

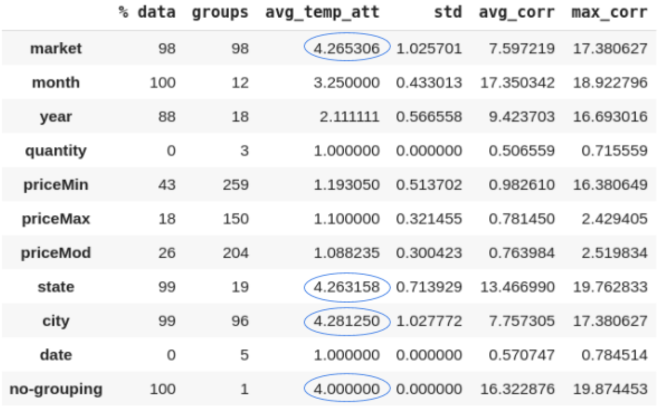


market	month	year	quantity	priceMin	priceMax	priceMod	state	city	date
LASALGAON(MS)	1900-01-01	1996	225063	160	257	226	MS	LASALGAON	January-1996
LASALGAON(MS)	1900-02-01	1996	196164	133	229	186	MS	LASALGAON	February-1996
LASALGAON(MS)	1900-03-01	1996	178992	155	274	243	MS	LASALGAON	March-1996
LASALGAON(MS)	1900-04-01	1996	192592	136	279	254	MS	LASALGAON	April-1996
LASALGAON(MS)	1900-05-01	1996	237574	154	312	269	MS	LASALGAON	May-1996
...	...	...	...	...	...	...	...	...	...
SURAT(GUJ)	1900-02-01	2016	29450	697	1269	983	GUJ	SURAT	February-2016
UDAIPUR(RAJ)	1900-02-01	2016	4422	289	1006	656	RAJ	UDAIPUR	February-2016
VANI(MS)	1900-02-01	2016	42275	522	1006	688	MS	VANI	February-2016
VARANASI(UP)	1900-02-01	2016	17300	1415	1465	1433	UP	VARANASI	February-2016
YEOLA(MS)	1900-02-01	2016	272527	347	984	730	MS	YEOLA	February-2016

Fig. 8. Market arrivals database

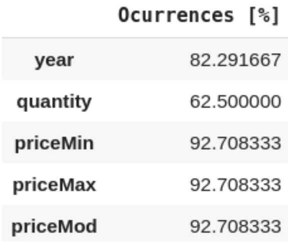
The output suggest to group by attribute “city” ranking close by “market” and “state”, which when we look back at the database all of them are almost the same. The ”no-grouping” rank was also high and it is not surprising as even if there is no grouping the database is ordered by time and is not unreasonable that many cities from the same country share a common behaviour.

Once we know the attribute, we obtain the name of the attributes that can have autocorrelation. In this case the threshold is 73% dropping out all except “quantity”.



	% data	groups	avg_temp_att	std	avg_corr	max_corr
market	98	98	4.265306	1.025701	7.597219	17.380627
month	100	12	3.250000	0.433013	17.350342	18.922796
year	88	18	2.111111	0.566558	9.423703	16.693016
quantity	0	3	1.000000	0.000000	0.506559	0.715559
priceMin	43	259	1.193050	0.513702	0.982610	16.380649
priceMax	18	150	1.100000	0.321455	0.781450	2.429405
priceMod	26	204	1.088235	0.300423	0.763984	2.519834
state	99	19	4.263158	0.713929	13.466990	19.762833
city	99	96	4.281250	1.027772	7.757305	17.380627
date	0	5	1.000000	0.000000	0.570747	0.784514
no-grouping	100	1	4.000000	0.000000	16.322876	19.874453

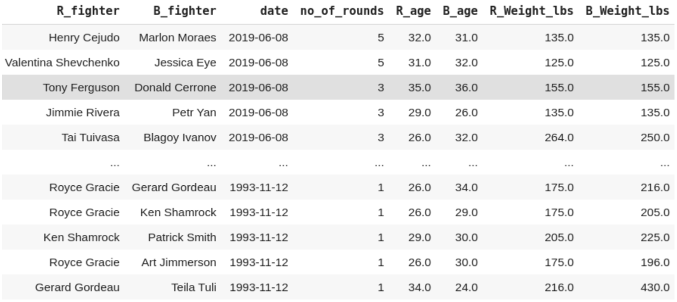
Fig. 9. Market Arrivals grouping attribute analysis



	Ocurrences [%]
year	82.291667
quantity	62.500000
priceMin	92.708333
priceMax	92.708333
priceMod	92.708333

Fig. 10. Market Arrivals results

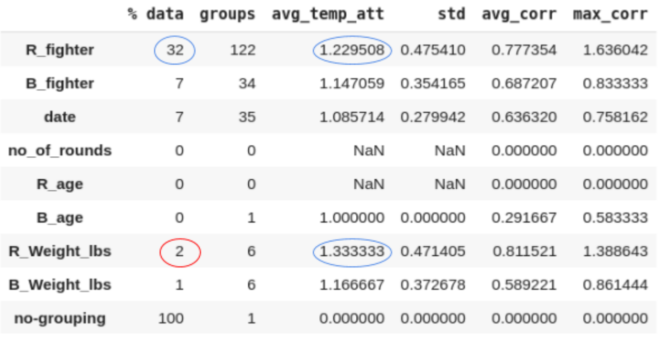
The “UFC-Fight historical dataset” is a list of every UFC fight in the history of the organisation ordered by time. In this case, it is not obvious which output to expect as, despite the database has a time order, each row is a sample from different ‘processes’. Therefore, no autocorrelation is expected. But, because with both “R_fighter” and “B_fighter” two independent tables can be constructed with data not related to the other table if grouping by one of those attributes a temporal sequence should be detected (for example in age of the fighter).



R_fighter	B_fighter	date	no_of_rounds	R_age	B_age	R_Weight_lbs	B_Weight_lbs
Henry Cejudo	Marlon Moraes	2019-06-08	5	32.0	31.0	135.0	135.0
Valentina Shevchenko	Jessica Eye	2019-06-08	5	31.0	32.0	125.0	125.0
Tony Ferguson	Donald Cerrone	2019-06-08	3	35.0	36.0	155.0	155.0
Jimmie Rivera	Petr Yan	2019-06-08	3	29.0	26.0	135.0	135.0
Tai Tuivasa	Blagoy Ivanov	2019-06-08	3	26.0	32.0	264.0	250.0
...	...	...	...	...	...	...	...
Royce Gracie	Gerard Gordeau	1993-11-12	1	26.0	34.0	175.0	216.0
Royce Gracie	Ken Shamrock	1993-11-12	1	26.0	29.0	175.0	205.0
Ken Shamrock	Patrick Smith	1993-11-12	1	29.0	30.0	205.0	225.0
Royce Gracie	Art Jimmerson	1993-11-12	1	26.0	30.0	175.0	196.0
Gerard Gordeau	Teila Tuli	1993-11-12	1	34.0	24.0	216.0	430.0

Fig. 11. UFC Fights database

As expected “no-grouping” data shows no presence of temporal dependence, but “R_Weight_lbs” has the highest amount in average of sequences detected. Never-

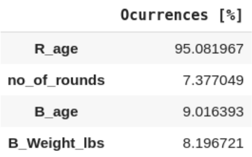


	% data	groups	avg_temp_att	std	avg_corr	max_corr
R_fighter	32	122	1.229508	0.475410	0.777354	1.636042
B_fighter	7	34	1.147059	0.354165	0.687207	0.833333
date	7	35	1.085714	0.279942	0.636320	0.758162
no_of_rounds	0	0	NaN	NaN	0.000000	0.000000
R_age	0	0	NaN	NaN	0.000000	0.000000
B_age	0	1	1.000000	0.000000	0.291667	0.583333
R_Weight_lbs	2	6	1.333333	0.471405	0.811521	1.388643
B_Weight_lbs	1	6	1.166667	0.372678	0.589221	0.861444
no-grouping	100	1	0.000000	0.000000	0.000000	0.000000

Fig. 12. UFC Fights grouping attribute analysis

theless, seeing that it uses only 2% of the entire database we can discard it and look at the next candidate which is “R_fighter” and then “B_fighter”, which is what we would expect.

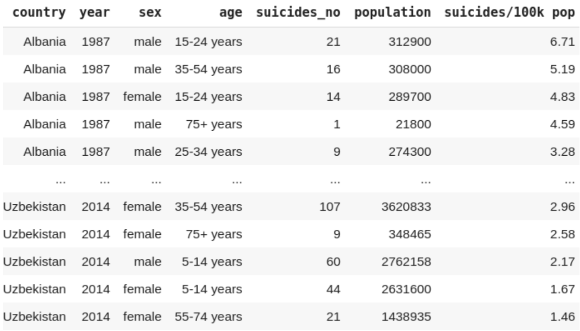
The attribute with autocorrelation is “R_age” and the rest is discarded by the threshold.



	Ocurrences [%]
R_age	95.081967
no_of_rounds	7.377049
B_age	9.016393
B_Weight_lbs	8.196721

Fig. 13. UFC Fights results

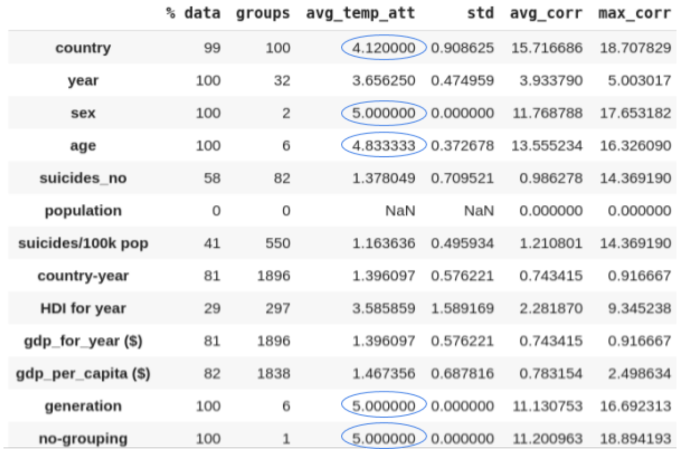
“Suicide Rates Overview” is a database comprising suicides statistic since 1987 from different countries. It has more than one potential grouping attribute such as country, age and sex.



country	year	sex	age	suicides_no	population	suicides/100k pop
Albania	1987	male	15-24 years	21	312900	6.71
Albania	1987	male	35-54 years	16	308000	5.19
Albania	1987	female	15-24 years	14	289700	4.83
Albania	1987	male	75+ years	1	21800	4.59
Albania	1987	male	25-34 years	9	274300	3.28
...	...	...	...	...	...	...
Uzbekistan	2014	female	35-54 years	107	3620833	2.96
Uzbekistan	2014	female	75+ years	9	348465	2.58
Uzbekistan	2014	male	5-14 years	60	2762158	2.17
Uzbekistan	2014	female	5-14 years	44	2631600	1.67
Uzbekistan	2014	female	55-74 years	21	1438935	1.46

Fig. 14. Suicides database

The algorithm gave a good ranking to all of the expecting grouping attributes but for country was lower than expected. After analyzing the root cause we found that the autocorrelation threshold (Figure 16) used failed in detecting autocorrelation in the “suicides_no” sequence for some of the countries. For a detailed analysis refer to the Appendix.



	% data	groups	avg_temp_att	std	avg_corr	max_corr
country	99	100	4.120000	0.908625	15.716686	18.707829
year	100	32	3.656250	0.474959	3.933790	5.003017
sex	100	2	5.000000	0.000000	11.768788	17.653182
age	100	6	4.833333	0.372678	13.555234	16.326090
suicides_no	58	82	1.378049	0.709521	0.986278	14.369190
population	0	0	NaN	NaN	0.000000	0.000000
suicides/100k pop	41	550	1.163636	0.495934	1.210801	14.369190
country-year	81	1896	1.396097	0.576221	0.743415	0.916667
HDI for year	29	297	3.585859	1.589169	2.281870	9.345238
gdp_for_year ($)	81	1896	1.396097	0.576221	0.743415	0.916667
gdp_per_capita ($)	82	1838	1.467356	0.687816	0.783154	2.498634
generation	100	6	5.000000	0.000000	11.130753	16.692313
no-grouping	100	1	5.000000	0.000000	11.200963	18.894193

Fig. 15. Suicides grouping attribute analysis

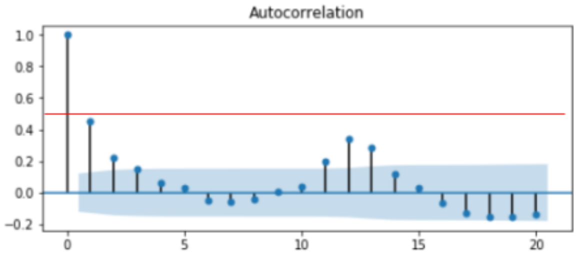
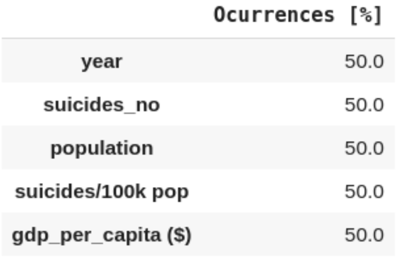


Fig. 16. Suicides’s attribute suicides_no autocorrelation

All of the following attributes have been found with autocorrelation and should be tested for outliers.



	Ocurrences [%]
year	50.0
suicides_no	50.0
population	50.0
suicides/100k pop	50.0
gdp_per_capita ($)	50.0

Fig. 17. Suicides results

V. Discussion and Future Research

We have presented an approach on how to classify databases as temporally dependent or not, and how to extract meaningful information to enhance the detection of outliers. The metric table proposed in this paper has shown to be useful for the latter task, in which the percentage of data used in the analysis, the number of groups and the average of autocorrelated sequences found were the three metrics that provided the most relevant information to make a decision. The others metrics were not used in this case examples but we believe that they could come handy in larger databases. For example, the standard deviation should not be too large as it would mean that there is a particular attribute value that has many more temporal sequences than the rest which is probably an outlier and should be handled carefully not to return a false positive attribute. Both the average autocorrelation and the maximum autocorrelation can be used as tiebreakers when the other metrics have close values.

We were able to successfully classify databases containing both temporal dependence only when filtering by some attribute and also in those cases where temporality exist in the whole database and also when filtering by attributes. The latter result is important because the outlier detection algorithm can be run on both cases and find different outliers.

The autocorrelation threshold was set arbitrarily and we could see it worked for most of the experiments but not for the last one. There should be an analysis on which are good options to set this parameter. It could be based on statistical analysis on what works better or based on theory or even it could be adjusted dynamically for every particular database. The suicide database also made evident that there should be a refinement on the lag choice. Because when we filter the database by some attribute the sub-databases obtained have different cardinality making crucial to standardized the cumulative autocorrelation to being able to make a comparison between them.

A list of attributes with autocorrelation is outputted so the outlier detection algorithm can know which attributes it can analyze. Similar to previous outputs, we use a simple threshold that can be further improved in order to minimize false positives and false negatives. The main advantage of this result is that it can improve the performance of the outlier detection as it does not need to go over all attributes, and it also allows during research to validate the classification.

A. Future work

a) Statistical Exploration and Optimization: It could be interesting to investigate if within the Ljung-Box or Box-Pierce test different types of correlation can be used, such as Pearson, Kendall, Spearman or estimation from the power spectral density. If so, a deeper analysis could be done on which autocorrelation function to use specially when no prior information on the data is known.

Currently, the algorithm goes over all numeric attributes searching for autocorrelation. This is time consuming and should be, if possible, improved.

The main flaw of this approach is that it classify as temporal dependent data those databases that present other type of autocorrelation than temporal. Therefore, an analysis using the outlier detection algorithms could be done to observe if autocorrelation is the property actually being exploited rather than temporality. We think that both stochastic and machine learning approaches at the end utilize either linear or non-linear autocorrelation to relate data points, thus making the tagging of the type of relation as temporal, spatial or any other irrelevant as long as they present some sort of autocorrelation.

b) Working with categorical attributes: Databases consist of more than just numeric values, there may be attributes such as Yes/No, Names, IDs, dates, etc. These attributes could contain temporal information as well. For example, a patient that got positive in a non-curable disease cannot get a negative in the future. Thus, finding a way to process this attributes is important. Converting them to one-hot representations could be a potential exploration path.

c) Label analysis: Parsing attribute labels could be a way of getting prior information from the database that could be harnessed. For example, when one of the attributes correspond to the row number or to a monotonically increasing ID may mislead our current approach by returning high values of autocorrelation. On the other hand, detecting the presence of time-indicator attributes can ease the classification. Also, reducing the amount of attributes analyzed for temporal dependence will improve the overall performance of the algorithm.

VI. Conclusion

In this paper, we have presented a technique that uses autocorrelation to classify the presence or not of temporal dependence within its attributes in any unknown database. The algorithm was tested for different databases having and not having temporal dependence data, and specifically focused on databases containing sub-time-sequences for a specific attribute. For these cases, we proposed metrics to find the grouping attributes that unveil the sub-sequences, and which are the sub-sequences to be tested for outliers. The results show that we were able to make successful classifications. Finally, we discussed the limitations of the approach and potential improvement paths.

VII. Author Contributions

Houmayouni proposed the problem to be solve. Cuomo and Mehrotra conceived of the presented idea. Cuomo developed the theory and performed the computations. All authors discussed the results. Cuomo and Mehrotra wrote the manuscript. Ray and Houmayouni helped supervise the project.

References

[1] M. Allen. *The SAGE Encyclopedia of Communication Research Methods*. SAGE Publications, 2017.

[2] G. E. P. Box and G. C. Tiao. Intervention analysis with applications to economic and environmental problems. *Journal of the American Statistical Association*, 70(349):70–79, 1975.

[3] G.E.P. Box and D.A. Pierce. Distribution of residual autocorrelations in autoregressive-integrated moving average time series models. *Journal of the American Statistical Association*, 72:397–402, 01 1970.

[4] Richard A. Davis. Introduction to statistical analysis of time series.

[5] Stata documentation. wntestq portmanteau (q) test description.

[6] Anais Dotis-Georgiou. Autocorrelation in time-series data. Influx Data article, accessed 25th Nov 2019.

[7] Jean-Marie Dufour and Lynda Khalaf. Monte carlo test methods in econometrics. 2007.

[8] J. Carlos Escanciano and Ignacio N. Lobato. An automatic Portmanteau test for serial correlation. *Journal of Econometrics*, 151(2):140–149, August 2009.

[9] D.F. Groebner, P.W. Shannon, P.C. Fry, and R.A. Donnelly. *Business Statistics: A Decision-making Approach*. Pearson Higher Education, 2017.

[10] Hajar Homayouni, Sudipto Ghosh, and Indrakshi Ray. Adquate: An automated data quality test approach for constraint discovery and fault detection. pages 61–68, 07 2019.

[11] Weiqiang Lin, Mehmet Orgun, and Graham Williams. An overview of temporal data mining. 11 2019.

[12] Greta Ljung and G. Box. On a measure of lack of fit in time series models. *Biometrika*, 65, 08 1978.

[13] Helmut Luetkepohl and Markus Krätzig. In applied time series econometrics. *Applied Time Series Econometrics*, 01 2004.

[14] G.S. Maddala. *Introduction to Econometrics*. Wiley, 2001.

[15] G. P. Nason, R. Von Sachs, and G. Kroisandt. Wavelet processes and adaptive estimation of the evolutionary wavelet spectrum. *Journal of the Royal Statistical Society Series B*, 62(2):271–292, 2000.

[16] Daniel Peña and Julio Rodríguez. A powerful portmanteau test of lack of fit for time series. *Journal of the American Statistical Association*, 97(458):601–610, 2002.

[17] Kira Rehfeld, Norbert Marwan, Jobst Heitzig, and Juergen Kurths. Comparison of correlation analysis techniques for irregularly sampled time series. *Nonlinear Processes in Geophysics*, 18:389–404, 06 2011.

[18] David Scott. Applied econometrics with r by christian kleiber, achim zeileis. *International Statistical Review*, 77:164–164, 04 2009.

[19] David Stoffer and Clélia Toloi. A note on the ljung–box–pierce portmanteau statistic with missing data. *Statistics Probability Letters*, 13:391–396, 04 1992.

[20] Ruey Tsay. *Analysis of Financial Time Series. Financial Econometrics*. 01 2002.

[21] Eric Ziegel, G. Box, G. Jenkins, and G.C. Reinsel. Time series analysis, forecasting, and control. *Technometrics*, 37:238, 05 1995.

[22] Andrea Zuur. Spatial correlation. San Fransisco State University article, accessed on 24th Nov 2019.

VIII. Appendix

A. Code

The code used for this paper can be find in the GitHub repository https://github.com/JCuomo/TemporalDependeceInDB

B. Databases

- **Metro Interstate Traffic**
  https://archive.ics.uci.edu/ml/datasets/Metro+Interstate+Traffic+Volume
- **Bank Marketing**
  https://archive.ics.uci.edu/ml/datasets/Bank+Marketing
- **Market Arrivals**
  https://github.com/selva86/datasets/blob/master/MarketArrivals.csv
- **UFC-Fights**
  https://www.kaggle.com/rajeevw/ufcdata
- **Suicide Rates Overview**
  https://www.kaggle.com/russellyates88/suicide-rates-overview-1985-to-2016

C. Databases detailed analysis

Here we present an analysis of each attribute to validate the lack of autocorrelation.

1) Bank Marketing analysis: Duration attribute has significant value for higher lags but the quantitative threshold in the ACF plot at 0.5 set the test to False.

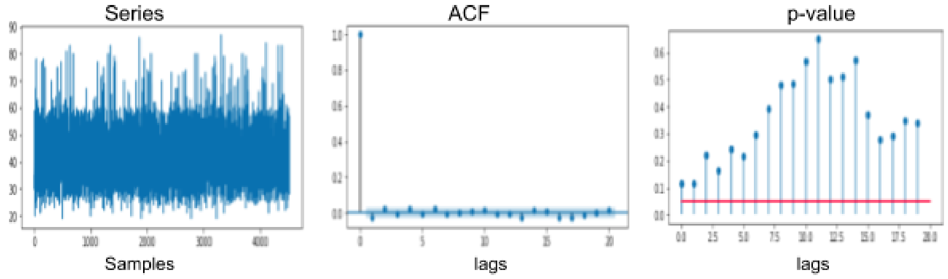


Fig. 18. Age attribute.

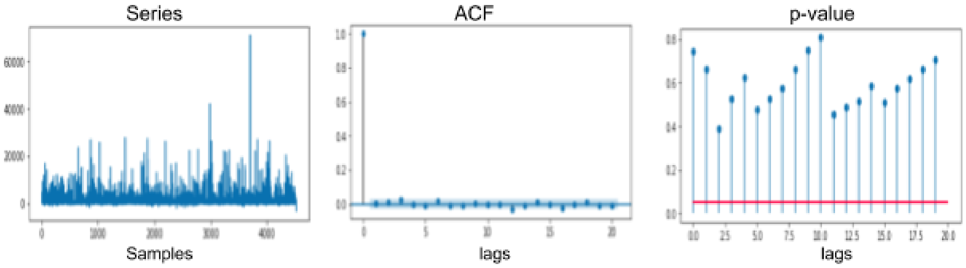


Fig. 19. Balance attribute.

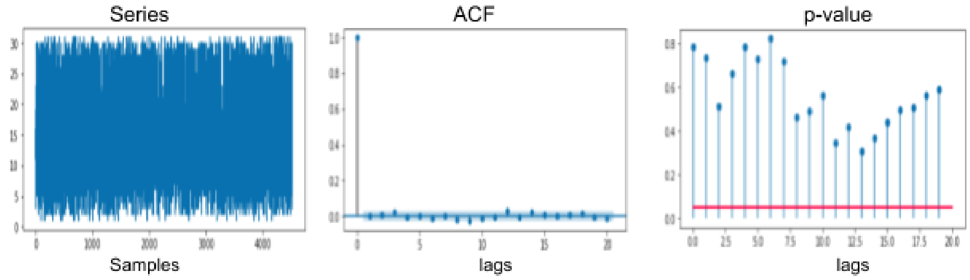


Fig. 20. Day attribute.

Fig. 21. Duration attribute.

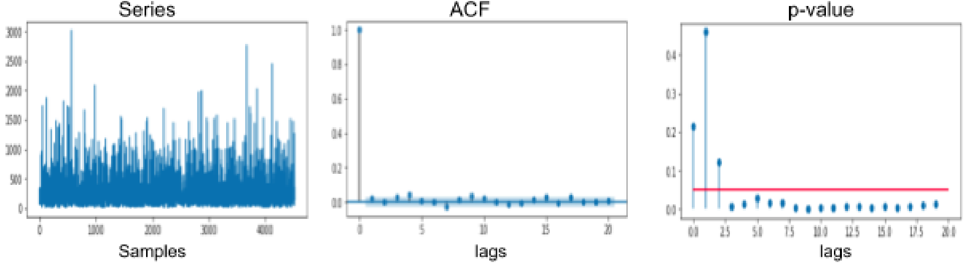


Fig. 22. Duration attribute.

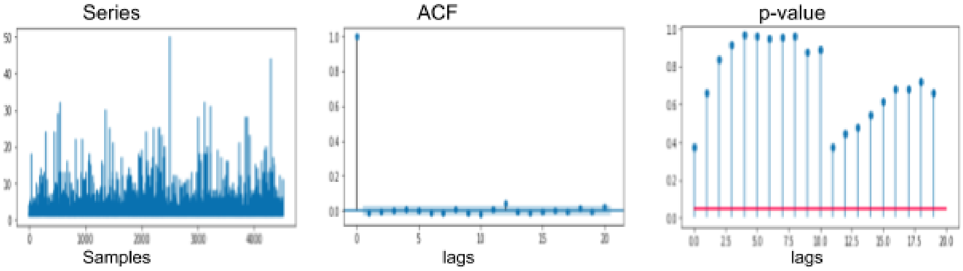


Fig. 23. Campaign attribute.

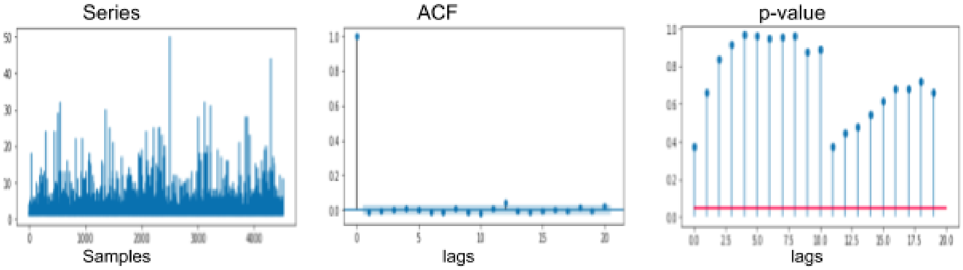


Fig. 24. Pdays attribute.

.

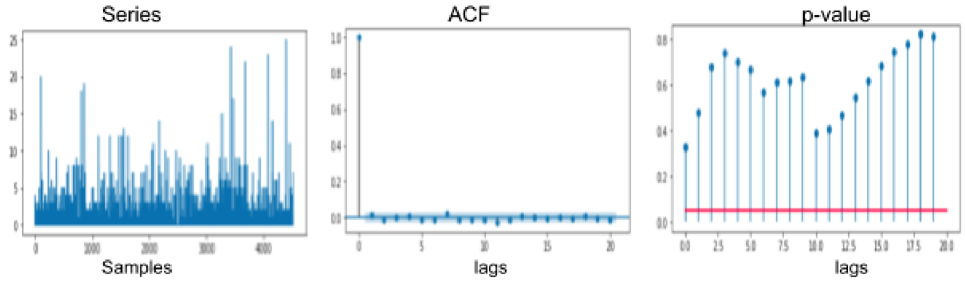


Fig. 25. Previous attribute.

2) Metro Interstate Traffic analysis data-set: There must be some text here to position the images correctly, I believe.

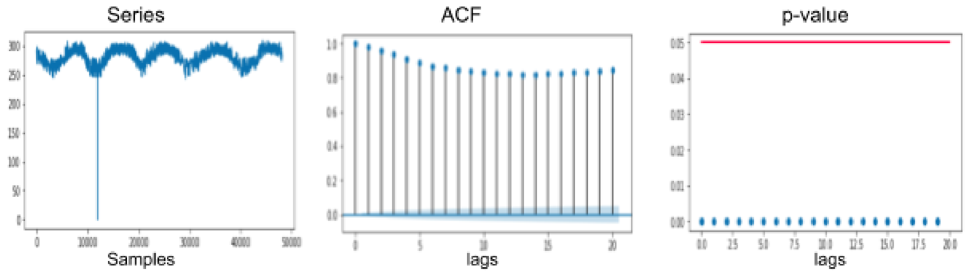


Fig. 26. temp column.

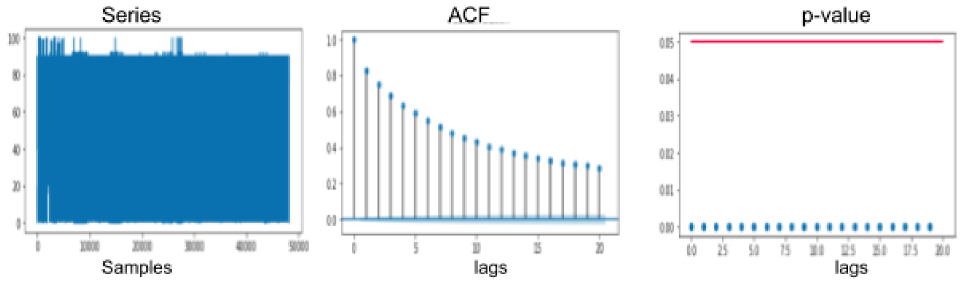


Fig. 27. clouds all column.

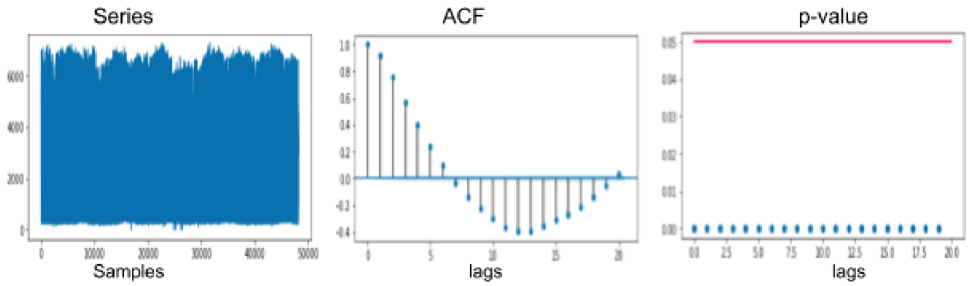
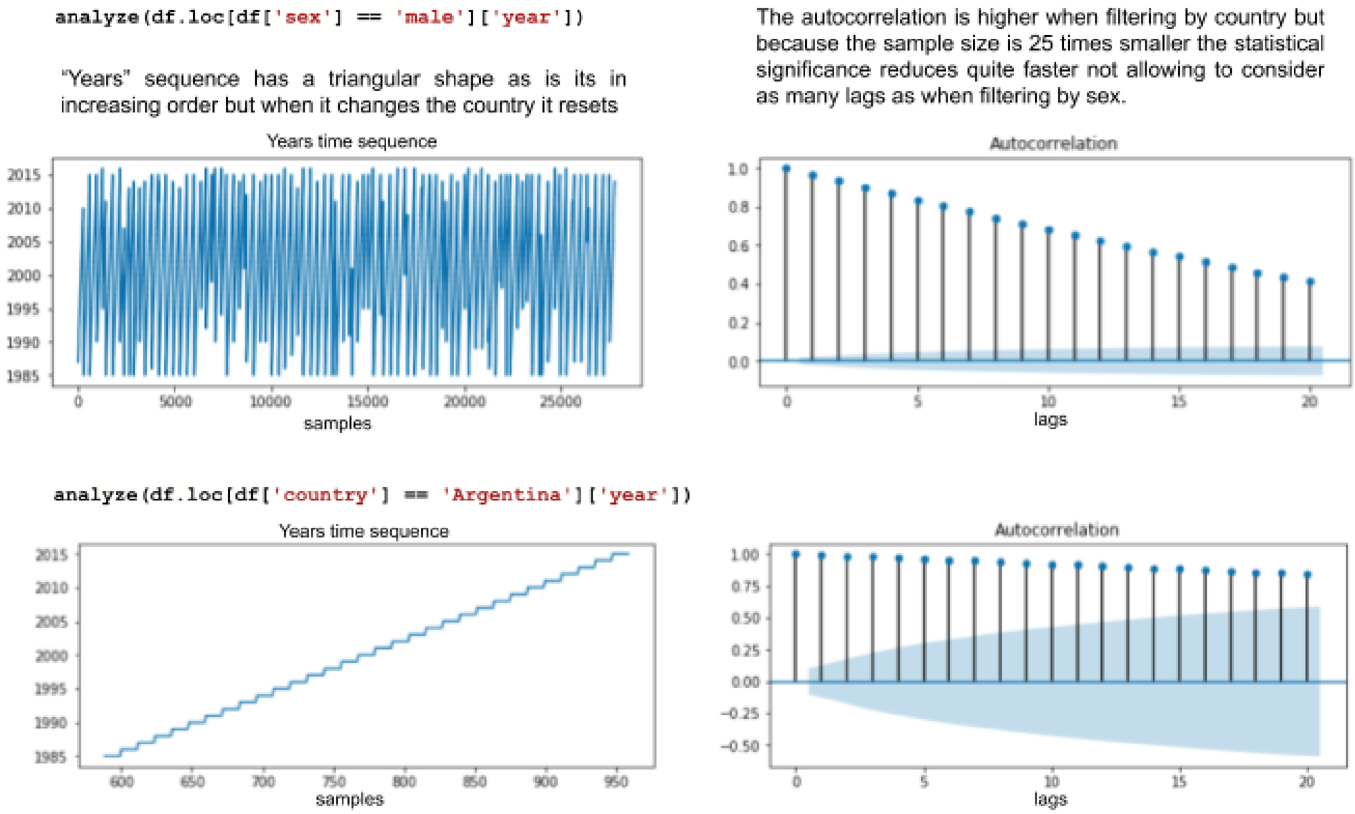


Fig. 28. traffic volume column.

3) *Suicides data-set:* .



```
analyze(df.loc[df['sex'] == 'male']['year'])
```

“Years” sequence has a triangular shape as is its in increasing order but when it changes the country it resets

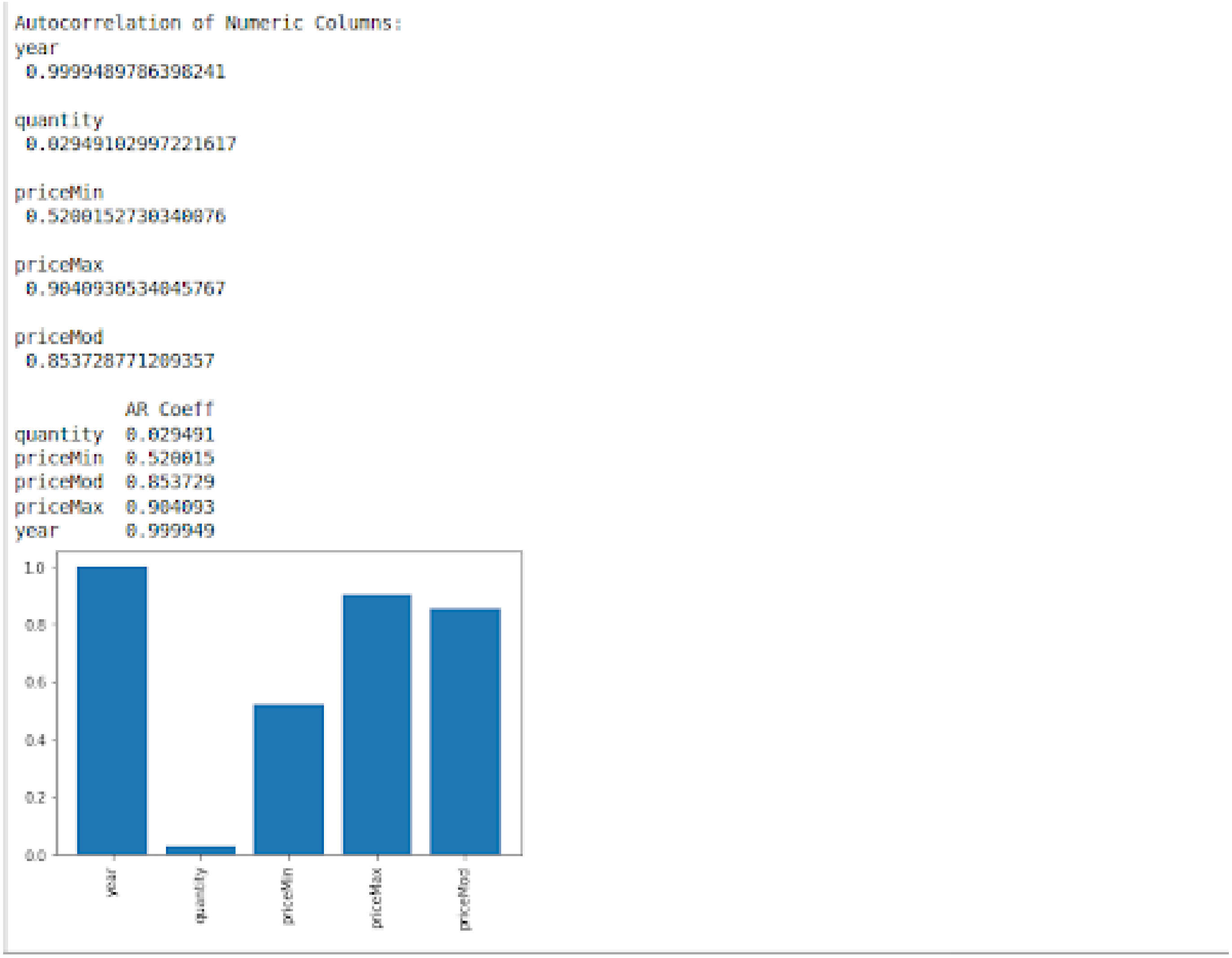
The autocorrelation is higher when filtering by country but because the sample size is 25 times smaller the statistical significance reduces quite faster not allowing to consider as many lags as when filtering by sex.

```
analyze(df.loc[df['country'] == 'Argentina']['year'])
```

Fig. 29. Analysis of the year attribute’s filtering by country or sex

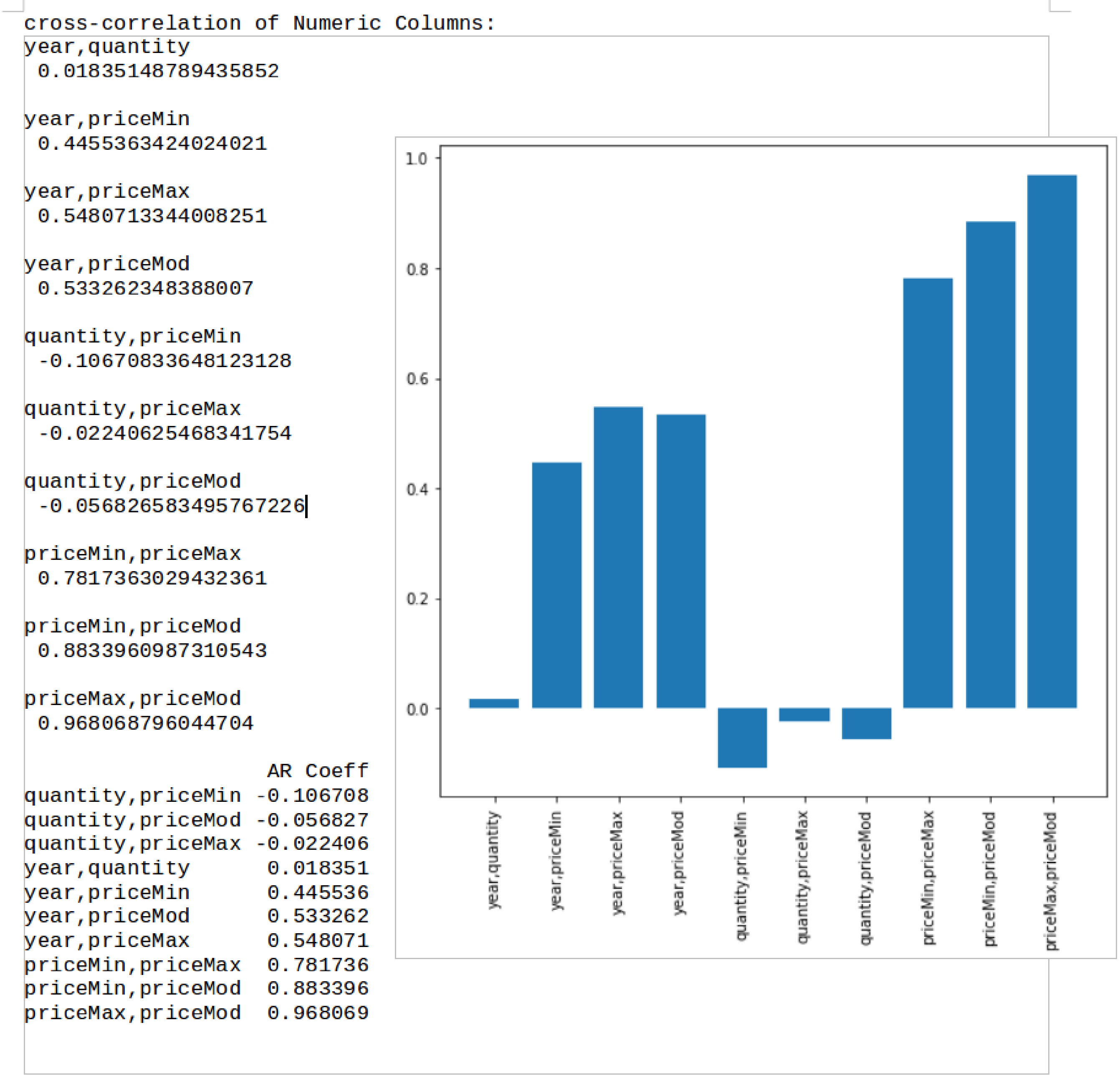
D. Indian Markets data-set

.



```
Autocorrelation of Numeric Columns:
year
 0.9999489786398241

quantity
 0.02949102997221617

priceMin
 0.5200152730340076

priceMax
 0.9040930534045767

priceMod
 0.853728771209357

          AR Coeff
quantity  0.029491
priceMin  0.520015
priceMod  0.853729
priceMax  0.904093
year      0.999949
```

Fig. 30. Plotting the autocorrelation for the numeric columns in the dataset



```
cross-correlation of Numeric Columns:
year,quantity
 0.01835148789435852

year,priceMin
 0.4455363424024021

year,priceMax
 0.5480713344008251

year,priceMod
 0.533262348388007

quantity,priceMin
 -0.10670833648123128

quantity,priceMax
 -0.022406254683417 54

quantity,priceMod
 -0.05682658349576722б

priceMin,priceMax
 0.7817363029432361

priceMin,priceMod
 0.8833960987310543

priceMax,priceMod
 0.968068796044704

                   AR Coeff
quantity,priceMin -0.106708
quantity,priceMod -0.056827
quantity,priceMax -0.022406
year,quantity      0.018351
year,priceMin      0.445536
year,priceMod      0.533262
year,priceMax      0.548071
priceMin,priceMax  0.781736
priceMin,priceMod  0.883396
priceMax,priceMod  0.968069
```

Fig. 31. Plotting the autocorrelation for the numeric columns in the dataset