

Available online at www.sciencedirect.com**ScienceDirect**

Procedia Engineering 149 (2016) 288 – 296

**Procedia
Engineering**www.elsevier.com/locate/procedia

International Conference on Manufacturing Engineering and Materials, ICMEM 2016,
6-10 June 2016, Nový Smokovec, Slovakia

Algorithmic conversion of data displayed on a weekly basis to the monthly level using the spreadsheet

Marko Martinović^{a*}, Marija Stoić^a, Miroslav Duspara^b, Ivan Samardžić^b, Antun Stoić^b

^aCollege of Slavonski Brod, Dr. Mile Budaka 1, 35000 Slavonski Brod

^bMechanical engineering Faculty in Slavonski Brod, Trg I.B. Mazuranic 2, 35000 Slavonski Brod

Abstract

This paper presents an algorithmic conversion process of data written on a weekly basis in the format of data on a monthly basis, including all the data specifics. The aim is to graphically and numerically display the results of the conversion and the size of the errors that is the time running. The main method used for averaging data is average data value. Research shows that the conversion error, ie. loss of the original value is below 2.5%.

© 2016 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

Peer-review under responsibility of the organizing committee of ICMEM 2016

Keywords: spreadsheet, data conversion, algorithm, compiling

1. Introduction

The present work describes the process of aggregation or conversion of data between the two data sets. As a source of input data for this study, the data from the Google Trends service were used. These data show trends in the use of certain keywords or phrases within the Google search engine. [1]

In certain cases there is a need to adapt the original data for input into the appropriate computer procedures and calculations [2]. If the input method for processing of data accepts the modified source data, the same data needs to be adjusted to make the smallest error. If the data are not well adapted, conversion error increases. This paper describes the process of conversion where necessary from larger, show fewer data using the method of compression or aggregation. Used aggregation functions, ie. More compression of data points in a single point between two different time frame is a weighted average.

Furthermore, there will be described a process of converting data presented with a weekly to a monthly level. For example, when calculating seasonal phenomenon frequently used data on a monthly and the original data are recorded on a weekly. Although at first glance you might not see the complexity, data conversion computer requires a set of algorithmic steps to final solution.

Source data can be retrieved in two ways, directly from the website using JavaScript code and through the "CSV" input file [5]. Data downloaded via HTML code are "finished" aggregate monthly data that GoogleTrends used to display the graph in a browser, and are still used to work for the "benchmark", ie. The accuracy of comparison between the results obtained. Another way to reach is by "CSV" file format, which is the original data displayed on a weekly basis, and used as input to the algorithm. Further, it will be made a comparison between the original and the results obtained from the evaluation of errors. As a tool for the

* Corresponding author. Tel.: +385914928010
E-mail address: mmartinovic@vusb.hr

implementation of the algorithm will use the spreadsheet "Excel", so that all of these functions are shown as steps of the algorithm.

2. Materials and methods

Format of data to be processing and display is in two forms, the first is the entry "CSV" format, and the second represents the data obtained directly from the graph web browser. Presented graph assumes monthly data using the "JavaScript", which is located in the source code shown in the website, this is the same used for the loading of data from the Web into a database. Input "CSV" formats are shown in Table 1.

Table 1: Table overview of the original form data exported in "CSV" format. [1]

	A	B
1	Interes na usluzi Pretraživanje weba: "laptop"	
2	Cijeli svijet 2004. - danas	
3		
4	Interes kroz vrijeme	
5	Tjedan,laptop	
6	2004-01-04 - 2004-01-10,54	
7	2004-01-11 - 2004-01-17,54	
8	2004-01-18 - 2004-01-24,56	
9	2004-01-25 - 2004-01-31,55	
10	2004-02-01 - 2004-02-07,53	
11	2004-02-08 - 2004-02-14 51	

Records of information within the base are stored into the rows of the table. Originally CSV means "Comma Separated Value" - values delimited with a comma. The used program for interpretation of input data and for all budgets over them is MS Excel (spreadsheet).

On the chart 1 shows the original loaded value on a weekly basis and the original monthly values obtained through java script, from January 2004 to March 2016.

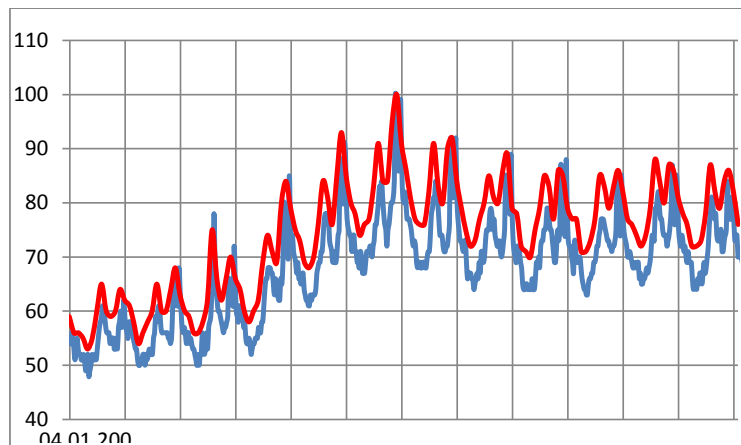


Figure 1. Display the original data from 01.01.2014. to 03.01.2016.

Weekly record is 637, and the monthly 146. They are together on the same Figure with a scale of 637, where he used interpolation monthly values between the points of the scale. The original monthly data downloaded from the website but are rounded to whole numbers for the sake of simplicity of presentation that introduces additional error. Comparison error budget will mainly be carried out on the original and the converted data, and will eventually be made, and comparison with data taken from Google Figure.

Before starting treatment, the data has to be adapted for computing. Data from the "CSV" is imported in the first column of the table. The first five lines represents a header, while from the sixth to the 643 line are the original records. According to Table 1, the structure of a single track in the column A is a form of "2004-01-04 - 2004-01-10,54", and presents a set of characters from

which it is necessary to allocate a separate variable. (For example, the last two digits represent the week value). Source data (weekly and monthly values) are normalized so that the maximum value of 100 units, while absolute values are not considered.

For recognition at the problem of conversion between two specified time frame, it is necessary to look at areas of overlap monthly and weekly intervals shown in Table 2.

In Table 2, columns B, C and D are obtained by converting the input variables of the original array. Variables B (i) and D (i) the type of date while variable average weekly value of C (i) is a real number, and the index "i" represents the row number. The area shaded with brown refers to the value for February 2004, while the area shaded in blue refers to the March of the same year. Troubleshooting procedure will be made in the case of records relating to March 2004.

Table 2. Overlapping of weeks in March 2004

<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>
<i>Column/row</i>	<i>from</i>	<i>to</i>	<i>Weekly values</i>
5	2004-01-18	2004-01-24	56
6	2004-01-25	2004-01-31	55
7	2004-02-01	2004-02-07	53
8	2004-02-08	2004-02-14	51
9	2004-02-15	2004-02-21	53
10	2004-02-22	2004-02-28	55
11	2004-02-29	2004-03-06	53
12	2004-03-07	2004-03-13	52
13	2004-03-14	2004-03-20	52
14	2004-03-21	2004-03-27	51
15	2004-03-28	2004-04-03	51
16	2004-04-04	2004-04-10	52
17	2004-04-11	2004-04-17	51

Merits of the algorithm is to calculate the true average value of the month that include average values for each week or part of week, which is located within one month and thereafter weighted values obtained in accordance with the number of days in the current month. In order to do this it is necessary to determine the correct parameters for each week; ie. whether the entire week falls within the month or the border; if the limit for how many days is in one, and how much in the second month; what is the average value of the day within the week (relying on the value of the week); with which weights weighted average, and so on.

In order to more closely define the boundary problem of weeks, it will be described in more detail in Example 3 months 2004. Line 11 is the last record of February, while line 12 represents the first week of March. The far right are the values for the reference week, so for a week 11 weeks value is 53, while for the twelve line is 52. The conceptual problem is that the week 11 physically begins in February 2004, but ends in March. Also, a similar problem is in the records between lines 15 and 16, where the fifteenth week begins in March and ends in April. Accordingly, the value of the last week in March is 51, but as the week border (stretches in the last 4 days in March and the first 3 days of April) apparently did not understand how to accurately distribute the total value.

The simplest method of conversion from weekly to monthly level is that the value of weeks to physically start in the observed month averaged, which in March amounted to $(52 + 52 + 51 + 51) / 4 = 51.5$. The presented approach will be referenced as "naive" method of conversion. This method to assume show a higher level of errors budget which will be subsequently tested.

As noted above, the fifteenth week is on the boundary of 3 and 4 months, covering the last 4 days in March (28,29,30,31) and the first 3 days of April (1,2,3). In order to properly distribute amounts between them, it is first necessary to analyse the value on a daily basis. Average daily value is calculated simply by the variable weekly value D_i divided by 7. The results are stored in the variable (column E) $E_i = D_i / 7$. For instance variables fifteenth line, this value is $E_{15} = 51/7 = 7.2857$. Far as possible the total amount of the Week (51) divided in a ratio of 4/3 between 3 and 4 months (partial value), so that the amount is merged March 29.1429 ($4 * E_{15}$), until April 21.8571 ($3 * E_{15}$). Also, this action is necessary to make and to border February and March (line 11) in the ratio 1/6 with days of February {29} / {1,2,3,4,5,6} March with amounts $(1 * E_{11} / 6 * E_{11}) = (7.5714 / 45.4286)$, using the daily value for the month of February (E_{11}). After analysis Week {45.4286; 52; 52; 51; 29.1429} and the number of days in a given week, {6,7,7,7,4} it is possible to calculate the weighted average of (partial) for the month of March {45.4286 / (31/7), 52 / (7.31) 52 / (07/31), 51 / (7.31), 29.1429 / (31/7)} = {10,258; 11,742; 11,742; 11,516; 6.581}. The sum of all partial weighting value gives the final result of the monthly average of $R = 51,839$.

To solve the problem algorithmically below will be defined by auxiliary variables as well as other needy parameters. First, it is important to note that the algorithmic problem is solved by using an iterative loop structure of iterations and the counter ("i"). Viewed from the side spreadsheet counter "i" represents a row in the table. Inside the variables are placed values, intermediate

results and are marked in capital letters that represent the columns in the table. For example, the variable M (89) = M89 = M89 with the value of counter 89 represents row 89 and column M table ..

Another important thing is the interpretation of a particular iteration in the current month. Table 3 shows that the first week in March (which is partly in February) as well as other all week are marked with the same index of the line $i = 12$ th For the purpose of resolving the month value corresponding auxiliary variables are defined, eg. F (i) and G (i), such that G (s) represents a partial value of the first, and F (s) of all other values of the weeks in the month. If this is the first record of the month the index and represents the boundary week (eg. 12). $G(12) = 45.428$, $F(12) = 52$

Table 3. Summary and description of the individual variables

	<i>B(i)</i>	<i>C(i)</i>	<i>D(i)</i>	<i>E(i)</i>	<i>F(i)</i>	<i>G(i)</i>	<i>H(i)</i>	<i>I(i)</i>
	<i>from</i>	<i>to</i>	<i>Weekly value</i>	<i>Daily value</i>	<i>full/boundary I</i>	<i>Boundary week II</i>	<i>Partial weekly value</i>	<i>Pondered weekly value</i>
5	2004-01-18	2004-01-24	56	8,000	56	0	56	14
6	2004-01-25	2004-01-31	55	7,857	55	0	55	13,75
7	2004-02-01	2004-02-07	53	7,5714	53	0	53	12,793
8	2004-02-08	2004-02-14	51	7,2857	51	0	51	12,310
9	2004-02-15	2004-02-21	53	7,5714	53	0	53	12,793
10	2004-02-22	2004-02-28	55	7,8571	55	0	55	13,275
11	2004-02-29	2004-03-06	53	7,5714	7,571	0	7,571	1,8275
12	2004-03-07	2004-03-13	52	7,4286	52	45,428	97,428	22
13	2004-03-14	2004-03-20	52	7,4286	52	0	52	11,741
14	2004-03-21	2004-03-27	51	7,2857	51	0	51	11,516
15	2004-03-28	2004-04-03	51	7,2857	29,142	0	29,142	6,580
16	2004-04-04	2004-04-10	52	7,4286	52	21,857	73,857	17,233
17	2004-04-11	2004-04-17	51	7,2857	51	0	51	11,9

Before defining themselves variables and the process of adding value in continuing to define the terms used in the work:

"Full Week" - the week with his entire interval fall within months. Start and end dates are within the current month. (Eg. Line 13)

"Marginal week I (m)" - a kind of week in which the first date of the current month (m), and last in the next (for example, line 15)

"Border Week II (m)" - a kind of week in which the first date of last month, and the last in the current. (Eg. Line 11),

"Current / Reference Week (m)" - the week that you are watching. It depends on the definition of the boundary of weeks. It is therefore appropriate to the type of week mark for that month in order to, for example, line 11 represents a "borderline week I" in reference month of February, a "borderline week II" from the perspective of March

"Week of the month (m) = T (m)" - the number of the week within a single month, which may take the numbers from the set {1,2,3,4,5,6}, where the size of 5 or 6 depending on the particular month . Weeks may be 6, which means that up to 6 different weeks can partially or completely located in the month. "Border Week II (m)" in it almost always contains the first two weeks of the month $T = \{1:02\}$ (unless the start of the week starts with the beginning of the month). For example, line 12 represents the week 1 and 2 together, the line 13, 14 and 15, 3,4 and weekly 5th

Input variables:

A (i) - the original CSV file (eg. A (14) = "2004-03-21 - 2004-03-27,51"), in which is stored the character string (text). The first part of the series until the first sign of emptiness " refers to the starting date of the week ("21/03/2004"), the second part of a series that is between characters other gaps in the series " and sign " refers to the end date of the week ("27/03/2004") and of the characters ',' until the end of the record is a series of two characters ("51"), and the value attributed to the respective week.

B (i) - the starting date of the week; - The first part of the series A (i) to sign the first gaps ". The variable type is "date". (Eg. B (14) = "2004-03-21"). Using the built-in spreadsheet functions, the value is obtained by the expression: $B(i) = MID(A(i); 1; FIND(" "; A(i)) - 1)$. Formula MID returns the text of A (i) bounded by the first character position "1" and the position of the sign of the first gap ($FIND(" ", A(i)) - 1$), then between the positions of characters 1 and 10th

C (i) - the final date of the week; - A second part of the sequence (i) the sign of the other gaps ", to the character " is a variable of type "Date". (Eg. C (14) = "2004-03-27"). The value is calculated by the expression: C (i) = MID (A (i), FIND ("" ; A (i)) + 3; FIND (" ; A (i)) - (FIND ("" ; A (i)) + 3)). MID formula returns the text A (i) bounded by the second position the gaps "14" (FIND ("" ; A (i)) + 3) and the point position of the sign "24", ie. The length of a series of dates FIND ("" ; A (i)) - (FIND ("" ; A (i)) + 3).

D (i) - the original value for the week and the index (eg. D (14) = 51) is part of a series between the point and the end of the string and an integer. Application original value on a weekly basis over that account monthly level. The length of the string may be 2 or 3 (if it is a value of 100). After extraction string value needs to be converted into a number. (Formula VALUE ()). The final value is calculated by D (i) = s (MID (A (i), FIND (" ; A (i) , 1) +1; len (A (i)) - FIND (" ; A (and))))). Position point is FIND (" ; A (i) , 1) +1 (npr.24), the number of length len (A (i)) - FIND (" ; A (i))) (example 2), where the function LEN () calculates the total length of sequence A (i) in the signs, (eg. 26).

Table 4. Summary and description of the individual variables

	<i>B(i)</i>	<i>C(i)</i>	<i>J(i)</i>	<i>K(i)</i>	<i>L(i)</i>	<i>M(i)</i>	<i>N(i)</i>
	<i>from</i>	<i>to</i>	<i>year</i>	<i>month</i>	<i>Day in month</i>	<i>Nr of day marginay/boundary II</i>	<i>Nr of day boundaryi II</i>
5	2004-01-18	2004-01-24	2004	1	28	7	0
6	2004-01-25	2004-01-31	2004	1	28	7	0
7	2004-02-01	2004-02-07	2004	2	29	7	0
8	2004-02-08	2004-02-14	2004	2	29	7	0
9	2004-02-15	2004-02-21	2004	2	29	7	0
10	2004-02-22	2004-02-28	2004	2	29	7	0
11	2004-02-29	2004-03-06	2004	2	29	1	6
12	2004-03-07	2004-03-13	2004	3	31	7	0
13	2004-03-14	2004-03-20	2004	3	31	7	0
14	2004-03-21	2004-03-27	2004	3	31	7	0
15	2004-03-28	2004-04-03	2004	3	31	4	3
16	2004-04-04	2004-04-10	2004	4	30	7	0
17	2004-04-11	2004-04-17	2004	4	30	7	0

Among the variables:

L (i) - the number of days in the current month to the start date of the week - integer. (Eg. L (14) = 31). For the purpose of calculating the weighted mid important to know how many days there this month. The number of days refers to the variable B (i), which defines the beginning of the track. It is calculated by the formula EOMONTH (x, y) which returns the last day of the month to date ("x" plus or minus the number of months "y") For example EOMONTH (04.15.2016.; 0) = 30.04.2016 and EOMONTH (15.04. 2016.; - 1) = 31.03.2016. If we subtract one date from another (operator forfeiture account the number of days between two dates), we get the number of days that lies between them, and he just represents the number of days in the current month. So that the final expression is: L (i) = EOMONTH (B (i) 0) -EOMONTH (B (i) - 1).

M (i) - the number of days of the week of the current month - whole number. It describes the number of days of the week that is attributable to the current month. Applies only kind of "full week" and "borderline week I (m) '(variable value for' border Week II (m) is 0)" full week "value is 7, while the "border week I (mj)" depends on the number of days that you can be classified within the observed month - "month". (Eg. M (15) = 4, then the first four days of the present track located in March). In the case of "border week I" value is calculated so that the end date of the week (which is located in the following month) subtracting the last date of the current month and the number 7 is reduced to this value. For example, M (15) = 7 - ("03.04.2004" - "31.03.2004") = 7-3 (week three days in April are therefore left for 4 days), where a subtraction operator calculates the number of days between date. If the first and last month the same week, then the value M (i) = 7 (for example, M (13) = 7) Final formula:

$$M (i) = IF (MONTH (B (i)) > MONTH (C (i)); 7 - (C (i) -EOMONTH (B (i) , 0)); 7)$$

N (i) - the number of days a week for the next month - whole number. It describes the number of days of the week that are attributable to the following month. Applies only kind of "border Week II (m)", while for other types of week value is 0. "border Week II (m)" value depends on the number of days that can be classified into the following month - "m + 1". (Eg. N (15) = 3, then the last 3 days week starting in March (m), and ends in April (m + 1)). The value is calculated so that the end date of the week (which is located in the following month) take away the last date of the month. For example, M (15) = "03/04/2004" -

"31/03/2004" = 3, where an operator forfeiture account the number of days between two dates. If the first and last month the same week, then the value $M(i) = 7$ (for example, $M(13) = 7$) Final formula:

$$M(i) = C(i) - \text{EOMONTH}(B(i), 0);$$

Variables:

D(i) - the original value of the index and weeks (eg. $D(11) = 53$)

E(i) - the daily average value, the fact that the $E(i) = D(i) / 7$ (ie. $E(11) = 53/7 = 7.571$), and 7 represents the number of days a week.

F(i) - the original value of "full week" or partial amount "border week I (m)". It refers to all the weeks of the month T, except for the first week (eg. $F(12; "Oz") = 52$; $F(15; "Oz") = 29,142$)

G(i) - the partial amount "border Week II (m)". Applies only to the first week in the month, $T = 1$ (eg. $G(12; "Oz") = 45,428$; $F(13,14 \text{ and } 15; "Oz") = 0$)

M(i) - the number of days' border week and "entering the current month

N(i) - the number of days' border week I' exceeding the next month or

N(i-1) - the number of days' border Week II "that are included in the current month

$N(i) = -M(7(i))$

H(i) - the partial value of the average week;

D(m, i) - a final average value $D(m, i) = H(s) + H(i+1) + \dots + H(i+n-1)$,

$N \in \{1,2,3,4,5\}$ - the number of records for the current month, m - current month

$m \in \{1,2,3 \dots 12\}$ for which the averaged,

and E and index - hands in the database on the first record of the current month, and the integer.

For the weekly intervals with their whole fall within the month ("marginal week") to the average value of the week is taken the value of $D(i)$ of that week, $H(i) = D(i)$.

For the weeks in which the starting date is within the current month, and the final the following month ("borderline week I"), partial week average value of the current month (part of the average value) includes all days of the week within one month from the set $\{1,2, 3,4,5,6\}$. The formula for calculating the partial average value is $H(s) = E(i) * M(i)$, where $M(i)$ of $\{1,2,3,4,5,6\}$, and $E(i)$ is the mean daily values of the current week.

For the weeks in which the commencement date for the week was the finding last month, and the final date of the current month ("border Week II"), partial average value is calculated based on the number of days of that week falling within the current month $N(i-1) \in \{1,2,3,4,5,6\}$, the formula $H(i) = E(i-1) * N(i-1)$, where $E(i-1)$ represents the mean daily values for the previous week.

In other words the amount that enters the current month is calculated by the data the week that started last month.

To realize the above algorithmic expressions within one iteration, the variables used are $F(s)$ and $G(i)$,

where $H(s) = F(i) + G(i)$

If n is the number of records (rows) in the month, while the index and shows absolutely in the database on the first record of the current month (eg. From March 2004 and is = 12 and $n = 1.4$), then is

for „marginal weeks“

$$F(i+n-1) = D(i+n-1) = 7 * E(i+n-1)$$

$$G(i+n-1) = 0$$

za, $n \in \{1,2,3,4,5\}$ – number of rows for current month

for „boundary weeks I“ .

$$F(i+N-1) = E(i+N-1) * M(i+N-1),$$

$$G(i+N-1) = 0$$

$N = |n| = \{1,2,3,4,5\}$ - size of collection

$$M(i) \in \{1,2,3,4,5,6\},$$

for „boundary weeks II“, ($n=1$) result has been recorded into variable $G(i)$ with equation:

$$G(i) = E(i-1) * N(i-1),$$

$$F(i) = E(i) * 7 = D(i)$$

$$N(i) \in \{1,2,3,4,5,6\}$$

Where $H(i+n-1) = F(i+n-1) + G(i+n-1)$ represents of partial weekly value of average and is calculated as for all , $n \in \{1,2,3,4,5\}$

The final results of the conversion of R (m,i) obtained final partial weighting value $H(i)$, in accordance with the number of days of the week that are in the current month.

Basic equation for average

$$R = \frac{w_1 \cdot x_1 + w_2 \cdot x_2 + \dots + w_t \cdot x_t}{w_1 + w_2 + \dots + w_t}, [3]$$

weight $w_1 \dots w_n$ represent the proportion of the number of days the week of the current month (m), while $t \in \{1,2,3,4,5\}$, the number of the week. The sum of all the weight is equal to one (the sum of all the days of weeks must equal the number of days of the month $L(i)$). For all weekly full weight w is $W(i) = 7 t / L(i)$, where $L(i)$ indicates the number of days of the current month. For "border week II" (the first week) week weight is $w(i) = 1 N(i) / L(i)$, and the "border week I" (last week) $w(i) = T M(i) / L(i)$ and $N(i)$ and $M(i)$ represent the number of weeks given limit ("border week II" and "border week I") which fall within the current month. The values $x_1 \dots x_t$ represent weekly value $D(i)$

When the result of the expression is simplified:

$$R(m) = \frac{7}{L(m)} \cdot [H(i) + H(i+1) + \dots + H(i+1-n)], n \in \{1,2,3,4,5\}$$

, which is a formula of the algorithm for converting from which it can be seen that the final result represents the sum of the partial value of weeks in the month multiplied by a factor of $7 / L$.

3. Results and discussion

The results of the conversion process of input (on the Figure 2 "orig.") to output data using an algorithm conversion (the Figure 2 "Average") are shown in Figure 2

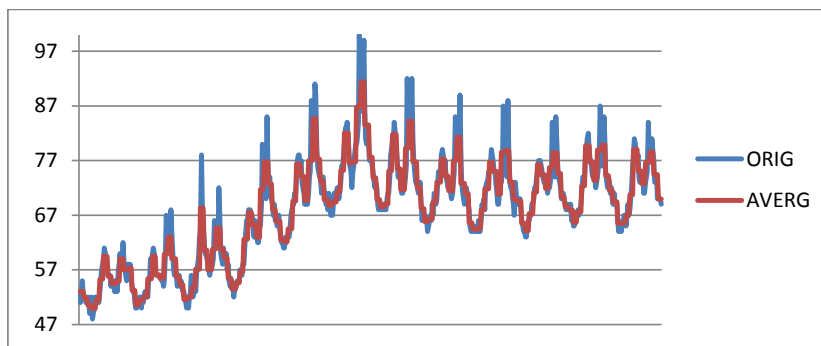


Figure 2. Comparison of original and converted data.

Figure 2 shows the original input value weeks and converted the monthly value data. One can see that the Figure calculated monthly value fits into the original value and that the extremes smoothed.

In order to detail could see how "good" it fitting, calculated the level of errors. Monthly data represent average weekly value in a defined time frame (calendar month). How many weeks passes under a time frame months for calculation errors need to compare the data with the weekly average monthly data. For example, if the weekly value of $\{53,51,53,55 \dots\}$, and the average weighted monthly value is 53, then for each week in the month need to compare the difference on a weekly basis. $\{53-53; 51-53; 53-53; 55-53; \dots\}$ and use it to calculate the absolute, square and relative errors. Since it is a comparison within the time frame of one month that cutting boundaries of individual weeks, it is necessary to translate source data into comparative domain with the results.

For comparison, the degree of integration of the data obtained, the method of conversion was compared with two other methods; "Naive approach" and "java script method".

"Naive method" ignores the border and simply account weekly average weekly value whose start dates are within a single month. While "java script method" takes the original monthly data drawn from web sources.

In the Figure 3 are buddy comparisons of these methods for the first four years of the observed. Most of all methods overlap with minimal fluctuation, while data obtained with "java script method" something more distorted.

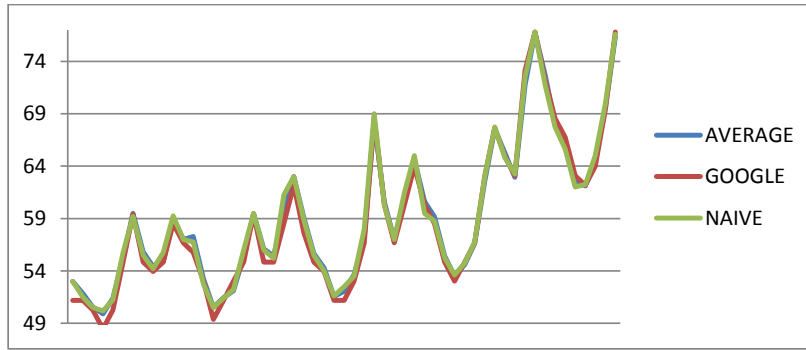


Figure 3. Comparison of methods for data from 2004 – 2008 year

In order to obtain the true image conversion, ie, the degree of switching, calculated the error of conversion for all three methods. On the error of conversion can be seen as a loss of information that results summarizing data from a number of data on a small number, which in turn constitute an error compression.

The table below shows the absolute amounts, square and relative errors. [4]

Table 5 Error display conversion of three methods

	AVE RG	NAI VE	GOOG LE
MAE	1,697	1,731	1,815
MSQ			
W	6,614	7,434	7,354
MAP		2,39	
E	2,45%	%	2,63%

It is to expect that the conversion method described in this paper gives the best results as the most accurate account average, which was proved to be true at the level of absolute and square error, but surprising is that the relative smallest mistake to say "naive method".

The interpretation is that the average data set is not an optimal solution for reducing errors.

Figure 4 shows points (nodes of the Figure) and a horizontal line. If it is assumed that the red line is the average, nodes weekly value (blue), and the x-axis (1-5) monthly interval, the conversion error is the sum of the difference between the nodes and the horizontal lines. However, the average horizontal line is not necessarily the line that says the smallest mistake.

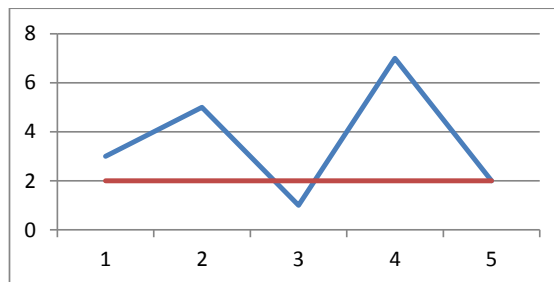


Figure 4. Show of data and average value

For a given set of points there is a horizontal line L with the smallest error. If P is average then the $L < P$ or $L = P$. Therefore, it is possible that there is a method that gives a better performance than the existing method of calculating the weighted average ("average"), so that the used method is sub-optimal. Interestingly, the "naive method" which contains additional error inputs (data not comply with the weeks to outdo the other months, so there is a random component) gives better results relative error of the "correct" method (in which there is no such error). Also, it is noted that the direction to which refers the accuracy of the model is horizontal and is a special case of linear regression with the amount of $\beta = 0$; ($Y = \beta * x + \epsilon$), which in most cases it is not (because there is a component of $\beta > 0$, and is not shown in this model). But even if the $\beta = 0$ it is possible to optimize the error

with a parameter $\varepsilon \ll \varepsilon_0$, where " ε_0 " is average. Leaving further room for improvement in conversion using linear (linear regression $\beta \ll 0$) and non-linear models of interpolation between the data points which will increase the accuracy of the model.

4. Conclusion

The algorithm and method of converting data recorded on a weekly basis in the month can be applied to any case in which the data were presented with clearly defined intervals. Although this is a specific conversion process, it is possible with minor adjustments applied algorithm for general conversion between any two data frames D_1 and D_2 , where $D_1 < D_2$, whether it's the weather or any other domain. Therefore, the application of the algorithm can be wide. Overall, the algorithm can be seen as a method of data compression with the loss of information. The efficiency of the algorithm is higher if the ratio of the size of data that you can summarize and error that occurs k'tome higher.

The results are good "fit" in the original data and information loss below 2.5%. But it turned out that there is a more optimal solution, ie. That the conversion error can be even smaller, which leaves extra space for further research. Furthermore, it is possible to continue research by constructing nonlinear models that contributed to a further reduction of errors.

References

- [1] Google Trends, *Keyword „Laptop“*, <https://www.google.hr/trends/explore#q=laptop>, March 2016 .
- [2] M. Martinović , I. Samardžić I., A. Stoić "*Analysis of laptop demand curve through the internet search*", 4th international conference Inovations, Technology, Education and Management, 2016
- [3] D. Terr, „*Weighted Mean*“, <http://mathworld.wolfram.com/WeightedMean.html>, [2016]
- [4] Absolute and Relative Error, <http://www2.phy.ilstu.edu/~wennings/slh/Absolute%20Relative%20Error.pdf>, [2016]
- [5] Hooijmans M., „*Google Trends: How to Extract Monthly Data*“ <http://www.ommax-marketing.com/blog/google-trends-how-to-extract-monthly-data/> [20.10.2014]