Carpathian Journal of Electronic and Computer Engineering 13/1 (2020) 5-12

DOI: 10.2478/cjece-2020-0002

R Libraries {dendextend} and {magrittr} and Clustering Package scipy.cluster of Python For Modelling Diagrams of Dendrogram Trees

Polina Lemenkova
College of Marine Geo-sciences,
Ocean University of China,
Qingdao, China
ORCID ID: 0000-0002-5759-1089
pauline.lemenkova@gmail.com

Abstract—**The paper presents a comparison of the two languages Python and R related to the classification tools and demonstrates the differences in their syntax and graphical output. It indicates the functionality of R and Python packages {dendextend} and scipy.cluster as effective tools for the dendrogram modelling by the algorithms of sorting and ranking datasets. R and Python programming languages have been tested on a sample dataset including marine geological measurements. The work aims to detect how bathymetric data change along the 25 bathymetric profiles digitized across the Mariana Trench. The methodology includes performed hierarchical cluster analysis with dendrograms and plotted clustermap with marginal dendrograms. The statistical libraries include Matplotlib, SciPy, NumPy, Pandas by Python and {dendextend}, {pvclust}, {magrittr} by R. The dendrograms were compared by the model-simulated clusters of the bathymetric ranges. The results show three distinct groups of the profiles sorted by the elevation ranges with maximal depths detected in a group of profiles 19-21. The dendrogram visualization in a cluster analysis demonstrates the effective representation of the data sorting, grouping and classifying by the machine learning algorithms. The programming codes presented in this study enable to sort a dataset in a similar research aimed to group data based on the similarity of attributes. Effective visualization by dendrograms is a useful modelling tool for the geospatial management where data ranking is required. Plotting dendrograms by R, comparing to Python, presented functional and sophisticated algorithms, refined design control and fine graphical data output. The interdisciplinary nature of this work consists in application of the coding algorithms for spatial data analysis.**

Keywords—clustering, data analysis, data sorting, data ranking, dendrogram, R, Python, machine learning, programming language

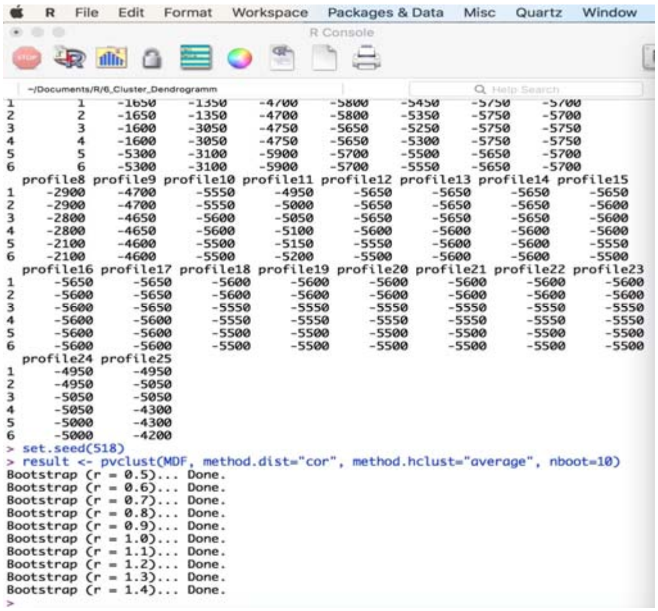
I. Introduction

To statistical analyze and sort information presented in a large geospatial datasets as unsorted raw data, it is necessary to discretize attributes and analyze variables, which have values common or similar in nature [1-2]. A typical ranking, grouping or sorting data algorithm splits a large dataset into several subgroups according to their values. An example of such a technique is an unsupervised clustering that has a graphical output as a dendrogram, a hierarchically plotted tree with sorted and ranked data. This study aims at the following question: since a raw dataset contains quantitative variables (depths) with known characteristics and a large number of values (25 cross-sections of the bathymetric profiles) should all these values be kept as such or it is more sensible to group them into clusters through data sorting and hierarchical ranking? The solution to this question was proposed by a comparative testing of Python and R. Thus, the focus of this research is laid on visualizing and plotting dendrograms by cluster analysis using machine learning algorithms of R and Python programming languages.

The structure of the paper consists of six sections, ten figures and three code listings. Section 1 (Introduction) summarizes the scope, focus and goal of the research aimed on geospatial data sorting by Python and R through the unsupervised clustering with dendrograms as a graphical output. This section also formulates the research problem through introducing questions of the hierarchical data ranking by the machine learning methods in Python and R. Section 2 (Methods) briefly describes the existing works on the statistical analysis and algorithms of the machine learning. It presents first a theoretical background of the cluster analysis (Subsection A) and then a programming background demonstrating technical solutions of the statistical methods by Python and R libraries (Subsection B). Section 3 (Programming scripts: Python and R) starts with a short description of the data capture and then presents three code listings of R (Subsection A) and Python (Subsections B and C), each supported by the visualized graphical output. Section 4 (Results) reflects the previous section by presenting the results of the statistical analysis by R and Python in subsections A and B, respectively. Section 5 (Discussion) compares the approaches of R and Python libraries for dendrogram visualization and discusses the functionality of both tools. Section 6 (Conclusion) resumes the presented research and ends with a recommendations for further studies. In a summary, it reflects the perspectives of the geostatistical analysis in marine geology using other tools besides Python and R (Matlab, Octave, GMT, AWK, C++). The link to the GitHub repository with available data and methods used in this paper is provided.

II. Methods

A Hierarchical Cluster Analysis (HCA) is one of the existing unsupervised statistical methods described in the literature on general statistical analysis and data analysis [18-22]. The HCA enables to classify a given data frame of variables into the sorted groups (or clusters, from which the name of the method is derived). According to the algorithm of the machine learning, clusters are defined by similarities between variables which enable them to be assigned into one cluster group. There are various approaches of the hierarchical data clustering algorithms. The divisive clustering recursively performs the partition of the samples from a single cluster to the least similar ones [23-24]. The agglomerative clustering computes the similarity between each of the clusters and joins the two most similar ones [25]. The single linkage approach measures the shortest distance between two points in each cluster [26-27]. The complete linkage computes the longest distance between two points in each cluster [28]. The average linkage calculates the average distance between each point in one cluster to every point in the other cluster [29-30]. This theoretical principle of the statistical data sorting was used as a background [31-35] with the programming codes modified for the data set.



```
1        1   -1650   -1350   -4700   -5800   -5450   -5750   -5700
2        2   -1650   -1350   -4700   -5800   -5350   -5750   -5700
3        3   -1600   -3050   -4750   -5650   -5250   -5750   -5750
4        4   -1600   -3050   -4750   -5650   -5300   -5750   -5750
5        5   -5300   -3100   -5900   -5700   -5500   -5650   -5700
6        6   -5300   -3100   -5900   -5700   -5550   -5650   -5700
  profile8 profile9 profile10 profile11 profile12 profile13 profile14 profile15
1    -2900    -4700     -5550     -4950     -5650     -5650     -5650     -5650
2    -2900    -4700     -5550     -5000     -5650     -5650     -5650     -5600
3    -2800    -4650     -5600     -5050     -5650     -5650     -5650     -5600
4    -2800    -4650     -5600     -5100     -5600     -5600     -5600     -5600
5    -2100    -4600     -5500     -5150     -5550     -5600     -5600     -5550
6    -2100    -4600     -5500     -5200     -5500     -5600     -5600     -5500
  profile16 profile17 profile18 profile19 profile20 profile21 profile22 profile23
1     -5650     -5650     -5600     -5600     -5600     -5600     -5600     -5600
2     -5600     -5650     -5600     -5600     -5600     -5600     -5600     -5600
3     -5600     -5650     -5550     -5550     -5550     -5550     -5550     -5550
4     -5600     -5600     -5550     -5550     -5550     -5550     -5550     -5550
5     -5600     -5600     -5500     -5500     -5500     -5500     -5500     -5500
6     -5600     -5600     -5500     -5500     -5500     -5500     -5500     -5500
  profile24 profile25
1     -4950     -4950
2     -4950     -5050
3     -5050     -5050
4     -5050     -4300
5     -5000     -4300
6     -5000     -4200
> set.seed(518)
> result <- pvclust(MDF, method.dist="cor", method.hclust="average", nboot=10)
Bootstrap (r = 0.5)... Done.
Bootstrap (r = 0.6)... Done.
Bootstrap (r = 0.7)... Done.
Bootstrap (r = 0.8)... Done.
Bootstrap (r = 0.9)... Done.
Bootstrap (r = 1.0)... Done.
Bootstrap (r = 1.1)... Done.
Bootstrap (r = 1.2)... Done.
Bootstrap (r = 1.3)... Done.
Bootstrap (r = 1.4)... Done.
>
```

Figure 1. Workflow for hierarchical clustering with p-values of data frame.

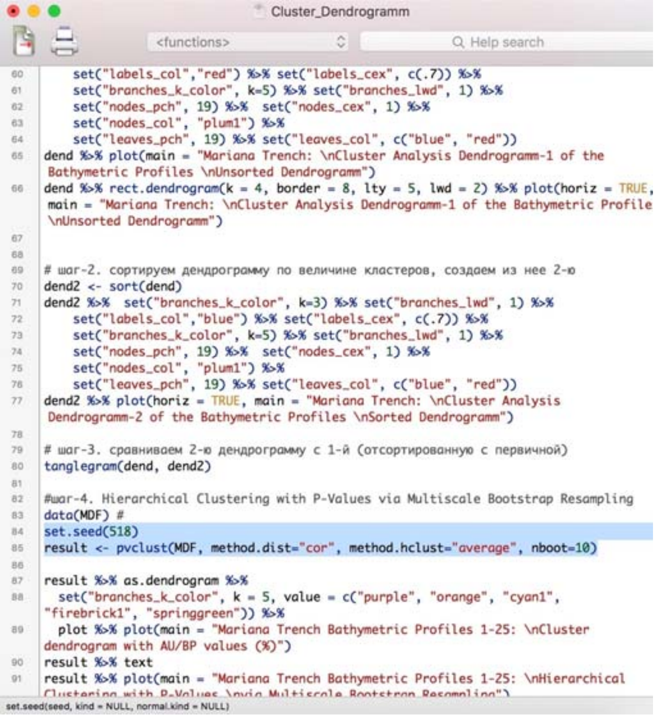
A. *Theoretical Background*

The dendrograms plotted in a cluster analysis represent the results of the statistical method aimed at the classification of the data samples. The core idea of the data sorting by dendrogram techniques is a digital pattern recognition which refers to a division of the dataset into a set of classes. The algorithms is based on the assessment of the similarity of their attribute values. The dendrogram trees as a diagram of the cluster analysis are used in a machine learning for cases where classes are not pre-defined and should be recognized by the algorithm of the statistical analysis [3-5].

Clustering data aims at the dividing of the initial dataset into a number of groups by the defined criteria for data sorting based on the similarity of their attribute values. For each cluster in clustering, quantities (called „p-values“) are calculated via the multiscale bootstrap resampling. A p-value of a cluster is a value between 0 and 1 which indicates how strong the cluster is supported by the data [6].

B. *Programming Background*

The study is technically based on two programming languages: Python and R. The data modelling and plotting are performed by the statistical libraries and packages embedded in both languages. The libraries of Python include Matplotlib, SciPy (cluster function), NumPy and Pandas (for reading data from .csv files). The main package of R used for dendrogram plotting and visualizing cluster analysis is {dendextend} developed and described by [7].



```
    set("labels_col","red") %>% set("labels_cex", c(.7)) %>%
    set("branches_k_color", k=5) %>% set("branches_lwd", 1) %>%
    set("nodes_pch", 19) %>%  set("nodes_cex", 1) %>%
    set("nodes_col", "plum1") %>%
    set("leaves_pch", 19) %>% set("leaves_col", c("blue", "red"))
dend %>% plot(main = "Mariana Trench: \nCluster Analysis Dendrogramm-1 of the
 Bathymetric Profiles \nUnsorted Dendrogramm")
dend %>% rect.dendrogram(k = 4, border = 8, lty = 5, lwd = 2) %>% plot(horiz = TRUE,
 main = "Mariana Trench: \nCluster Analysis Dendrogramm-1 of the Bathymetric Profile
 \nUnsorted Dendrogramm")


# шаг-2. сортируем дендрограмму по величине кластеров, создаем из нее 2-ю
dend2 <- sort(dend)
dend2 %>%  set("branches_k_color", k=3) %>% set("branches_lwd", 1) %>%
    set("labels_col","blue") %>% set("labels_cex", c(.7)) %>%
    set("branches_k_color", k=5) %>% set("branches_lwd", 1) %>%
    set("nodes_pch", 19) %>%  set("nodes_cex", 1) %>%
    set("nodes_col", "plum1") %>%
    set("leaves_pch", 19) %>% set("leaves_col", c("blue", "red"))
dend2 %>% plot(horiz = TRUE, main = "Mariana Trench: \nCluster Analysis
 Dendrogramm-2 of the Bathymetric Profiles \nSorted Dendrogramm")

# шаг-3. сравниваем 2-ю дендрограмму с 1-й (отсортированную с первичной)
tanglegram(dend, dend2)

#шаг-4. Hierarchical Clustering with P-Values via Multiscale Bootstrap Resampling
data(MDF) #
set.seed(518)
result <- pvclust(MDF, method.dist="cor", method.hclust="average", nboot=10)

result %>% as.dendrogram %>%
  set("branches_k_color", k = 5, value = c("purple", "orange", "cyan1",
"firebrick1", "springgreen")) %>%
  plot %>% plot(main = "Mariana Trench Bathymetric Profiles 1-25: \nCluster
dendrogram with AU/BP values (%)")
result %>% text
result %>% plot(main = "Mariana Trench Bathymetric Profiles 1-25: \nHierarchical
Clustering with P-Values \nvia Multiscale Bootstrap Resampling")
set.seed(seed, kind = NULL, normal.kind = NULL)
```

Figure 2. Clustering via multiscale bootstrap resampling.

Additionally, the activated packages of R included the {pvclust} developed by [9], a package for assessing uncertainty in the hierarchical cluster analysis, the {magrittr} providing a mechanism for chaining commands by a forward-pipe operator, %>% [10-11], and a set of auxiliary packages specially for graphical visualization, e.g. {RColorBrewer}, {grid}, with embedded techniques by common R syntax [12]. Common methodologies of using syntax of the programming languages were derived from the existing literature, general manuals and technical references of the programming and data analysis [13-17].

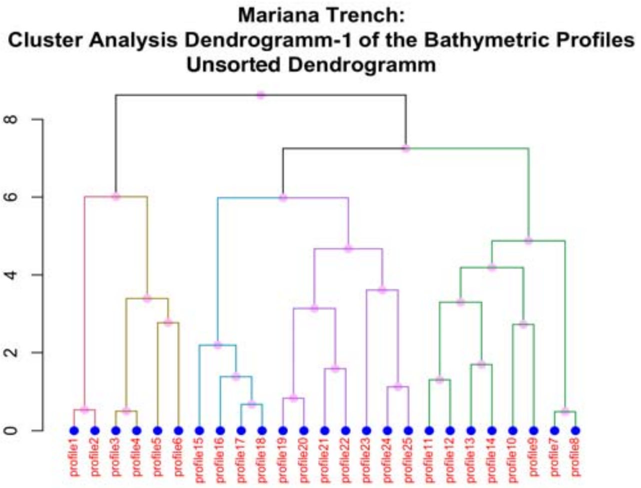


Figure 3. Unsorted dendrogram diagram tree showing 25 profiles.

III. Programming scripts: Python and R

The dataset was derived from the QGIS project where 25 cross-section bathymetric profiles were digitized across the Mariana Trench, the deepest place on the Earth with complex geological settings located in the Pacific Ocean [18-20]. The length of each profile was 1000 km, and the distance between every pair was set to 100 km using existing methodology [21-22]. The attribute information has been derived from each profile and stored in a table. The table (.csv) with three columns contained coordinates (latitude and longitude) and elevations depths (m). Afterwards the table was imported for data analysis by the statistical libraries of Python and R.

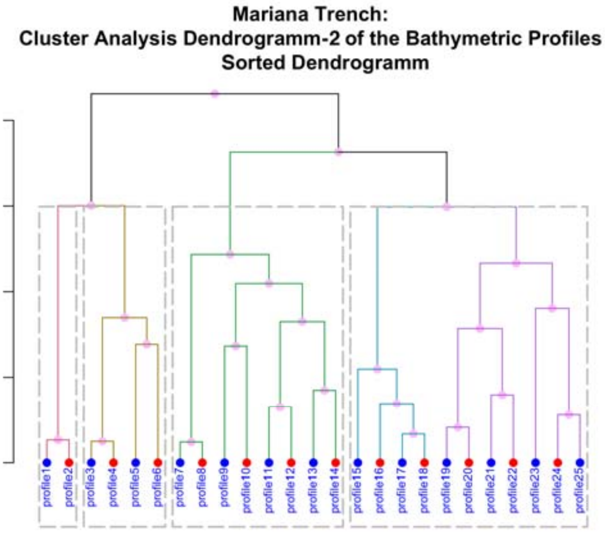


Figure 4. Sorted dendrogram diagram tree showing sorted and grouped 25 profiles, ranked by bathymetric depths into three clusters.

The {magrittr} package of R was used to decrease the development time and to improve the readability and maintainability of the R code. the main feature of {magrittr} is a 'pipe'-like operator '%>%', which 'pipes' a value of the previous expression to forward it into the next expression or a function call, as demonstrated below.

A. R code used for plotting dendrograms

```
# Step-1. Generate data frame from the raw table
MDepths <- read.csv("Depths.csv", header=TRUE, sep = ",")
# Step-2. Delete non available values (NAs)
MDF <- na.omit(MDepths)
row.has.na <- apply(MDF, 1, function(x){any(is.na(x))})
sum(row.has.na) # sum up NA, should be: [1] 0
head(MDF) # check up data frame
# Step-3. Plot the 1st dendrogram (here: by 25 clusters)
library(dendextend)
dend <- MDF[1:25,] %>%
scale %>%
dist %>% # calculate a distance matrix,
        hclust (method = "average") %>%
        as.dendrogram %>%
        set("labels", c(("profile"), rep(1:25), sep=""))
%>%
        set("labels_col","blue") %>%
        set("labels_cex", c(.7)) %>%
        set("branches_k_color", k=5) %>%
        set("branches_lwd", 1) %>%
        set("nodes_pch", 19) %>%
        set("nodes_cex", 1) %>%
        set("nodes_col", "plum1") %>%
        set("leaves_pch", 19) %>%
        set("leaves_col", c("blue", "red"))
dend %>% plot(main = "Mariana Trench: \nCluster Analysis
        Dendrogramm-1 of the Bathymetric Profiles
\nUnsorted      Dendrogramm")
# Step-4. Plot the 2nd dendrogram from the 1st by sorted
clusters
dend2 <- sort(dend)
dend2 %>%
        set("branches_k_color", k=3) %>%
        set("branches_lwd", 1) %>%
        set("labels_col","blue") %>%
        set("labels_cex", c(.7)) %>%
        set("branches_k_color", k=5) %>%
        set("branches_lwd", 1) %>%
        set("nodes_pch", 19) %>%
        set("nodes_cex", 1) %>%
        set("nodes_col", "plum1") %>%
        set("leaves_pch", 19) %>%
        set("leaves_col", c("blue", "red"))
dend2 %>% plot(main = "Mariana Trench: \nCluster Analysis
Dendrogramm-2 of the Bathymetric Profiles \nSorted
Dendrogramm")
# Step-5. Comparing two dendrograms: sorted and unsorted
tanglegram(dend, dend2)
tanglegram(dend, dend2) %>%
plot(main = "Mariana Trench: \  nComrapison of the
Cluster Dendrogramms 1 and 2")
# Step-6. Hierarchical Clustering with P-Values via
Multiscale Bootstrap Resampling
data(MDF)
set.seed(518)
result←  pvclust(MDF,  method.dist="cor",
method.hclust="average", nboot=10)
# Default plot of the result
plot(result, main = "Mariana Trench Bathymetric Profiles 1-
25: \nHierarchical Clustering with P-Values (AU/BP, %)
        \nvia Multiscale Bootstrap Resampling")
pvrect(result)
# Step-7. Plotting the results pvclust and dendextend
result %>% as.dendrogram %>%
        set("branches_k_color", k = 5, value = c("purple",
                "orange", "cyan1",
                "firebrick1", "springgreen")) %>%
        plot(main = "Mariana Trench Bathymetric Profiles
        1-25: Cluster Dendrogram\nwith AU/BP Values
        (%).  nAU: Approximately Unbiased p-Value \n
        and BP: Bootstrap Probability")
result %>% text
result %>% pvrect
# The final plots can be saved as pdf from R via "Save As".
# Step-8. Additional functionality of R enables to get
information on clusters vis this code snippet:
dend %>% get_nodes_attr("members", id = c(2,5))
dend <- MDF[1:25,] %>%  scale %>% dist %>%
        hclust %>% as.dendrogram
        set("branches_k_color", k=3) %>%
        set("branches_lwd", 1.2) %>%
        set("labels_colors") %>%
        set("labels_cex", c(.9,1.2)) %>%
        set("leaves_pch", 19) %>%
        set("leaves_col", c("blue", "red"))
plot(dend)
# Step-9. Finally, an alternative variant to plot
dendrograms is possible via the hclust function:
model <- hclust(dist(MDF), "ave")
dhc <- as.dendrogram(model)
ddata <- dendro_data(dhc, type = "rectangle")
pclusters <- ggplot(segment(ddata1)) +
        geom_segment(aes(x = x, y = y, xend = xend,
                yend = yend)) + coord_flip() +
                scale_y_reverse(expand = c(0.2, 0))
pclusters
```

The code presented above shows the stepwise process of the dendrogram plotting by R. Specifically, it includes the following steps divided into two parts. The first part of the code generates a data frame. It includes step 1 where the table was read in and a data frame was generated. The step 2 includes cleaning up the data frame from the non available (NA) data values. Now the main part 2 started after the data frame was generated and prepared. It includes a hierarchical cluster analysis and a dendrogram plotting. The step 3 includes plotting the 1st dendrogram (here: by 25 clusters). The step-4 is plotting the 2nd dendrogram from the initial 1st one, sorted by the cluster sizes. Afterwards, during the step 5th two dendrograms were compared (unsorted and sorted). At the step 6th the hierarchical clustering with P-values via the Multiscale Bootstrap Resampling was performed. The final step 7th visualizes results as sorted dendrogram by the pvclust and dendextend.

B. Python code used to plot dendrograms

```
# Step-1. Loading necessary Python libraries
import pandas as pd
from matplotlib import pyplot as plt
from scipy.cluster import hierarchy
from scipy.cluster.hierarchy import dendrogram, linkage
import numpy as np
import os
# Step-2. Generating data set from the raw table
os.chdir('/Users/pauline/Documents/Python')
df = pd.read_csv("Tab-Morph.csv")
df = df.set_index('profile')
del df.index.name
df
# Step-3. Calculating distances between each sample
Z = hierarchy.linkage(df, 'average')
# Make the dendrogram
dendrogram(Z,      labels=df.index,      leaf_rotation=0,
orientation="left",
   distance_sort='ascending')
plt.title('Hierarchical   cluster   dendrogram   for   the
geomorphic      similarity     \nof     the     25bathymetric
profiles, Mariana Trench',        fontsize=10,
fontfamily='sans-serif')
plt.show()
```

The code presented above shows a Python approach for the plotting dendrograms in several steps. Step 1 included loading necessary Python libraries: Pandas, Matplotlib, Scipy, Numpy and OS. Step 2 includes a setup of the working directory through the OS and then generating dataset from the raw table (csv). Step 3 includes the calculation of distances between each sample and plotting the dendrogram. It also includes technical characteristics of the visualization: title, orientation of the annotations, fonts and colors. The algorithm uses a Python syntax.

C. Python code used to plot dendrograms as marginal charts on a clustermap

```
# Step-1. Loading Python libraries
from string import ascii_letters
import numpy as np
import pandas as pd
import seaborn as sb
from matplotlib import pyplot as plt
import os
# Step-2. Importing data and generating data frame
os.chdir('/Users/pauline/Documents/Python')
sb.set(style="white")
df = pd.read_csv("Tab-Bathy.csv")
sb.set(color_codes=True)
# Step-3. Defining and plotting a clustermap
#g = sb.clustermap(df, metric="correlation")
g = sb.clustermap(df, cmap="mako", robust=True)
#g = sb.clustermap(df, z_score=0)
#g = sb.clustermap(df, standard_scale=1)
rotation = 45
for i, ax in enumerate(g.fig.axes):   ## getting all axes
of the fig object ax.set_xticklabels(ax.get_xticklabels(),
rotation = rotation)
# Step-4. Adding titles and defining the plot size
g.fig.suptitle('Mariana    Trench:    Clustermap    of    the
bathymetric   observations.   \nDataset:   25   cross-section
profiles,  518  observations  in  each.  \nMethod:  ignored
outliers in colormap limits')
plt.subplots_adjust(bottom=0.20,top=0.90,      right=0.90,
left=0.10)
plt.show()
```

The code presented above shows a Python approach for the plotting clustermap (also known as 'heatmap') using the following workflow. First, necessary libraries were loaded: OS, Numpy, Pandas, Matplotlib, Seaborn. The next step includes a visual setup of the graphics: selection of the Seaborn template variants for the plotting (here: a white background of the graphics, the color codes by Python and a 'paper' context). The next step includes a setup of the working directory and reading in a table. The next step includes defining variables from a table and instructing Python which ones should be plotted on the X and Y axes, and adjusting visual characteristics of the title and the subtitle (rotation of annotations, fonts). The final step includes saving the graphics and converting a file into a Portable Network Graphics (PNG) format.

IV. Results

A. Results of the dendrogram clustering by R programming

A stepwise workflow is demonstrated in the previous sections, which uses R codes for the cluster analysis, data partition, sorting and grouping, and dendrograms plotting. Dendrograms represent hierarchical schemes of grouping objects which is one of the most common forms of the graphical display in a cluster analysis [23-24]. The dendrograms, or the hierarchical tree diagrams, are one-dimensional graphs used to depict the mutual relationships between variables in a dataset, as commonly used in the geological science for finding correlations [33-34]. In the presented case, the correlation between the 25 profiles and their bathymetric values (absolute depths) were assessed.

The procedure for plotting dendrograms was performed using the code provided above and illustrated on the screenshots of the workflow, Fig. 1 and Fig. 2. The conceptual idea of the dendrograms of cluster analysis procedure consists in the application of the similarity assessment between the pairs of the object groups: profiles and depths. Hence, the profiles were sorted by their absolute depths, which is represented by an unsorted dendrogram (Fig. 3). Afterwards the profile variables were grouped into the clusters, which is illustrated by the sorted dendrogram (Fig. 4). The dendrogram of the bathymetric settings of the deep-sea trench (Fig. 3) reflects two pairs of variables characterized by a close correlation.

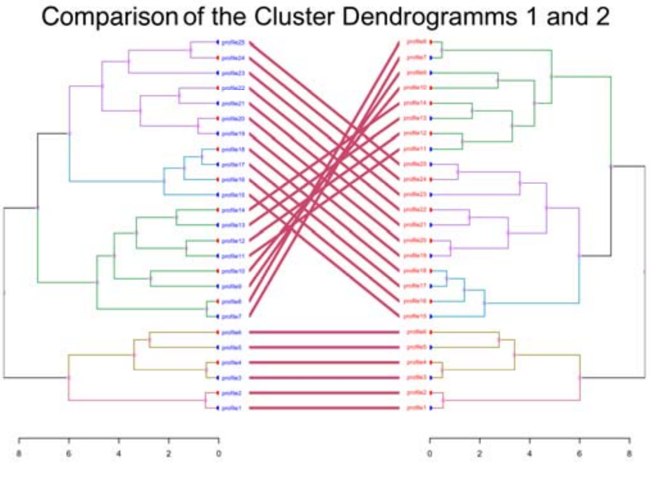


Figure 5. Comparison between cluster groups for dendrograms

The number of the bathymetric profiles and depths of the sampling points measured along each of these profiles are characterized by the Y axis, where r>8 shows a close correlation between the two groups. The second group includes the profiles 7–14 (green colored) with the profiles 15–18 (blue colored) and a group of 19–25 (purple colored), as shown on Fig. 3. The first group includes the connecting groups of the profiles Nr. 1 with 3, 5 and 6 from one side (beige and rose colored). The results of the dendrogram comparability were received using ‘tanglegram’ function when two dendrograms were visually compared using two approaches of the sorting algorithms (Fig. 5 and Fig. 6). The multiscale bootstrap resampling was performed by two types of the p-values using {pvclust} package: an AU, that is Approximately Unbiased p-value, and a BP, that is a Bootstrap Probability value.

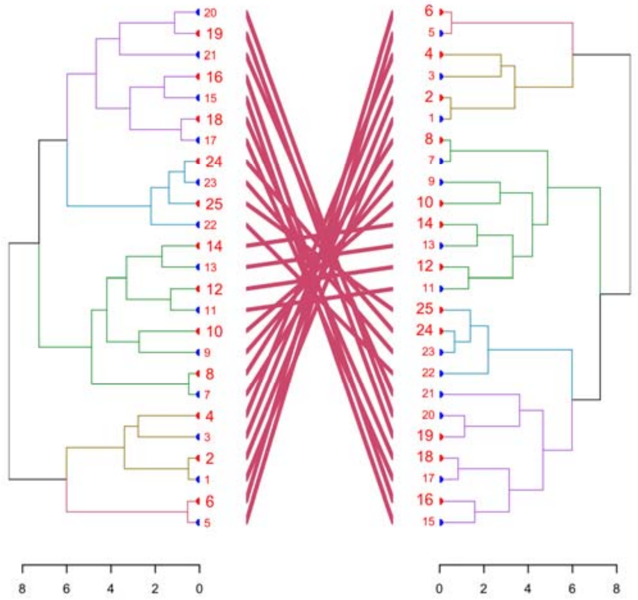


Figure 6. Comparison of two tree tangelgrams which contain the same groups of the bathymetric values (depth).

The two dendrograms were visually placed in front of each other (Fig. 5 and Fig. 6), and their labels were connected with the colored lines to make then more distinguishable.

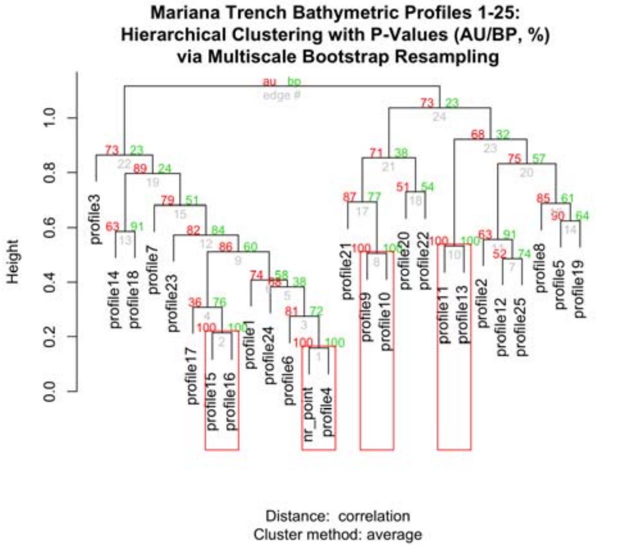


Figure 7. Hierarchical clustering with p-values by multiscale bootstrap resampling. Visualizing the grouped profiles by the correlation distance.

The profile branches as main classes indicating differences in the depth ranges are marked with a dashed line showing three main resulting groups in a dendrogram (Fig. 4). Finally, revealing the findings in an uncertainty of the hierarchical cluster analysis was done via the library {pvclust} of R. The AU p-value is computed by the multiscale bootstrap resampling (Fig. 7) which is a better approximation to the unbiased p-value comparing to the BP value computed by the normal bootstrap resampling. For each cluster in a hierarchical clustering, quantities called p-values were calculated via the multiscale bootstrap resampling (Fig. 7). The 26 attributes of the bathymetric profiles were assessed by the depths ranges which resulted in a hierarchical clustering (Fig. 7). The values on the edges of the clustering represent p-values (%). The red-colored values show the AU p-values while the green-colored ones mean the BP values. Four red-colored rectangles in a bootstrap probability clustering (Fig. 8) represent clusters with the AU larger than 95%, which means that the results show rather strong correlation by the data.

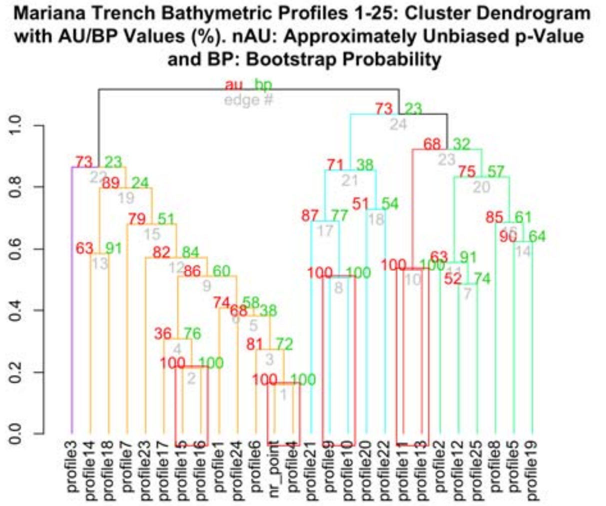


Figure 8. Cluster dendrogram with AU/BP values: approximately unbiased p-Values (in %) and bootstrap probability: the profiles are grouped into 5 distinct classes (colored ‘orange’, ‘red’, ‘cyan’, ‘green’ and ‘purple’)

The p-value of a cluster is a value between 0 and 1 (Y axis indicating height), which shows how strong does the cluster correlate with the data. In other words, it shows the similarity between the groups (X axis, indicating a distance for the correlation using average cluster method).

B. Results of the dendrogram clustering by Python

The Python based approach for plotting dendrograms of the hierarchical clustering was performed using the ‘scipy.cluster’ module of the Python SciPy library by methods described in the section 2.2. The results presented on Fig. 9 show groups of similar sample points based on their bathymetric values of the absolute depths throughout the observed 25 profiles. As can be seen on Fig. 9, four clusters of samples that have similar observed depths ranges were identified. Thus, the dendrogram graphically illustrates the final results of the hierarchical clustering.

The X axis indicates the depths ranged by the observed values of the mean elevations. The Y axis shows the profiles grouped by the similarity of their values. The samples are divided into four distinct classes according to their similarity with the main class (‘blue’) being further sorted into three subclasses (colored ‘green’, ‘red’ and ‘cyan’), Fig. 9. These clusters show that there is a significant change in elevations caused by the variations of the bathymetric patterns across the Mariana Trench. Particularly, it can be seen between the

Funding agency: CSC SOA Grant Nr. 2016SOA002.

samples No. 5 and 25 and a set of profiles 14–17 and 19–22. These areas are located in the south-western part of the trench where the deepest place of the Earth is located: the Challenger Deep.

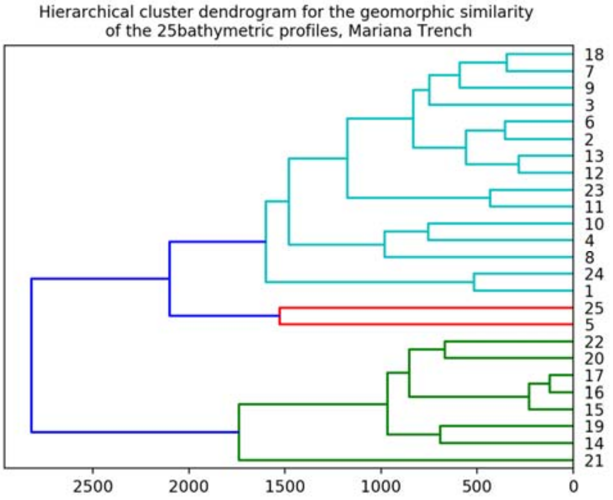


Figure 9. Dendrogram approach by Python: the lines of groups of a dendrogram are colored according to the connections based on isolation distance of the 25 bathymetric profiles in cluster analysis

Another way of representing the dendrograms by Python is visualizing a clustermap, a two dimensional matrix with the dendrograms located sidewise (Fig. 10). The dendrograms are obtained through the data sorting, dividing and grouping by Python. The main panel of the matrix is a graphical representation of the depth where individual matrix values are shown by colors (Fig. 10).

The colors are ranging by depth elevations colored almost black in the extreme depths over 7,500 m and light-colored in shallow areas. From Fig. 10 one can see that the maximal depths (black cells) are recorded by the profiles 21, 20 and 19. The majority of values are located between -3,500 to -4,500 m (colored by turquoise blue). The heatmap with marginal dendrograms visualizes patterns in the bathymetric values across the 25 profiles.

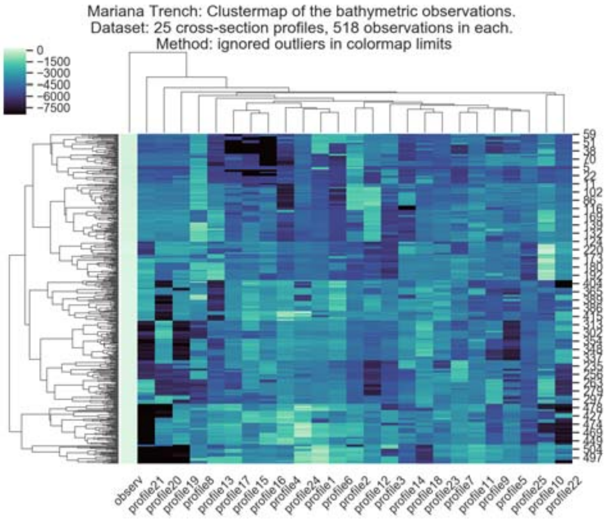


Figure 10. A cluster map visualizes the depth values (shown as colors in each cell) by a range of 25 profiles (X axe) at each of the observation points (Y axe, in total 518 points for each profile). Marginal dendrograms show groups of profiles sorted by depth values.

V. Discussion

As demonstrated in this research, the dendrograms visualized by R and Python libraries enable to perform the comparative analysis of the bathymetric data. The specific aim of this paper was to group and sort data by their absolute depths in order to detect similarities in the bathymetric patterns of the Mariana Trench. Various approaches of clustering analysis were tested by existing algorithms.

The comparison of the approaches provided by R and Python libraries for dendrogram visualization was presented in this research. According to the received results, the R based {dendextend} package is more feasible in plotting dendrogram diagrams comparing to Python. The R package {dendextend} demonstrated additional functions (e.g. bootstrap probability) aimed at the controlling of the graphical visualization of the dendrograms.

VI. Conclusions

The paper presented a comparison of the two graphical representation in Python and R. Currently, R demonstrated better graphical representation tools and technical functionality to visualize sorted data. However, in view to the constant source code development, maintenance and expanding of Python libraries, new packages might be introduced in the future to improve their graphical functionalities. The presented research demonstrated geological modelling of GIS retrieved data by programming tools usually used in the IT domain.

This study demonstrated a multidisciplinary data analysis with the algorithms of R and Python tested to perform data analysis, sorting and ranking for modelling variations in the bathymetry of the Mariana Trench. The work demonstrated effective visualization, plotting and statistical data processing by the programming algorithms in both languages through comparing their syntax and graphical output. An example of the geospatial dataset consists in the 25 bathymetric cross-section profiles of the Mariana Trench as a raw data array.

Several libraries of Python and R were tested and programming code snippets are provided for repeatability of these approaches in a similar research of the bathymetric quantitative data analysis. The work shown the effective methods for the geospatial data visualization and statistical data sorting, grouping and ranking. Using two different languages specifically for the dendrogram visualization provided positive results in a data analysis. However, R demonstrated more sophisticated functionality and refined graphical data plotting comparing to Python. However, Python demonstrated more flexibility, compactness and simplicity of the code syntax, and effective visualization.

Statistical modelling of the marine geological datasets can be done by a variety of multiple approaches, methods and algorithms, described in the available relevant literature on statistical methods in science [35-37]. To mention some specific examples, representing effective cases of data analysis include the following ones: regression models [38-39], k-means clustering, grouping and sorting [40], Python or R based advanced statistical scripting approaches [41-47].

Besides programming tools, analysis of the geological datasets can be performed by shell scripts as plugins embedded in GIS, or as scripting method of the geodata analysis and visualization using Generic Mapping Tools

(GMT). Despite being a specifically cartographic toolset primarily designed for mapping, GMT also includes available modules for the statistical data analysis: histograms, plotting median and mean in a datasets [48-55].

Other examples of the geospatial mapping for analysis of the datasets is illustrated by automated vectorization of the isolines using AutoTrace [56], geostatistical approaches by ArcGIS [57-63]. Effective way of the data analysis is a combination of various methods, for instance, C++, Matlab, cartographic mapping by ArcGIS [64-65], CARIS HIPS and GMT [66], or AWK and Octave with GMT [67]. However, few works use programming full with codes available for geological modelling. In view of this, the novelty of the presented work consists in its methodological demonstration of the programming tools used for geological datasets.

The source codes and datasets are available at the GitHub: https://github.com/paulinelemenkova/Dendrograms-by-Python-and-R

Acknowledgment

This research was supported by China Scholarship Council (CSC), State Oceanic Administration, Marine Scholarship of China, Grant Nr. 2016SOA002, China.

References

[1] A. D. Ciaccio, M. Coli, and A. J. M. Ibanez, “Studies in Theoretical and Applied Statistics. Selected Papers of the Statistical Societies”, chap. Advanced Statistical Methods for the Analysis of Large Data Sets, p. 464. Springer, 2012. doi: 10.1007/978-3-642-21037-2

[2] J. Grus, Data Science from Scratch. First Principles with Python. O’Reilly, 2015.

[3] G. Cowan, Statistical Data Analysis. Oxford Science Publications. Clarendon Press, Oxford, UK, 1998.

[4] L. J. Savage, The Foundations of Statistics, Dover, New York, 1972.

[5] E. B. Fowlkes, and C. L. Mallows, “A Method for Comparing Two Hierarchical Clusterings”, J. Am. Stat. Assoc., vol. 78, pp. 553-569, 1983.

[6] T. Galili, dendextend: an R package for visualizing, adjusting, and comparing trees of hierarchical clustering. Bioinformatics Advance Access, 2015. [Online] https://academic.oup.com/bioinformatics

[7] T. Galili, dendextend: Extending 'dendrogram' Functionality in R. [Online] https://www.rdocumentation.org/packages/dendextend/

[8] T. Galili, dendextend: an R package for visualizing, adjusting, and comparing trees of hierarchical clustering. Bioinformatics, 2015. doi: 10.1093/bioinformatics/btv428

[9] R. Suzuki and H. Shimodaira. pvclust An R package for hierarchical clustering with p-values. [Online] http://stat.sys.i.kyoto-u.ac.jp/prog/pvclust/

[10] S. B. Milton. and H. Wickham (2014) magrittr: magrittr – a forward-pipe operator for R. [Online] https://www.rdocumentation.org/packages/magrittr/versions/1.5

[11] S. Milton. Simpler R coding with pipes > the present and future of the magrittr package. [Online]. https://www.r-statistics.com/2014/08/simpler-r-coding-with-pipes-the-present-and-future-of-the-magrittr-package/

[12] R Development Core Team (2014) R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing. Vienna, Austria. [Online] http://www.R-project.org

[13] D. Cielen, A. D. B. Meysman, M. Ali. Introducing Data Science. Big Data, Machine Learning and More, Using Python Tools. Manning, Shelter Island, U.S., 2016.

[14] G. van Rossum. Python Programming Language, 2011. [Online] https://www.python.org/

[15] Downey, A.B. Think Python. How to think like a computer scientist. 2nd Ed., updated for Python 3. O’Reilly.

[16] Beazley D. M. Python essential reference. Addison-Wesley Professional. [Online] http:// www.python.org

[17] T. Gaddis, Starting Out with Python. 4th Ed. Pearson. New York, U.S.A. 2019.

[18] P. Lemenkova, “R scripting libraries for comparative analysis of the correlation methods to identify factors affecting Mariana Trench formation”, Journal of Marine Technology and Environment, vol. 2, pp. 35–42, 2018.

[19] P. Lemenkova, “Factor Analysis by R Programming to Assess Variability Among Environmental Determinants of the Mariana Trench”, Turkish Journal of Maritime and Marine Sciences, vol. 4(2), pp. 146–155, 2018.

[20] P. Lemenkova, “An Empirical Study of R Applications for Data Analysis in Marine Geology”, Marine Science and Technology Bulletin, vol. 8(1), pp. 1–9, 2019.

[21] P. Lemenkova, “Statistical Analysis of the Mariana Trench Geomorphology Using R Programming Language”, Geodesy and Cartography, vol. 45(2), pp. 57–84, 2019.

[22] P. Lemenkova, “Processing oceanographic data by Python libraries NumPy, SciPy and Pandas”, Aquatic Research, vol. 2(2), pp. 73–91, 2019.

[23] Y. Chen, L. Billard, “A study of divisive clustering with Hausdorff distances for interval data”, Pattern Recognition, vol. 96, pp. 106969. 2019.

[24] G. V. Subba Reddy, V. Ganesh, C. Srinivasa Rao, “Implementation of Genetic Algorithm Based Additive and Divisive Clustering Techniques for Unit Commitment”, Energy Procedia, vol. 117, pp. 493-500, 2017.

[25] Z. Cai, X. Yang, T. Huang, W. Zhu, “A new similarity combining reconstruction coefficient with pairwise distance for agglomerative clustering”, Information Sciences, vol. 508, pp. 173-182, 2020.

[26] F. Ros, S. Guillaume, “A hierarchical clustering algorithm and an improvement of the single linkage criterion to deal with noise”, Expert Systems with Applications, vol. 128, pp. 96-108, 2019.

[27] X. Bi, X. Luo, Q. Sun, “Branch tire packet classification algorithm based on single-linkage clustering”, Mathematics and Computers in Simulation, vol. 155, pp. 78-91, 2019.

[28] D. Krznaric, C. Levcopoulos. “Optimal algorithms for complete linkage clustering in dimensions”, Theoretical Computer Science, vol. 286(1), pp. 139-149, 2002.

[29] H. Seifoddini, “Machine grouping — Expert systems: Comparison between single linkage and average linkage clustering techniques in forming machine cells”, Computers & Industrial Engineering, vol. 15(1–4), pp. 210-216, 1988.

[30] H. K. Seifoddini, “Single linkage versus average linkage clustering in machine cells formation applications”, Computers & Industrial Engineering, vol. 16(3), pp. 419-426, 1989.

[31] R. I. Kogan, Y. P. Belov and D. A., Rodionov, Statistical ranking criteria in geology, Moscow: Nedra, in Russian, p. 321, 1983.

[32] R. I. Kogan, Interval estimation of the geological research, Moscow: Nedra, in Russian, 1986.

[33] Handbook of mathematical methods in geology, Moscow, Nedra, 1987.

[34] A. B., Kazhdan, O. I. Gus’kov, Mathematical methods in geology, Moscow: Nedra, 1990.

[35] J. Davis, Statistics and Data Analysis in Geology, Kansas Geological Survey John Wiley and Sons, 1990.

[36] D. G. Rossetier, Tutorial: An example of statistical data analysis using the R environment for statistical computing. 2017.

[37] R. Johansson, 2014. Introduction to Scientific Computing in Python. [Online], https://github.com/jrjohansson/scientific-python-lectures

[38] P. Lemenkova, “Regression Models by Gretl and R Statistical Packages for Data Analysis in Marine Geology”, International Journal of Environmental Trends, vol. 3(1), pp. 39–59, 2019.

[39] P. Lemenkova, “Testing Linear Regressions by StatsModel Library of Python for Oceanological Data Interpretation”, Aquatic Sciences and Engineering, vol. 34, pp. 51–60, 2019.

[40] P. Lemenkova, “K-means Clustering in R Libraries {cluster} and {factoextra} for Grouping Oceanographic Data”, International Journal of Informatics and Applied Mathematics, vol. 2(1), pp. 1–26, 2019.

[41] J. VanderPlas, Python Data Science Handbook. Essential Tools for Working with Data, O’Reilly, 2016.

[42] W. McKinney and PyData Development Team, Pandas: powerful Python data analysis toolkit Release 0.24.0. 2019. [Online] http://www.python.org

[43] P. Lemenkova, “Processing oceanographic data by Python libraries NumPy, SciPy and Pandas”, Aquatic Research, vol. 2, pp. 73–91, 2019.

[44] P. Lemenkova, (2019). “Calculating slope gradient variations in the submarine landforms by R and Python statistical libraries”. MANAS Journal of Engineering, 7(2), pp. 99–113.

[45] Duchesnay, E. Löfstedt, T. Statistics and Machine Learning in Python Release 0.2. [Online] http://www.python.org R Development Core Team (2014) R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing. Vienna, Austria. [Online] http://www.R-project.org

[46] P. Lemenkova, “Geospatial Analysis by Python and R: Geomorphology of the Philippine Trench, Pacific Ocean”, Electronic Letters on Science and Engineering, vol. 15(3), pp. 81–94, 2019.

[47] P. Lemenkova, “Plotting Ternary Diagrams by R Library ggtern for Geological Modelling”, Eastern Anatolian Journal of Science, vol. 5(2), pp. 16–25, 2019.

[48] P. Lemenkova, “GMT Based Comparative Analysis and Geomorphological Mapping of the Kermadec and Tonga Trenches, Southwest Pacific Ocean”, Geographia Technica, vol. 14(2), pp. 39–48, 2019.

[49] P. Lemenkova, “Topographic surface modelling using raster grid datasets by GMT: example of the Kuril-Kamchatka Trench, Pacific Ocean”, Reports on Geodesy and Geoinformatics, vol. 108, pp. 9–22, 2019.

[50] P. Lemenkova, “Automatic Data Processing for Visualising Yap and Palau Trenches by Generic Mapping Tools”, Cartographic Letters, vol. 27(2), pp. 72–89, 2019.

[51] P. Lemenkova, “Geomorphological modelling and mapping of the Peru-Chile Trench by GMT”, Polish Cartographical Review, vol. 51(4), pp. 181–194, 2019.

[52] P. Lemenkova, “Geophysical Modelling of the Middle America Trench using GMT. Annals of Valahia University of Targoviste. Geographical Series”, vol. 19(2), pp. 73–94, 2019.

[53] P. Lemenkova, “GMT Based Comparative Geomorphological Analysis of the Vityaz and Vanuatu Trenches, Fiji Basin”, Geodetski List, vol. 74(1), pp. 19–39, 2020.

[54] P. Lemenkova, “Visualization of the geophysical settings in the Philippine Sea margins by means of GMT and ISC data”, Central European Journal of Geography and Sustainable Development, vol. 2(1), pp. 5–15, 2020.

[55] P. Lemenkova, “GMT-based geological mapping and assessment of the bathymetric variations of the Kuril-Kamchatka Trench, Pacific Ocean”, Natural and Engineering Sciences, vol. 5(1), pp. 1–17, 2020.

[56] H. W. Schenke and P. Lemenkova, “Zur Frage der Meeresboden-Kartographie: Die Nutzung von AutoTrace Digitizer für die Vektorisierung der Bathymetrischen Daten in der Petschora-See”, Hydrographische Nachrichten, vol. 25(81), pp. 16–21, 2008.

[57] I. A. Suetova, L. A. Ushakova and P. Lemenkova, “Geoinformation mapping of the Barents and Pechora Seas”, Geography and Natural Resources, vol. 4, pp. 138–142, 2005.

[58] F. Yulianto, Suwarsono, T. Maulana and M. R. Khomarudin, “Analysis of the dynamics of coastal landform change based on the integration of remote sensing and GIS techniques: Implications for tidal flooding impact in Pekalongan, Central Java, Indonesia”, Quaestiones Geographicae, vol. 38(3), pp. 17–29, 2019.

[59] I. Suetova, L. A. Ushakova and P. Lemenkova, “Geoecological Mapping of the Barents Sea using GIS”. In: Proceedings of the International Cartographic Conference, July 2005, La Coruña, Spain.

[60] M. Klaučo, B. Gregorová, U. Stankov, V. Marković, V. and P. Lemenkova, “Determination of ecological significance based on geostatistical assessment: a case study from the Slovak Natura 2000 protected area”, Central European Journal of Geosciences, vol. 5(1), pp. 28-42, 2013.

[61] M. Klaučo, B. Gregorová, U. Stankov, V. Marković and P. Lemenkova, “Landscape metrics as indicator for ecological significance: assessment of Sitno Natura 2000 sites, Slovakia”, Ecology and Environmental Protection, Proceedings of the International Conference, March 2014, Minsk: BSU Press, pp. 85–90.

[62] M. Klaučo, B. Gregorová, U. Stankov, V. Marković, P. and Lemenkova, “Land planning as a support for sustainable development based on tourism: A case study of Slovak Rural Region”, Environmental Engineering and Management Journal, vol. 2(16), pp. 449–458, 2017.

[63] P. Lemenkova, C. Promper and T. Glade, “Economic Assessment of Landslide Risk for the Waidhofen a.d. Ybbs Region, Alpine Foreland, Lower Austria”. Protecting Society through Improved Understanding. 11th International Symposium on Landslides & the 2nd North American Symposium on Landslides & Engineered Slopes (NASL), June 2–8, 2012. Banff, AB, Canada, pp. 279–285, 2012.

[64] J. J. Roberts, B. D. Best, D. C. Dunn, E. A. Treml and P. N. Halpin, “Marine geo-spatial ecology tools: an integrated framework for ecological geoprocessing with ArcGIS, Python, R, MATLAB, and C++”, Environmental Modelling and Software, vol. 25, pp. 1197-1207, 2010.

[65] M. Klaučo, B. Gregorová, U. Stankov, V. Marković and P. Lemenkova, “Interpretation of Landscape Values, Typology and Quality Using Methods of Spatial Metrics for Ecological Planning”, 54th International Conference Environmental & Climate Technologies. October 14, 2013. Riga, Latvia.

[66] S. Gauger, G. Kuhn, K. Gohl, T. Feigl, P. Lemenkova and C.-D. Hillenbrand, “Swath-bathymetric mapping”, Reports on Polar and Marine Research, vol. 557, pp. 38–45, 2007.

[67] P. Lemenkova, “AWK and GNU Octave Programming Languages Integrated with Generic Mapping Tools for Geomorphological Analysis”, GeoScience Engineering, vol. 65(4), pp. 1–22, 2019.