



# Analysing Data with the R Programming Language to Control Machine Operation

Marek KESEK<sup>1)</sup>

<sup>1)</sup> AGH University of Science and Technology, Faculty of Mining and Geoengineering, 30 059 Kraków, Poland; e-mail: kesek@agh.edu.pl

<http://doi.org/10.29227/IM-2019-01-42>

Submission date: 11-07-2018 | Review date: 02-05-2019

## Abstract

*This article describes the benefits offered by the analysis of data from production processes. With the correct processing, such data allows issues to be identified both within the analysed process and in the ways machines are used. The paper presents an initial analysis of data from the fragmentation process in a hard coal mine using longwall shearers. The analysis is described using R programming language functions.*

*Keywords: data analysis, R language, analysing machine operation*

## Introduction

For many years now, effective management of any company has relied more and more extensively on quantitative methods. Analyses of large amounts of data, modelling, and simulations are becoming the basis for informed decision-making like in [3,4,5,6]. Such calculations have become widespread and easy to make as a result of the tremendous advancement of computer technologies.

By identifying issues in their production processes, companies can reduce production costs through avoiding long downtimes, reducing repair costs, enforcing and maintaining discipline among employees, and optimising machine uptime to ultimately enhance production processes. Appropriate lessons learned from the analysis of production-process data can also be the basis for the adjustment of machine operation procedures and troubleshooting.

IT personnel, who usually are well-versed in the details of individual processes, can develop software to control and analyse machines' and people' work. With contemporary programming languages, it is possible to quickly create applications that offer such functionalities, and if they use server databases which store information also from outside the controlled area, this facilitates a more general analysis of correlations between issues identified in production processes. In-depth analysis of production processes and the knowledge it offers can provide a company with competitive advantage, which is why businesses should employ advanced computing technologies to optimise their processes. [7]

The R programming language, which has been designed to support statistical computations, has recently been upgraded to handle data exploration. Examples of such computations were presented by [9].

Below you will see how the operation of a longwall shearer can be analysed using the R programming language by examining the electrical current drawn by its motors. This analysis is part of a larger system developed to support decision-making on the basis of an expert system. For more information on the system, see [1,2].

## R programming language

R is an R software environment programming language and is designed mainly for the purposes of statistical computations. It is available under the GNU General Public Licence, so it is completely free of charge. Furthermore, as an open-source project, it is continuously expanded to include libraries of new functions available on the Internet as packages. Many users of this language claim that it can be used to develop any program, but what is of greatest benefit for analysts is that it supports modelling, testing, classification, clustering, time-series analysis, etc. What is also important to note are its capabilities related to the production of a wide range of high-quality graphs.

Another asset of the R software environment is its open-source package repository (Comprehensive R Archive Network, CRAN). It is a collection of libraries created by users from all around the world and includes relevant documentation. Commercial equivalents of such packages can be worth at least tens of thousands of zlotys. Anyone can upload their library to that repository to make it available to other people. With interfaces to many other programming languages, code for such additional libraries can be written not only in R, but also in Java, C or Fortran. [10]

R can be used to organise data in various structures, the simplest of which is a vector, i.e. a series of numbers arranged in a specific order. There are no simpler structures in R, so a single number is a vector with a length of 1. [11]

## An analysis example

The example provided below shows the basic analysis of the operation of a longwall shearer on the basis of the current drawn by its four motors. A similar analysis, carried out on the basis of a spreadsheet and the VBA language, was provided by [8]. Shearer motors are responsible for its movement, coal-cutting action, and power-cable handling. Their symbols are presented in Table 1 below.

The electronic system on board of the shearer measures the current drawn at each of the four sensors every second. Then these data are immediately communicated to the database, from which it can be analysed with a negligible delay.

Tab. 1. Shearer motor symbols. Source: Author  
 Tab 1. Oznaczenia silników kombajnu

Motor	description	No.
M1	Coal-cutting component	2992
M2	Cable-carrier tensioner	2919
M3	Main drive	2871
M4	Auxiliary drive	2895

The data are communicated as three key-value pairs, including motor number, reading date and time, and current. In order to avoid unnecessary data recording during shearer downtime, a solution was adopted to communicate data only if the current changed between 1-second intervals. This helped reduce the number of recorded data. When a motor is turned off, its current is zero and is recorded as such in the database with its timestamp. The next piece of data will be recorded for a time when the measurement is greater than zero, so that no data is recorded for motor downtime. Any downtime can be computed on the basis of the difference between consecutive timestamps. To analyse machine operation, an additional algorithm has to be used to identify and parse any gaps in changes to current values.

#### Importing data to the R programming language

Originally, measurement data is stored as csv (comma separated values) files, in which, as its name suggests, the data has the form of rows with values separated by commas.

In the R programming language, such data can be imported using the function `read.csv`:

```
data <- read.csv(file="prady1.csv",sep=";", header=TRUE, dec=".")
```

This imports the data from the csv file into computer's operating memory as a data frame named "data".

#### Data decompression

Data can only be analysed after they are converted back into their original form. This is achieved through a function designed to read each row from the data array and complete it with subsequent rows containing timestamps (at one-second intervals) and the last recorded current. This action is performed as many times as indicated by the time between the consecutive rows of compressed data. This cycle is repeated for the time interval specified in the function parameters. The other parameters of the function, in addition to the start and end times, include data frame name and the name of the motor for which the data is to be decompressed. The signature of this function is presented below with comments.

```
decPrady <- function(dt, motor, st, en){
  'dt – compressed data frame
  'motor – motor number
  'st – start time
  'en - end time
  'filtering data that meet the time and motor number
  criteria
  dx <- filter(dt,time>=as.double(strptime(st,for-
```

```
mat="%Y-%m-%d %H:%M:%S")),
  time<=as.double(strptime(en,format="%Y-%m-%d
%H:%M:%S")),motor_no==motor)
  'reading the number of data that meet the criteria
  cx <- as.integer(count(dx))
  'preparing a vector for the loop
  x <- c(2:cx)
  'reading the first time value
  k <- dx[1,2]
  'preparing a current vector
  nt <- dx[1,3]
  'preparing an empty data frame
  d <- filter(dx,time==0)
  'looping through filtered data
  for(i in x){
    'preparing a vector with missing timestamps
    c1 <- (k:dx[i,2])
    'eliminating the first element in vectors with more than
    1 element
    if (length(c1)>1) c1 <- head(c1,-1)
    'timestamp loop
    for(j in c1) {
      'inserting parsed time and current values into the
      prepared data frame
      d <- rbind(d, data.frame(motor_no=motor,time=-
      j,electr_curr=nt))
    }
  }
  'returning decompressed data frame
  return(d)
}
```

Data is decompressed when the function is called with accurate arguments:

```
d <- decPrady(data, 2992, "2013-04-02 10:30:00",
"2013-04-02 11:21:00")
```

The resulting data frame `d` contains data about the current drawn by motor 2992 every second.

#### Analysis of electrical current

The R programming language makes it possible to create box plots, which can be used to identify the basic characteristics of data. The lower side of the rectangle corresponds to the first data quartile (Q1), the line inside the rectangle corresponds to the median (Q2), and the upper side of the rectangle is the third quartile (Q3). Below and above the rectangle there are whiskers, which indicate the limits beyond which outliers can be identified. The ends of the whiskers are determined as

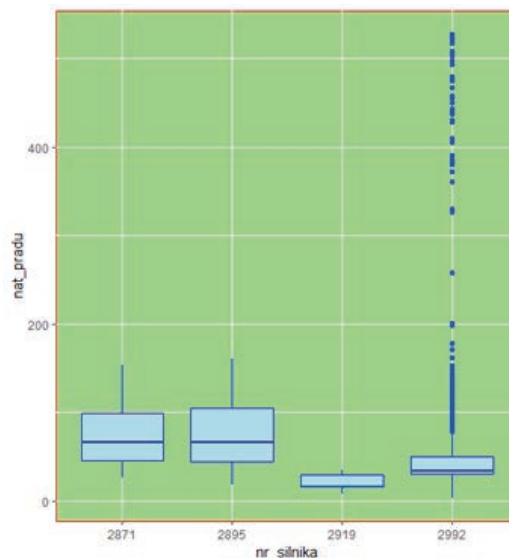


Fig. 1 Current distribution. Source: Author [x – motor\_no., y – electr\_curr]  
Rys. 1. Aktualny rozkład

follows on the basis of the interquartile range (IQR):  
start of the upper whisker:

$$W1 = Q1 - 1.5 * IQR,$$

end of the upper whisker:

$$W2 = Q3 + 1.5 * IQR,$$

where  $IQR = Q3 - Q1$

If the data contains values lower than W1, or greater than W2, these are treated as outliers and should not be taken into account during the development of the model, but, on the other hand, should be analysed to explain them.

Fig. 3 shows a box plot for the distribution of recorded current drawn by machine's motors. This data visualisation accounts only for values greater than zero.

The graph was created in the R programming language, using the ggplot function.

```
ggplot(data=filter(data2,electr_curr>0))+geom_boxplot(mapping=aes(y=electr_curr, x=motor_no., group=motor_no))
```

Electrical current outliers can be observed (as individual points above the upper whisker) for the motor that powers the coal-cutting component (2992). Since such large current can cause damage, it is important to analyse what situations caused these surges to occur.

To identify outliers, the global dataset (exclusive of zeroes) needs to be processed to identify any data with current greater than W2. In the R programming language, this can be done as follows:

```
'filtering the data to identify current values greater than zero
data2992gz <- filter(data2992,electr_curr>0)
```

'calculating W2

```
W2 <- quantile(data2992gz$electr_curr,0.75)+1.5*IQR(-
data2992gz$electr_curr)
```

'filtering the data to identify current values greater than W2

```
data2992ggw <- filter(data2992,electr_curr>W2)
```

As you can see in the graph (Fig. 1), the continuity of current values is disrupted as values reach around 200A. Therefore, for the purposes of further analysis, time data for values greater than 200A was identified.

```
t <- select(filter(data2992,electr_curr>200), time)
```

The identified times are the basis for finding in the primary data the currents at times immediately (by a second) preceding values greater than 200A.

```
filter(data2992,time %in% (t[,1]-1), electr_curr>0)
```

The resulting dataset contains only the data where current is equal to zero. This means that the data which were identified during the previous analysis as outliers are recorded when motor starts.

As a result, the electrical current during motor start was examined. This allowed the identification of the times when current at time t was equal to zero, but at time t+1 was greater than zero:

```
k <- select(filter(data2992,electr_curr==0, data2992[-
data2992$X+1,4]>0),time)
```

and then to identify current at times t+1

```
starts <- filter(data2992,time %in% (k[,1]+1), electr_
curr>0)
```

The resulting dataset was the basis for the creation of a histogram (Fig. 2) which presents the distribution of current



Fig. 2. Current distribution as motor starts. Source: Author [electr\_curr starts]  
Rys. 2. Aktualny rozkład w momencie startu

values during the first second after the motor starts.

```
ggplot(data=starts)+geom_histogram(mapping=aes(x=
start$electr_curr), binwidth=25)
```

The histogram presented in Fig. 2 shows two groups of data. The current values observed during motor start can be grouped into two ranges – up to around 150A and above 300A. There are also single data items between these two ranges, but these are marginal cases. Such distribution of data suggests two types of motor starts – with and without a load.

- softstart – without a load, is characterised by low motor current values,
- hardstart – with a load, characterised by high motor current values, which makes them undesirable, as they cause a significant increase in machine wear and failure rate. [Kesek, Zagórska, 2015]

### Summary

Under current economic conditions, the accurate assessment of production processes is vital. It can help maximise machine use, reduce downtime, bring down repair costs, and,

consequently, lower production costs and increase revenue by boosting production over time. With the right computer tools to analyse database input and process such data, production processes can deliver valuable knowledge.

While initially perhaps incomprehensible, the ways to use R programming language functions described in the article quickly become clear and useful, and the opportunities for automating such computations prove instrumental. What is also great about this language is its graph-generating capability, which can be tailored to one's needs with additional commands. Another important functionality of R is its communication with databases through the ODBC interface, which makes it useful for analytical purposes in complex database systems.

The performed analysis produced a dataset with critical cases where electrical current during motor start was high. These findings can be used to individually examine each case identified during the analysed time period.

This paper was supported by AGH University of Science and Technology [nr 11.11.100.693].

## Literatura – References

1. Brzychczy E., Kęsek M., Napieraj A., Magda R. (2017): An expert system for underground coal mine planning - Mineral Resources Management, vol. 33, pages: 113 – 128.
2. Brzychczy, E., Kęsek, M., Napieraj, A., Sukiennik, M. (2014): The use of fuzzy systems in the designing of mining process in hard coal mines - Archives of Mining Sciences, 59 (3), pages: 741-760.
3. Fuksa D.: Consolidation of mining enterprises as a chance to make the Polish mining industry profitable. Inżynieria Mineralna. 2018 R. 20 nr 2, s. 47–54
4. Fuksa D.: Metoda oceny wpływu zmiennego zapotrzebowania odbiorców węgla na rentowność wielozakładowego przedsiębiorstwa górnictwo. Inżynieria Mineralna. 2017 R. 18 nr 1, s. 249–256.
5. Kęsek M., Adamczyk A., Kłaś M.: Computer simulation of the operation of a longwall complex using the “Process Flow” concept of FlexSim software, Intelligent Systems in Production Engineering and Maintenance, Springer Nature Switzerland AG, ISSN 2194-5357 ,vol. 8351, pages: 97–106, 2019
6. Kęsek M. (2017): Visual Basic as a Tool for Monitoring and Analyzing Machines, Inżynieria Mineralna - Journal of the Polish Mineral Engineering Society, No. 2, pages: 195–200, DOI: 10.29227/IM-2017-02-21.
7. Kęsek M., Lach K. (2017): Kontrola pracy kombajnu ścianowego z użyciem Visual Basic for Applications, Przegląd Górniczy; ISSN 0033-216X. t. 73 nr 9, pages: 6–10.
8. Kęsek M., Zagórska J. (2015): The engine load analysis of extracting-loading head as an element of evaluating its usage correctness, Logistyka; ISSN 1231-5478 nr 4, pages: 9124–9129.
9. Kęsek M. (2010): Language R in mining knowledge discovery, Przegląd Górniczy; ISSN 0033-216X. t. 66 nr 9, pages: 188–191.
10. Komsta Ł.: Wprowadzenie do środowiska R. publikacja internetowa, <http://cran.r-project.org/doc/contrib/Komsta-Wprowadzenie.pdf>, 2004.
11. Kopczewska K., Kopczewski T., Wójcik P.: Metody ilościowe w R. Aplikacje ekonomiczne i finansowe. Wydawnictwa Fachowe CeDeWu, Warszawa 2009.

### *Analiza danych z wykorzystaniem języka R w kontroli pracy maszyn*

*W artykule przybliżono korzyści płynące z analizy danych pochodzących z procesu produkcyjnego. Odpowiednia ich analiza pozwala na wykrywanie nieprawidłowości zarówno w realizowanym procesie, jak i w sposobie użycia maszyn. Zaprezentowano wstępną analizę danych pochodzących z procesu urabiania w kopalni węgla kamiennego za pomocą kombajnu ścianowego. Przedstawiono sposób prowadzenia analizy z wykorzystaniem funkcji języka R.*

*Słowa kluczowe: data analysis, R language, analysing machine operation*