



The K-means Grouping Method as a Mean to Control the Performance of the Production Process

Marek KĘSEK¹⁾

¹⁾ AGH University of Science and Technology, Faculty of Mining and Geoengineering, 30 059 Cracow, Poland; email: kesek@agh.edu.pl

<http://doi.org/10.29227/IM-2020-01-83>

Submission date: 19-12-2019 | Review date: 28-01-2020

Abstract

The paper presents a concept of using clusters of objects using the k-means method to control the performance of the production process, which runs under variable conditions. The distribution of the production process performance in production cycles grouped according to similarity is the basis for controlling the performance of subsequent production cycles. The practical part of the paper contains an example of calculations carried out according to this concept using the VBA and R languages, and is related to the bolting process in underground mines.

Keywords: production process performance, clustering, k-means, production cycle, R language, VBA, process mining, bolting

1. Introduction

Planning of mining production is an extremely complex process. The specific nature of mining production means that not only performance indicators related to the production process or economic factors must be taken into account, but also environmental factors [14]. On the other hand, it is the environment, among other things, that ensures that the conditions for production process are never fully repeatable [1]. Technical conditions related to the equipment, environmental conditions related to the strength of the rock mass and conditions related to the employees' competences may vary, and there may be many other differences, even those that may not have been identified yet. Therefore, in addition to standard analyses, attempts can be made to analyse the production process itself and, based on its course, to predict the effects achieved.

Quantitative methods play an increasingly important role in an effective management of a mining company. The analysis of large amounts of data [2], modeling and simulations of production processes in mining [9, 10] become the basis for rational decision-making, e.g. in the works [4, 5, 6]. The universality and ease of performing such calculations has become possible as a result of the enormous development of computer methods of data processing and the associated development of calculation software.

The calculation part of the article presents the concept of using the clustering method for ongoing control of the performance of implemented processes based on the previously collected data.

The adopted concept assumes that an appropriate partition into groups of previously implemented production cycles makes it possible to show differences in the production results achieved in them. In addition, it assumes that an object assigned to a group of similar objects will achieve similar values of production performance indicators to other objects from the same group.

This requires prior clustering of objects into mutually unlike groups of objects similar to each other. The scope of use of

taxonomic methods, the possibilities of their application and ordering, grouping procedures, selection of representatives, aggregate variable structures, etc., used in taxonomic calculations, are widely discussed in the literature, while this paper is limited to those that are necessary to present the proposed concept.

2. Methods and tools used

2.1. The k-means method

The adopted concept assumes the clustering of data stored in the computer system. In the numerical example presented later in the article, the k-means method based on the k-means clustering was used [7]. In this method, a priori, a predetermined number of clusters is given and subsequently the cluster centers (centroids) are determined. Centroids can be determined arbitrarily or randomly. In many cases, the algorithm is carried out repeatedly while the results are observed, and the best model is selected subsequently. In the next step the distance of subsequent objects to be clustered from centroids is to be calculated. Distances can be calculated based on various metrics (Euclidean, Manhattan, Chebyshev, etc.), which most often originate from Minkowski metric (formula No. 1).

$$d_{ik} = \left[\sum_{j=1}^m |x_{ij} - x_{kj}|^p \right]^{\frac{1}{p}} \quad (i, k = 1, \dots, n) \quad (1)$$

where:

x_{ij}, x_{kj} – implementations j-identifier for i-entity k-entity,

n – number of entities

m – number of identifiers,

p – natural number,

In practice it is used:

Metric space ($p = 1$),

Euclidean space ($p = 2$),

Chebyshev metric ($p \rightarrow \infty$),

in the latter case, the metric comes down to a formula:

$$d_{ik} = \max_j |x_{ij} - x_{jk}| \quad (i, k - 1, \dots, n) \quad (2)$$

The next step of the algorithm is to assign an entity to a cluster for which its distance from the centroid is the smallest. Assigning an entity to a cluster changes the position of its centroid, and calculating its new position is the next step of the algorithm. The above steps are then repeated for the subsequent entities to be clustered.

2.2. IT tools

The calculations were made using R and VBA languages.

The R language is a programming language for the R environment, mainly dedicated to statistical calculations and data mining [3]. It is distributed under a GNU license, so it does not require any fees. Furthermore, as an "open software" it is constantly being expanded with new libraries of functions available on the Internet in the form of packages. Many users of the language claim that it can be used to write any program, but its biggest advantage for analysts is that it supports issues such as: modeling, testing, classification, clustering, time series analysis, etc. The possibilities of creating a wide range of high quality charts are also worth emphasizing [11].

R allows organizing data into various structures, and the simplest of them is a vector. This is a series of numbers in order. There are no simpler structures in R language; therefore, a single numeric is a vector with a length equal to 1 [13]. Vectors can be combined into lists or matrices, matrices can be transformed into data frames, and for each of these structures appropriate operators and functions are provided to enable them to perform calculations on them. It is also important that this language is constantly evolving. New function libraries collected in the open, free package repository (CRAN, Comprehensive R Archive Network) are constantly being published. This is a collection of function libraries

with documentation created by system users from around the world.

The Visual Basic for Application (VBA) language is part of the Microsoft Visual Studio development environment and is also a Microsoft Office tool. The success of this language is due to its simplicity, transparency and flexibility with a wide range of possibilities. Therefore, it is used by people with high programming skills as well as by beginners. AutoDesk – a manufacturer of software supporting engineering design, has also noticed the advantages of this language. For its most important product, namely AutoCad, from the 2015 version it provides a module that allows using the Visual Basic language working with this program. [12].

Programming in Visual Basic is an event programming. This means that the program code is called as a result of an event, e.g. pressing a key, pressing a button, selecting from a list, etc. Such an event triggers the procedure that supports it, followed by a return to waiting for the next initiating event [15]. This language is a structural language with sequential processing, in which extracted structures include fixed variables, arrays, functions, subprograms, conditional operations and loops. Visual Basic, after attaching the relevant libraries, has the ability to connect to MSSQL databases and query them. The responses received from the database are then processed accordingly.

3. Example of calculation

As part of the research related to the implementation of the smartHUB project, as one of many tasks, data on the implementation of the bolting process was analyzed. The data was recorded automatically and contained the duration of individual activities related to the bolting process. In the analysed period of time, 71205 activities were registered within 854 cycles. A segment of the analysed data is shown in Fig. 1.

A	B	C	D	E	F
end	start	duration [sec]	activity type	shift id	cycle id
31-07-2019 22:20:05	31-07-2019 22:19:14	52	TRAVELING	2019-08-01_3.M10401	
31-07-2019 22:20:09	31-07-2019 22:20:06	4	IDLE	2019-08-01_3.M10401	
31-07-2019 22:20:28	31-07-2019 22:20:10	19	HOLE_SETUP	2019-08-01_3.M10401	A4612194.0.M10401
31-07-2019 22:20:50	31-07-2019 22:20:29	22	TRANSITIONAL_DELAY	2019-08-01_3.M10401	A4612194.0.M10401
31-07-2019 22:20:53	31-07-2019 22:20:51	3	HOLE_SETUP	2019-08-01_3.M10401	A4612194.0.M10401
31-07-2019 22:20:54	31-07-2019 22:20:54	1	TRANSITIONAL_DELAY	2019-08-01_3.M10401	A4612194.0.M10401
31-07-2019 22:21:14	31-07-2019 22:20:55	20	HOLE_SETUP	2019-08-01_3.M10401	A4612194.0.M10401
31-07-2019 22:21:16	31-07-2019 22:21:15	2	TRANSITIONAL_DELAY	2019-08-01_3.M10401	A4612194.0.M10401
31-07-2019 22:22:03	31-07-2019 22:21:17	47	DRILLING	2019-08-01_3.M10401	A4612194.0.M10401
31-07-2019 22:22:04	31-07-2019 22:22:04	1	TRANSITIONAL_DELAY	2019-08-01_3.M10401	A4612194.0.M10401
31-07-2019 22:22:15	31-07-2019 22:22:05	11	ANCHORING	2019-08-01_3.M10401	A4612194.0.M10401
31-07-2019 22:22:17	31-07-2019 22:22:16	2	TRANSITIONAL_DELAY	2019-08-01_3.M10401	A4612194.0.M10401

Rys. 1. Fragment zbioru analizowanych danych. (Source: author's own elaboration)

Fig. 1. A segment of a set of analyzed data. (Source: own elaboration)

	A	B
1	Activity	Code
2	HOLE_SETUP	H
3	TRANSITIONAL_DELAY	T
4	DRILLING	D
5	ANCHORING	A

Rys. 2. Przyporządkowanie kodów do czynności. (Źródło: opracowanie własne)

Fig. 2. Assigning codes to activities. (Source: own elaboration)

As it can be observed, the data contain the duration of the activities that make up the bolting process and have been assigned change and production cycle identification number. Many cycles are carried out during one shift. The activities carried out within a cycle are "Hole Setup", "Transitional delay", "Drilling" and "Anchoring". Cycles are separated by activities that are not part of them (e.g. "Travelling"), which are carried out as part of a production shift.

3.1. Searching for similarities

In the first place, the repeatability of activities in the production cycle was analyzed. For this purpose, a script was created in VBA that assigned one-way codes to subsequent actions (Fig. 2.)

Subsequently, these codes were combined in sequences of letters, according to the order of implementation of activities within the whole cycles. The code of the script implementing this combination is shown in Fig. 3.

```

Dim C(1 To 2, 1 To 4) As String
Private Sub CreatePatterns()
    Dim p, a As Worksheet
    Set a = ActiveWorkbook.Sheets("assignment")
    Set p = ActiveWorkbook.Sheets("patterns")
    For i = 1 To 4
        C(1, i) = a.Cells(i, 1)
        C(2, i) = a.Cells(i, 2)
    Next
    cid = ""
    code = ""
    p.Cells(1, 1) = "cycleID"
    p.Cells(1, 2) = "pattern"
    k = 2
    For i = 2 To 71206
        If Cells(i, 6) <> cid Then
            If cid <> "" Then
                p.Cells(k, 1) = cid
                p.Cells(k, 2) = code
                k = k + 1
            End If
            cid = Cells(i, 6)
            code = ""
        End If
        code = code & FindS(Cells(i, 4))
    Next
End Sub
Private Function FindS(a As String) As String
    FindS = "X"
    For i = 1 To UBound(C, 2)
        If C(1, i) = a Then
            FindS = C(2, i)
            Exit For
        End If
    Next
End Function

```

Rys. 3. Skrypt w języku VBA generujący kody czynności w ramach cykli. (Źródło: opracowanie własne)

Fig. 3. VBA script for generating activity codes for cycles. (Source: own elaboration)

	A	
1	cycleID	pattern
2	A4612194V	HTHTHTDTATHDHTHTHTDTATHDTATHHTDTATHDHTDAHTHTDTAHTHT
3	A4613254V	THHTHTDT
4	A4613537V	TATHHTDT
5	A4614246V	DATHDTATHHTHTDTATHDTATHDTATHHTHTDATHHTDATHHTDTATHHTDTATHHTDTATHHTDTATHHTHT
6	A4614843V	HTDATHHTHTDTATHHTHTDATHHTHTHTDTATHHTHTDTATHHTDTATHHTDTATHHTHTDTATHHT
7	A4615149V	HTDTATHDTATHDTATHDTATH
8	A4616691V	THHTDTATHHTDTATHHTDTATHHTHTDTATHHTDHTDATHHTDTATHHTDATHHTHTHTHTHTDATHHTDHTDT
9	A4616803V	THHTDATHHT
10	A4618621V	THHTDTATHHTDTATHHTDTATHHTHTDTATHHTDTATHHTDTATHHTHTHTHTHTDATHHTDATHHTDTATHHTDT
11	A4619592V	HTDTATHDTATHHTDTATHHTDTATHHTHTDTATHHTDTATHHTDTATHHTHTHTHTDATHHTDATHHTDTATHHTHTDATHHT

Rys. 4. Układy kodów wygenerowane dla każdego cyklu produkcyjnego. (Źródło: opracowanie własne)

Fig. 4. Code systems generated for each production cycle. (Source: own elaboration)

The result of the script is shown in Figure 4. It clearly shows that the letter arrangements are of different lengths and regularities are difficult to find. This confirms that the cycles are very different.

In order to quantify the degree of repetition of activities within the cycles, activity systems were clustered. This was accomplished by importing codes into the R language and the number of repetitive systems was counted subsequently.

Import of data from an Excel file was carried out using the read.xlsx function from the "openxlsx" library

```
pat <- read.xlsx("patterns.xlsx", sheet=1)
```

The object created in this way contains data in the same arrangement as in Fig. 4. The next step is to cluster the data with the same code system. In each group, the occurrence of entities was counted, resulting in a summary of number of repeating codes. The last activity of these

calculations is sorting the groups in order of decreasing numbers.

```
pat %>% group_by(pattern) %>% summarize(n=n())
%>% select(pattern,n) %>% arrange(desc(n)) ->
patterns;
View(patterns)
```

From the aggregation shown in Figure 5 it can be read that the most numerous groups contained 4 elements (this corresponds to approx. 0.5% cycles). This gives an idea of how different the systems of activities in cycles are from each other.

Another attempt to look for similarities consists in analysing the direct sequences of activities. As a result of appropriate data processing, it was transformed into a matrix in which the number of pairs of directly sequential activities was compiled. As before, a corresponding VBA script was written for this purpose. Such an aggregation ensures a more detailed picture of the implementation of the analyzed cycles by identifying unusual cases. Atypical cases are cases where an activity occurs after an activity that should not precede it.

Table 1 shows the matrix, in which the sequences of actions in all analysed cycles are listed. The rows of the matrix represent the preceding actions, while the columns contain the subsequent actions. It can be read from Table 1 that immediately after the "DRILLING" activity in 4860 cases the "TRANSITIONAL_DELAY" activity followed, in 575 cases it was the "HOLE_SETUP" activity, in 1257 cases "ANCHORING", while in 13 cases this activity ended the cycle.

As already mentioned, Table 1 summarises data from all analyzed cycles. For comparison, Tables 2 and 3 show aggregation from single cycles. It can be noted that they differ in their numerical values, but equally importantly, they differ in the places where the numbers occur. Unlike the cycle A46121946V, the cycle A4614246V started with the activity

"HOLE_SETUP" and after the activity "ANCHORING" the activity "HOLE_SETUP" was repeated twice.

The framework of this article does not allow for presenting more differences between cycles, however, it can also be seen in Figure 6, which presents a segment of an aggregation of the activities sequence in subsequent cycles.

In the table, the segment of which is presented in Figure 6, compared to tables 1, 2, 3 those columns were removed in which the number of data was residual (e.g. "ANCHORING-DRILLING"), data on the number of the „ANCHORING” activities was added instead, as well as the total duration of all activities in a cycle ([s]), the quotients of which were taken as a measure of the performance of individual cycles. For the sake of simplicity, the names of the columns with the following actions were changed (e.g. „DRILLING-ANCHORING” for „DA”).

In the data processed in this way, similarities can be observed due to the distribution of consecutive activities as well as their number. This data set was therefore adopted as material for clustering the cycles into different sets of similar cycles.

3.2. Data clustering

Data clustering was performed using the k-means method. For this purpose, the k-means function provided by the R language was used. However, the data had to be processed before in order to eliminate outliers that could significantly affect the quality of the clustering. An example of such data is the occurrence of a cycle lasting one second in the data that occurred in the data set probably due to an error.

The following are the next steps in R:

- loading the data from an Excel file into the entity bmat 221 (data shown in Figure 6),


```
bmat221 <- read.xlsx("boltingmatrix1.xlsx", sheet=2)
```

pattern	n
1 THTHTDTATHTHT	4
2 THTHTHTDTATHT	4
3 THTHTDTATHTH	3
4 THTHTHTDTATHTHTDTATHTHTHT	3
5 THTDTATHTHTHTDTATHTHT	2
6 THTHTDTATHTHTDTATHTHT	2
7 THTHTDTATHTHTDTATHTHTHTDTATHTHTDTATHTHT	2
8 THTHTDTATHTHTDTATHTHTHTH	2
9 THTHTDTATHTHTH	2
10 THTHTDTATHTHTHT	2
11 THTHTDTATHTHTHTDTATHTHT	2
12 THTHTDTATHTHTHTDTATHTHTHT	2
13 THTHTDTATHTHTHTDTATHTHTHTH	2
14 THTHTHTDTATHTHT	2
15 THTHTHTDTATHTHTHTDTATHTHTHT	2
16 THTHTHTHTHTHTDTATHTHTHTDTATHTHTH	2
17 A	1
18 ATHTHT	1
19 DATHTDTATHTHTHTDTATHTDTATHTDTHTHTDATHTHTDAT...	1

Rys. 5. Zestawienie ilościowe powtarzalności kodów. (Źródło: opracowanie własne)
Fig. 5. Quantitative aggregation of codes repeatability. (Source: own elaboration)

Tab. 1. Aggregation of the sequences of the activities in all cycles. Source: own elaboration

Tab. 1. Agregacja sekwencji czynności we wszystkich cyklach. Źródło: opracowanie własne

All cycles	START	TRANSITIONAL_ DELAY	HOLE_SETUP	DRILLING	ANCHORING	STOP
START		832	19	2	1	
TRANSITIONAL_DELAY			19344	6649	4624	362
HOLE_SETUP		19628		42	9	477
DRILLING		4860	575		1257	13
ANCHORING		5782	105			2
STOP						

Tab. 2. Aggregation of the activities sequences in the A4612194V cycle. Source: own elaboration

Tab. 2. Agregacja sekwencji czynności w cyklu A4612194V. Źródło: opracowanie własne

A4612194V	START	TRANSITIONAL_ DELAY	HOLE_SETUP	DRILLING	ANCHORING	STOP
START			1			
TRANSITIONAL_DELAY			11	8	5	
HOLE_SETUP		16				1
DRILLING		5	2		1	
ANCHORING		4	2			
STOP						

Tab. 3. Aggregation of the activities sequences in the A4612194V cycle. Source: own elaboration

Tab. 3. Agregacja sekwencji czynności w cyklu A4612194V. Źródło: opracowanie własne

A4614246V	START	TRANSITIONAL_ DELAY	HOLE_SETUP	DRILLING	ANCHORING	STOP
START				1		
TRANSITIONAL_DELAY			20	12	6	
HOLE_SETUP		21				1
DRILLING		7	1		5	
ANCHORING		11				
STOP						

- creating the eff1 entity by adding a column with calculated cycle performance to bmat221,
`bmat221 %>% mutate(eff=3600*Number_Anch/
CycleTime) -> eff1`
- creating an eff2 entity in which the wrong cycle has been eliminated
`eff2 <- filter(eff1,CycleTime!=1)`
- elimination of outliers.

The distribution of performance values is shown in Figure 1. It presents a box chart created by the geom_boxplot function:

```
ggplot(eff2) + geom_boxplot(aes(y=eff))
```

Chart 1 shows the distribution of cycle performance values. The chart shows the first and third quartiles as the edges of the box, the second as a segment in the center of the box, and the mustache (lower and upper), beyond which there are outliers [8] calculated based on the interquartile range (IQR):

```
ul=quantile(eff2$eff, 0.75) + 1.5*IQR(eff2$eff)  

ll=quantile(eff2$eff, 0.25) - 1.5*IQR(eff2$eff)
```

- creating an eff2 entity in which outliers are omitted,
`eff2 <- filter(eff2, eff>=ll, eff<=ul)`
- determination of the number of groups based on the "elbow" method.

In this method, the number of groups is depicted on one axis and the sum of squares of the distances of individ-

ual observations from centroids on the other. We choose the number of groups after which adding another group does not bring such great benefits.

```
z <- 0; for(i in c(2:15)) {z[i-1] <- sum(k-  

means(select(eff2, -X1, -CycleTime, -Number_Anch,  

-eff), centers=i, nstart=15)$within)}
```

- creating an "elbow" chart,

```
ggplot(data.frame(x=c(0:13), y = z), aes(y=y, x=  

=x)) + geom_point()
```

Analyzing chart 2, it was assumed that after the tenth point, the measure decrease is insignificant, so a partition into 10 groups was assumed.

```
cent <- 10
```

```
f <- kmeans(select(eff2, -X1, -CycleTime, -Number_  

Anch, -eff), centers=cent)
```

As a result of clustering, we get information about which of the ten clusters has been assigned to each cycle. At the same time, the k-means function returns the coordinates of the group's centres. These are shown in Table 4.

- creating chart 3 to show the group arrangement,

```
df <- scale(select(eff2, -X1, -CycleTime, -Number_  

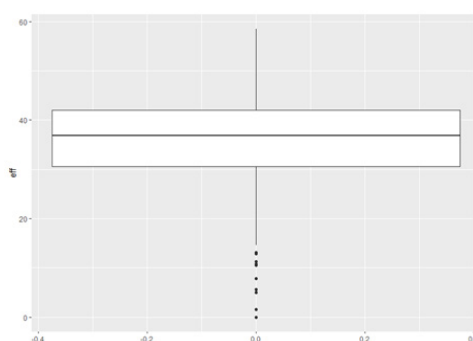
Anch, -eff))
```

```
res.km <- eclust(df, "kmeans", nstart = 25)
```

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1		AH	AT	DA	DH	DT	HA	HD	HT	TA	TD	TH	CycleTime	Number_Anch
2	A4612194V	2	4	1	2	5	0	0	16	5	8	11	585	6
3	A4613254V	0	0	0	0	1	0	0	2	0	1	2	1008	0
4	A4613537V	0	1	0	0	1	0	0	2	1	1	2	189	1
5	A4614246V	0	11	5	1	7	0	0	21	6	12	20	707	11
6	A4614843V	0	8	3	0	5	0	1	16	5	7	18	518	8
7	A4615149V	0	4	2	0	2	0	0	5	2	4	5	293	4
8	A4616691V	0	20	8	2	12	0	1	43	12	21	43	1502	20
9	A4616803V	0	1	1	0	0	0	0	3	0	1	3	105	1
10	A4618621V	0	21	7	0	17	0	0	44	14	24	44	1763	21
11	A4619592V	0	13	5	0	8	0	1	20	8	12	22	915	13
12	A4620044V	0	5	0	0	5	0	0	10	5	5	10	438	5
13	A4621894V	0	4	4	2	0	0	0	9	0	6	8	329	4
14	A4623048V	0	8	3	1	6	0	0	23	5	10	22	1084	8
15	A4635827V	0	1	0	0	1	0	0	18	1	1	18	2277	1
16	A4641661V	0	9	3	0	8	0	0	19	6	11	19	790	9
17	A4642110V	0	4	0	0	4	0	0	12	4	4	13	424	4

Rys. 6. Fragment zestawienia następstw czynności w poszczególnych cyklach.. (Źródło: opracowanie własne)

Fig. 6. A segment from the aggregation of the activities sequence in the particular cycles. Source: own elaboration



Wykres 1. Fragment zestawienia następstw czynności w poszczególnych cyklach. (Źródło: opracowanie własne)

Chart 1. Extract from the summary of the sequences of the activities in the particular cycles. Source: own elaboration

In the next stage of calculations, a column with the numbers of clusters to which the cycles were assigned was added to the data based on which the cycles were clustered. The `eff3` object was created in this way.

```
eff3 <- mutate(eff2, gr=f$cluster)
```

Subsequently, chart 4 was created, showing the distribution of performance values in individual clusters,

```
ggplot(eff3) + geom_boxplot(varwidth=TRUE, aes(y=eff, x=reorder(gr, gr, FUN=mean), group=gr), fill='lightblue') + theme(panel.background = element_rect(fill = "#ddeecc", colour = "#00dd00", size = 2, linetype = "solid"), panel.grid.major = element_line(size = 0.5, linetype = 'dotted', colour = "#777777"), panel.grid.minor = element_line(size = 0.5, linetype = 'dotted', colour = "#777777")) + scale_x_discrete(breaks=c(1:10))
```

3.3. Control of the performance achieved

The clusters designated above, together with the performance distributions observed in them, can be used for the ongoing process control. Due to the calculations carried out, it is possible to confront the performance in the current cycles with the performance observed before, but taking into account the type of production cycle. An example of this is given below.

The data concerning the activities system in the analyzed cycle are saved as a vector named `newdata`:

```
newdata <- c(0, 10, 2, 0, 7, 0, 0, 17, 8, 10, 20)
```

Then the `predict.kmeans` function is called, which returns the number of the cluster to which the new cycle will be classified.

```
predict.kmeans <- function(object, newdata){
  centers <- object$centersd
  n_centers <- nrow(centers)
  dist_mat <- as.matrix(dist(rbind(centers,
  newdata)))
  dist_mat <- dist_mat[-seq(n_centers), seq(n_
  centers)]
  max.col(t(-dist_mat))
}
```

```
predict.kmeans(f, newdata)
```

In the analyzed example, the prediction of the assignment of the new cycle described by the coordinates in the `newdata` object, resulted in the value 8, so the cycle was assigned to the cluster eight.

By filtering the data in the way that it only contains cycles belonging to this cluster, the average value and median performance in this cluster can be determined and compared with the performance observed in the new cycle.

```
eff3 %>% filter(gr==8) %>% summarize(eff-
mean=mean(eff), effmedian=median(eff))
```

Tab. 4. Aggregation of coordinates of the centres of the created groups. Source: own elaboration

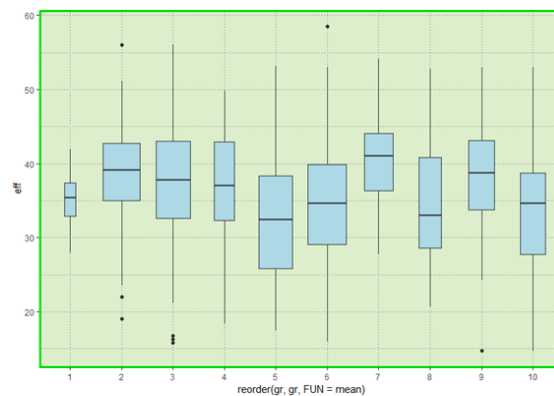
Tab. 4. Agregacja współrzędnych centrów utworzonych grup. Źródło: opracowanie własne

	AH	AT	DA	DH	DT	HA	HD	HT	TA	TD	TH
1	0.03	2.90	0.51	0.08	2.45	0.01	0.01	10.34	2.41	3.02	10.72
2	0.30	14.15	2.85	0.85	11.88	0.00	0.08	56.33	11.60	15.50	55.60
3	0.21	9.97	2.84	0.84	7.78	0.02	0.08	33.54	7.31	11.38	32.92
4	0.22	5.83	2.48	0.43	3.71	0.00	0.06	25.03	3.60	6.56	24.90
5	0.67	19.33	3.33	0.92	17.25	0.00	0.00	79.83	16.67	21.50	78.75
6	0.10	3.85	0.86	0.23	3.20	0.01	0.01	16.15	3.08	4.28	16.25
7	0.04	1.45	0.26	0.03	1.21	0.03	0.01	4.69	1.19	1.48	5.03
8	0.13	9.33	1.52	1.19	8.55	0.00	0.03	26.35	7.93	11.24	25.48
9	0.23	12.86	2.83	1.02	10.69	0.02	0.02	41.59	10.25	14.52	40.77
10	0.04	8.09	1.54	1.00	6.87	0.01	0.06	19.93	6.60	9.33	19.36



Wykres 3. Układ wydzielonych grup. (Źródło: opracowanie własne)

Chart 3. Arrangement of separate clusters. Source: own elaboration



Rys. 10. Rozkład wydajności procesu w poszczególnych grupach. (Źródło: opracowanie własne)

Fig. 10. The distribution of performance values in individual clusters. Source: own elaboration

For cluster eight, the values are: 36.9 and 37.3 bolting/h, respectively.

As a result of this comparison, it is possible to assess how the observed performance ranks among the others in this group and whether it should be assumed that it has achieved the expected result or whether actions should be taken to learn the effects of the deviation.

4. Conclusions

The performance of the production process can depend on many factors. In cases where it is not difficult to determine the impact of external factors on the production process, determining its expected performance is a simple task. In the mining industry, constant repeatability of the production process practically does not occur. There are a number

of areas of external factors (environment, equipment, people) that cause its variability. The analysis presented in the article clearly shows that the repeatability of the analyzed production cycles is basically incidental. The variability of production cycles results in the variability of their performance. The proposed concept of process performance control takes into account their variability by comparing them with the performance observed in separate groups of cycles.

At the same time, the k-means clustering method and the calculation process using VBA and R languages were presented in practice.

The paper presents results of research conducted in AGH University of Science and Technology no. 16.16.100.215.

Literatura – References

1. Bąk P.: Production planning in a mining enterprise - selected problems and solutions , Mineral Resources Management ; ISSN 0860-0953. - 2018 vol. 34 iss. 2, p. 97–116.
2. Brzychczy E.: Probabilistic modeling of mining production in an underground coal mine, Intelligent Systems in Production Engineering and Maintenance, Springer Nature Switzerland AG, cop., 2019.- S. 655-667.
3. Brzychczy E.: An overview of data mining and process mining applications in underground mining, Journal of the Polish Mineral Engineering Society, ISSN 1640-4920, 2019 R. 21 nr 1, s. 301–314.
4. Fuksa D.: The ways of solving non-linear decision problems through application of optimal production plans for mines. Mineral Resources Management; ISSN 0860-0953. 2007 t. 23 z. 3, p. 97–108.
5. Fuksa D.: Consolidation of mining enterprises as a chance to make the Polish mining industry profitable. Inżynieria Mineralna. 2018 R. 20 No. 2, p. 47–54.
6. Fuksa D.: The method for the assessment of the impact of variable demand of hard coal consumers on the profitability of a multi-facility mining enterprise. Inżynieria Mineralna. 2017 R. 18 No 1, p. 249–256.
7. Hartigan J. A., Wong M. A. : A K-Means Clustering Algorithm, Applied Statistics, 1979, Vol. 28, No. 1, p. 100–108.
8. Kaliyaperumal, Senthamarai & Kuppusamy, Manoj & Arumugam, S.: Labeling Methods for Identifying Outliers, International Journal of Statistics and Systems, 2015, 10. 231-238.
9. Kęsek M., Kłaś M., Adamczyk A.: A review of computer simulations in underground and open-pit mining, Journal of the Polish Mineral Engineering Society ; ISSN 1640-4920. 2018 R. 20 No. 2, p. 7–14.
10. 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.
11. Kęsek M.: Analysing data with the R programming language to control machine operation, Inżynieria Mineralna - Journal of the Polish Mineral Engineering Society ; ISSN 1640-4920. — 2019 R. 21 nr 1, s. 231–235.
12. Kęsek M.: Visual Basic as a Tool for Monitoring and Analyzing Machines, Inżynieria Mineralna - Journal of the Polish Mineral Engineering Society, 2017, No. 2, pages: 195–200.
13. Kopczewska K., Kopczewski T., Wójcik P.: Metody ilościowe w R. Aplikacje ekonomiczne i finansowe. Wydawnictwa Fachowe CeDeWu, Warszawa 2009.
14. Ogrodnik R.: Environmental performance indicators of hard coal mine, IOP Conference Series: Earth and Environmental Science ; ISSN 1755-1307. - 2019 vol. 214 [no.] 1 art. no. 012084, s. 1–11.
15. Walkenbach J.: Excel 2013 PL; programowanie w VBA; Vademecum Walkenbacha, Helion, Gliwice 2014.

Wykorzystanie metody grupowania k-średnich do kontroli wydajności procesu produkcyjnego

W artykule przedstawiono koncepcję wykorzystania grupowania obiektów metodą k-średnich do kontroli wydajności procesu produkcyjnego, który przebiega w zmiennych warunkach. Rozkłady wydajności procesu produkcyjnego w pogrupowanych pod względem podobieństwa cyklach produkcyjnych, stanowią podstawę kontroli wydajności kolejnych cykli produkcyjnych. Część praktyczna pracy zawiera przykład obliczeń przeprowadzonych według tej koncepcji z użyciem języka VBA oraz języka R i dotyczy procesu kotwienia w kopalniach podziemnych.

Słowa kluczowe: wydajność procesu produkcyjnego, grupowanie (clusterng), k-means, cykl produkcyjny, język R, VBA, process mining, kotwienie (bolting)