



OpenBudgets.eu: Fighting Corruption with Fiscal Transparency

Project Number: 645833

Start Date of Project: 01.05.2015

Duration: 30 months

Deliverable D2.4

Data Mining and Statistical Analytics Techniques

Dissemination Level	Public
Due Date of Deliverable	Month 31.12.2016,
Actual Submission Date	01.02.2017
Work Package	WP 2, Data Collection and Mining
Task	T2.4 Data Mining and Statistical Analytics Techniques
Type	Demo
Approval Status	Draft
Version	0.1
Number of Pages	129
Filename	Template - Deliverable H2020 OpenBudgets

Abstract: [Abstract: 2-3 sentences max.]

In this deliverable we describe computational techniques and systems for statistical analytics and data mining for the OpenBudgets.eu (OBEU) project. Based on Deliverable 2.3 and Deliverable 5.3, we developed a highly modularized, de-centralized, easy extendable modules for financial data analysis and processing within the OBEU project.

The information in this document reflects only the author's views and the European Community is not liable for any use that may be made of the information contained therein. The information in this document is provided "as is" without guarantee or warranty of any kind, express or implied, including but not limited to the fitness of the information for a particular purpose. The user thereof uses the information at his/ her sole risk and liability.



History

Version	Date	Reason	Revised by
0.1	25.01.2017	First version	Vojtěch,Tiansi
0.2	26.01.2017	External review	Diana Krebs
0.3	31.01.2017	External review	Alexandra Garatzogianni

Author List

Organisation	Name	Contact Information
UBONN	Tiansi Dong	tdong@uni-bonn.de
UBONN	Fathoni Musyaffa	musyaffa@iai.uni-bonn.de
UEP	Jaroslav Kuchař	jaroslav.kuchar@fit.cvut.cz
UEP	Stanislav Vojíř	stanislav.vojir@vse.cz
UEP	Václav Zeman	vaclav.zeman@vse.cz
UEP	David Chudán	david.chudan@vse.cz
UEP	Vojtěch Svátek	svatek@vse.cz
OKFGR	Kleanthis Koupidis	koupidis@okfn.gr
OKFGR	Aikaterini Chatzopoulou	kchatzopoul@okfn.gr
OKFGR	Charalampos Bratsas	charalampos.bratsas@okfn.org
Fraunhofer	Fabrizio Orlandi	fabrizio.orlandi@iais.fraunhofer.de
Fraunhofer	Thorsten Merten	thorsten.merten@iais.fraunhofer.de

Executive Summary

In this deliverable we describe computational techniques and systems for statistical analytics and data mining for the OpenBudgets.eu (OBEU) project. Based on Deliverable 2.3. and actual datasets, we developed a highly modularized, de-centralized, and easy extendable systems for financial data analysis and processing within the OBEU project.

We start with a quickstart to install the data-mining modules, then go through each data-mining modules, explaining functions with examples, in particular, descriptive statistics, time series analysis and prediction, comparative analysis, rule/pattern mining, clustering and similarity learning, outlier/anomaly detection.

Although this is a prototype deliverable that encompasses the implementation of several software tools, it also encompasses a sizeable report in particular to map the implemented method to the original user needs as collected in D2.3. with an update of D5.3.

Abbreviations and Acronyms

WP	Work Package
OS	OpenSpending
OBEU	OpenBudgets.eu
OKGR	Open Knowledge Greece
UEP	University of Economics, Prague

Table of Contents

1	Introduction.....	13
2	Short Guideline of Data-mining Modules.....	13
2.1	Installation	13
2.2	How is the data-mining request processed?.....	14
2.3	Data mining system EasyMiner	14
3	Implemented Data-Mining requests.....	15
3.1	Descriptive statistics	15
3.1.1	General description.....	15
	Table 1. Three identified needs in D2.3 can be satisfied by the “DescriptiveStats.OBeu” package.....	16
3.1.2	Input & output.....	16
	User input.....	16
	Table 2. Input of the descriptive statistics algorithm.....	17
	Pre-processing of input.....	17
	Figure 1 - Workflow of the Descriptive Analysis Module.....	17
	Central Tendency Measures.....	17
	Mean.....	18
	Median.....	18
	Dispersion Measures (Measures of Spread).....	18
	Range.....	18
	Quartiles and Interquartile range (IQR).....	18
	Figure 2: An illustration of Quantiles.....	19
	Figure 3: An illustration of Interquartile range.....	19
	Variance.....	19
	Standard Deviation.....	20
	Skewness	20
	Figure 4: An illustration of Skewness.....	21
	Kurtosis.....	21
	Figure 5: An illustration of types of kurtosis.....	22
	Boxplot.....	22
	Figure 6: An illustration of Boxplot and a probability density function of a Normal Population.....	22
	Histogram.....	23
	Bar graph	24
	Correlation.....	24
	Pearson's correlation coefficient (rank correlation coefficient).....	24
	Spearman's rank correlation coefficient	
	Spearman's rank correlation coefficient measures the (statistical) dependence between the ranking of two variables. It is used to assess if the relationship	

between two variables can be described using a monotonic function. The Spearman's coefficient is equal to Pearson's coefficient, but applied to the rank variables. A perfect Spearman correlation of +1 or -1 occurs when each of the variables is a perfect monotone function of the other.....	25
Kendall's Tau b.....	25
Output structure.....	26
Table 3: The main return components of descriptive statistics.....	26
Descriptives.....	27
Table 4: Interpretation of variables in descriptive statistics.....	27
Boxplot.....	27
Table 5: Interpretation of variables in boxplot.....	27
Histogram.....	27
Table 6: Interpretation of variables in histogram.....	28
Frequencies.....	28
Table 7: Interpretation of variables in frequencies.....	28
Correlation.....	28
Table 8: Interpretation of variables in correlation.....	28
3.1.3 Sample case.....	28
Data.....	28
Figure 7.: Fiscal dataset of Municipality of Athens.....	29
Descriptive measures.....	29
Figure 8.: Summary table of basic descriptive measures of Athens in 2004-2015 period.....	29
Boxplot	30
Figure 9.: Snapshot of Executed Expenditure Amounts of Athens in 2004-2015 period.....	30
Histogram.....	30
Figure 10. Histogram representation.....	32
Frequencies.....	32
Figure 11. Frequency representation.....	32
Correlation	33
Figure 12. Correlation representation.....	33
3.2 Time series analysis, predictions.....	34
3.2.1 General description.....	34
Table 9. List of needs covered in Time-Series Analysis.....	35
3.2.2 Input & output.....	35
User input.....	35
Table 10. Table of input parameters for time series analysis.....	36
Pre-processing of input.....	36
Figure 13.- Workflow of Time Series Analysis	37
Stationary tests.....	37
Autocorrelation function (ACF).....	37

Partial autocorrelation function (PACF).....	38
Kwiatkowski-Phillips-Schmidt-Shin test (KPSS).....	39
Augmented Dickey-Fuller test (ADF).....	39
Phillips-Perron test (PP).....	40
Cox Stuart test (CS).....	40
Mann-Kendall Test For Monotonic Trend (MK).....	41
Model Fit- Forecasts.....	41
Output structure.....	43
 Table 11.: The main return components of time series analysis.....	44
acf.param.....	44
Table 12. Table of output variables for autocorrelation function.....	44
Table 13. Table of output variables for partial autocorrelation function.....	44
Table 14. Table of output variables for autocorrelation function for the residual model.....	45
Table 15. Table of output variables for partial autocorrelation function for the residual model.....	45
Decomposition- stl.plot:.....	45
Table 16. Table of output variables in trend analysis.....	45
Forecasts:.....	45
Table 17. Table of output variables in forecast.....	46
3.2.3 Sample case.....	46
Input Data.....	46
Figure 13.: Time Series Data of Revised Expenditure Time Series Data of Municipality of Athens.....	46
Figure 14.: Time Series of Revised Expenditure Time Series Data of Municipality of Athens.....	46
Autocorrelation and Partial autocorrelation.....	47
Figure 15.: An illustration of autocorrelation and partial autocorrelation.....	47
Decomposition.....	47
Figure 16.: Decomposition of Revised Budget Phase Time Series.....	48
ARIMA Model Fit-Forecasts.....	48
Figure 17. Forecasts for 10 years forward.....	48
3.3 Clustering and Similarity learning.....	48
3.3.1 General description.....	48
Table 18. List of needs covered by clustering and similarity analysis.....	49
3.3.2 Input & output.....	50
User input.....	50
Table 19. Table of user input parameters for clustering and similarity analysis.....	50
Pre-processing of input.....	50
Figure 18.: Cluster Analysis Process.....	51
Hierarchical clustering.....	51
Table 20. Table of parameters used in hierarchical clustering.....	52

Table 21. Table of parameters used in linkage clustering.....	52
k-means clustering	52
Partitioning Around Medoids (PAM).....	53
Clustering for Large Applications (CLARA).....	54
Fuzzy clustering.....	54
Model Based Clustering.....	55
Principal Component Analysis	55
Figure. 19: Example of convex hulls and ellipses that visualize borders of clusters.....	56
Output structure.....	57
Table 22. List of output components of clustering analysis.....	57
Hierarchical Clustering.....	57
Table 23. List of output components of hierarchical clustering analysis.....	57
K-Means.....	57
Table 24. List of output components of k-means clustering analysis.....	58
Partitioning Around Medoids (Pam).....	58
Table 25. List of output components of Partitioning Around Medoids (clustering analysis).....	58
Clustering Large Applications (Clara).....	58
Table 26. List of output components of Clustering Large Applications (clustering analysis).....	59
Fuzzy Analysis Clustering (Fanny).....	59
Table 27. List of output components of Fuzzy Analysis Clustering (clustering analysis).....	59
Model Based Clustering.....	59
Table 28. List of output components of Model Based Clustering (clustering analysis).....	59
3.3.3 Sample case.....	59
Data.....	60
Figure 20.: Fiscal dataset of Athens and Thessaloniki.....	60
Figure 21.: Partitioning Around Medoids Visualization.....	61
Table 29.: The most representative expenditure budget phases amounts of Municipalities of Athens and Thessaloniki.....	61
3.4 Comparative analysis.....	62
3.4.1 General description.....	62
Table 30.: Comparative packages fulfills eight needs in D2.3.....	63
3.4.2 input & output.....	63
User input.....	63
Pre-processing of input.....	63
Log-likelihood.....	63
Akaike information criterion.....	64
Bayesian information criterion.....	65
Silhouette Visualization.....	65

Dunn's partition coefficient	65
Output structure.....	66
Time Serie Decomposition.....	66
Table 31.: Lists of parameters in Time Serie Decomposition.....	67
Model Fitting.....	67
Table 31.: Lists of parameters in model fitting.....	68
Hierarchical Cluster Analysis.....	68
Table 32.: List of parameters in hierarchical cluster analysis.....	68
K-Means Cluster Analysis.....	68
Table 33.: List of parameters in k-means cluster analysis.....	68
Partitioning Around Medoids (Pam).....	68
Table 34.: List of parameters in partitioning around medoids analysis.....	69
Silhouette Visualization.....	69
Table 35.: List of parameters in Silhouette Visualization.....	69
Clustering Large Applications Algorithm.....	69
Table 36.: List of parameters in Clustering Large Applications.....	69
Fuzzy Analysis Clustering.....	69
Table 37.: List of parameters in Fuzzy Analysis Clustering.....	70
Model Based Clustering.....	70
Table 38.: Lists of parameters in Model Based Clustering.....	71
3.4.3 Sample case.....	71
Data.....	71
Figure 22.: Fiscal dataset of Athens and Thessaloniki.....	72
Descriptive measures.....	72
Figure 23.: Summary table of basic descriptive measures of Executed Expenditure amounts of Athens and Thessaloniki in 2011-2015 period.....	73
Boxplot	73
Figure 24.: Boxplots of executed expenditures in Athens and Thessaloniki in 2004-2015 period.....	74
Frequencies- Bar graph.....	74
Figure 25.: Executed amounts in Athens and Thessaloniki from 2011-2015.....	75
Time Series.....	75
Figure 26.: Time series of Executed Expenditure amounts in Athens and Thessaloniki.....	75
Clusters Analysis evaluation.....	75
Figure 27.: Cluster silhouette plot.....	76
3.5 Rule/pattern mining.....	76
3.5.1 General description.....	76
Table 39.: Rule-mining packages fulfil three requirements in D2.3.....	77
3.5.2 GUHA (complex) association rules.....	77
Task results.....	78
Table 40. List of parameters used in rule/pattern mining.....	78
Table 41. List of measures in rule/pattern mining.....	78

3.5.3 Input & output.....	79
User input.....	79
Pre-processing of input.....	79
Task definition.....	80
Output structure.....	81
Visualization.....	83
3.5.4 Sample case.....	83
Analyzed dataset.....	83
Data preprocessing.....	84
Preprocessed dataset description.....	84
Table. 42. List of parameters of pre-processed input data.....	85
Data mining using EasyMiner API.....	85
Example data mining task.....	85
Figure 28. Graphical UI of association rule pattern in EasyMiner system.....	86
Figure 29. Result of the rule-mining.....	87
Interpretation of results - association rules.....	87
Figure 30. Interpretation of a simple rule.....	87
Figure 31. Interpretation of a longer rule.....	88
Figure 32. Interpretation of several rules.....	88
3.6 Outlier/anomaly detection	89
3.6.1 General description.....	89
Table 43.: Outlier-detection packages fulfil six requirements in D2.3.....	90
3.6.1.1 Local Outlier Factors based on Subpopulation.....	90
Generating possible constraints	90
Finding subpopulations	91
Figure 33. A lattice of subpopulation.	91
Outlier detection within a subpopulation and outlier scores	91
Figure. 34.. The density of A is much lower than densities of its neighbors, so A is an outlier	92
Outlier score and its interpretation	93
Figure 35. Data-items are clustered based on ratio of densities.....	93
3.6.1.2 Frequent patterns.....	93
3.6.1.3 Financial ratios.....	94
3.6.2 input & output.....	95
3.6.2.1 Local Outlier Factors based on Subpopulation.....	95
User input.....	95
Pre-processing of input.....	95
Output structure.....	95
3.6.2.2 Frequent patterns.....	95
User input.....	95
Pre-processing of input.....	95
3.6.2.3 Financial ratios.....	96
3.6.3 Sample case.....	96

3.6.3.1 Local Outlier Factors based on Subpopulation	96
User input	96
Figure 36.: A sample user interface for subpopulation-based LOF outlier-detection.....	96
Pre-processing and the input to the core algorithm.....	96
Figure 37.: Automatically generated Sparql query to extract data items from selected RDF files.....	97
Figure 38.: A CSV file is automatically generated for the input of the algorithm. .	98
Output of the core algorithm.....	98
Figure 39. The top 25 outlier data-items are saved in a csv file.....	99
3.6.3.2 Frequent patterns.....	99
Figure 40: Visualization of the anomaly instance - it is composed from less frequent items (red bars in the middle of each individual bar plot).....	101
Figure 41: Visualization of the regular instance - it is composed from more more frequent items (red bars in each individual bar plot).....	101
3.6.3.3 Financial ratios	102
Figure 42: Financial ratios visualization.....	103
Figure 43: Detail of the EAFRD fund for Netherlands.....	103
Figure 44: Detail of the EAFRD fund for Luxembourg.....	104
4 Guidance on methods and comparison to requirements.....	104
4.1 Relevant situations for applying the mining methods.....	105
4.2 Coverage of end-user requirements.....	106
Table 44. List of 37 needs in D2.3, with an evaluation to the coverage by data-mining tools.....	112
5 Conclusions and Ongoing work.....	112
Table 45. A statistics of the status of coverage, and needs.....	113
References.....	114
Referenced Internet Links:.....	116
7. Appendix.....	117
7.1 lists of open-source data-mining tools developed till now.....	117
7.2 Installation (Ubuntu) and start the server.....	117
Figure A1.: Screenshot of a running redis-server.....	119
Figure A2.: Screenshot of a running DAM back-end.....	119

List of Figures

1Introduction.....	20
2Short Guideline of Data-mining Modules.....	20
2.1 Installation	20
2.2 How is the data-mining request processed?.....	20
2.3 Data mining system EasyMiner	21
3Implemented Data-Mining requests.....	22
3.1 Descriptive statistics	22
3.1.1 General description.....	22
Table 1. Three identified needs in D2.3 can be satisfied by the “DescriptiveStats.OBeu” package.....	23
3.1.2 Input & output.....	23
User input.....	23
Table 2. Input of the descriptive statistics algorithm.....	24
Pre-processing of input.....	24
Figure 1 - Workflow of the Descriptive Analysis Module.....	24
Central Tendency Measures.....	24
Mean.....	25
Median.....	25
Dispersion Measures (Measures of Spread).....	25
Range.....	25
Quartiles and Interquartile range (IQR).....	25
Figure 2: An illustration of Quantiles.....	26
Figure 3: An illustration of Interquartile range.....	26
Variance.....	26
Standard Deviation.....	27
Skewness	27
Figure 4: An illustration of Skewness.....	27
Kurtosis.....	28
Figure 5: An illustration of types of kurtosis.....	28
Boxplot.....	29
Figure 6: An illustration of Boxplot and a probability density function of a Normal Population.....	29
Histogram.....	29
Bar graph	30
Correlation.....	31
Pearson's correlation coefficient (rank correlation coefficient).....	31
Spearman's rank correlation coefficient	
Spearman's rank correlation coefficient measures the (statistical) dependence between the ranking of two variables. It is used to assess if the relationship	

between two variables can be described using a monotonic function. The Spearman's coefficient is equal to Pearson's coefficient, but applied to the rank variables. A perfect Spearman correlation of +1 or -1 occurs when each of the variables is a perfect monotone function of the other.....	31
Kendall's Tau b.....	32
Output structure.....	32
Table 3: The main return components of descriptive statistics.....	33
Descriptives.....	33
Table 4: Interpretation of variables in descriptive statistics.....	34
Boxplot.....	34
Table 5: Interpretation of variables in boxplot.....	34
Histogram.....	34
Table 6: Interpretation of variables in histogram.....	34
Frequencies.....	34
Table 7: Interpretation of variables in frequencies.....	34
Correlation.....	35
Table 8: Interpretation of variables in correlation.....	35
3.1.3 Sample case.....	35
Data.....	35
Figure 7.: Fiscal dataset of Municipality of Athens.....	36
Descriptive measures.....	36
Figure 8.: Summary table of basic descriptive measures of Athens in 2004-2015 period.....	36
Boxplot	36
Figure 9.: Snapshot of Executed Expenditure Amounts of Athens in 2004-2015 period.....	37
Histogram.....	37
Figure 10. Histogram representation.....	38
Frequencies.....	39
Figure 11. Frequency representation.....	39
Correlation	39
Figure 12. Correlation representation.....	40
3.2 Time series analysis, predictions.....	40
3.2.1 General description.....	40
Table 9. List of needs covered in Time-Series Analysis.....	42
3.2.2 Input & output.....	42
User input.....	42
Table 10. Table of input parameters for time series analysis.....	42
Pre-processing of input.....	42
Figure 13.- Workflow of Time Series Analysis	43
Stationary tests.....	44
Autocorrelation function (ACF).....	44

Partial autocorrelation function (PACF).....	45
Kwiatkowski-Phillips-Schmidt-Shin test (KPSS).....	45
Augmented Dickey-Fuller test (ADF).....	46
Phillips-Perron test (PP).....	46
Cox Stuart test (CS).....	46
Mann-Kendall Test For Monotonic Trend (MK).....	47
Model Fit- Forecasts.....	48
Output structure.....	49
 Table 11.: The main return components of time series analysis.....	50
acf.param.....	50
Table 12. Table of output variables for autocorrelation function.....	50
Table 13. Table of output variables for partial autocorrelation function.....	51
Table 14. Table of output variables for autocorrelation function for the residual model.....	51
Table 15. Table of output variables for partial autocorrelation function for the residual model.....	51
Decomposition- stl.plot:.....	51
Table 16. Table of output variables in trend analysis.....	52
Forecasts:.....	52
Table 17. Table of output variables in forecast.....	52
3.2.3 Sample case.....	52
Input Data.....	52
Figure 13.: Time Series Data of Revised Expenditure Time Series Data of Municipality of Athens.....	53
Figure 14.: Time Series of Revised Expenditure Time Series Data of Municipality of Athens.....	53
Autocorrelation and Partial autocorrelation.....	53
Figure 15.: An illustration of autocorrelation and partial autocorrelation.....	53
Decomposition.....	54
Figure 16.: Decomposition of Revised Budget Phase Time Series.....	54
ARIMA Model Fit-Forecasts.....	54
Figure 17. Forecasts for 10 years forward.....	55
3.3 Clustering and Similarity learning.....	55
3.3.1 General description.....	55
Table 18. List of needs covered by clustering and similarity analysis.....	56
3.3.2 Input & output.....	56
User input.....	56
Table 19. Table of user input parameters for clustering and similarity analysis.....	57
Pre-processing of input.....	57
Figure 18.: Cluster Analysis Process.....	57
Hierarchical clustering.....	57
Table 20. Table of parameters used in hierarchical clustering.....	58

Table 21. Table of parameters used in linkage clustering.....	59
k-means clustering	59
Partitioning Around Medoids (PAM).....	60
Clustering for Large Applications (CLARA).....	60
Fuzzy clustering.....	61
Model Based Clustering.....	61
Principal Component Analysis	62
Figure. 19: Example of convex hulls and ellipses that visualize borders of clusters.....	63
Output structure.....	63
Table 22. List of output components of clustering analysis.....	63
Hierarchical Clustering.....	64
Table 23. List of output components of hierarchical clustering analysis.....	64
K-Means.....	64
Table 24. List of output components of k-means clustering analysis.....	64
Partitioning Around Medoids (Pam).....	64
Table 25. List of output components of Partitioning Around Medoids (clustering analysis).....	65
Clustering Large Applications (Clara).....	65
Table 26. List of output components of Clustering Large Applications (clustering analysis).....	65
Fuzzy Analysis Clustering (Fanny).....	65
Table 27. List of output components of Fuzzy Analysis Clustering (clustering analysis).....	66
Model Based Clustering.....	66
Table 28. List of output components of Model Based Clustering (clustering analysis).....	66
3.3.3 Sample case.....	66
Data.....	66
Figure 20.: Fiscal dataset of Athens and Thessaloniki.....	67
Figure 21.: Partitioning Around Medoids Visualization.....	68
Table 29.: The most representative expenditure budget phases amounts of Municipalities of Athens and Thessaloniki.....	68
3.4 Comparative analysis.....	69
3.4.1 General description.....	69
Table 30.: Comparative packages fulfills eight needs in D2.3.....	70
3.4.2 input & output.....	70
User input.....	70
Pre-processing of input.....	70
Log-likelihood.....	70
Akaike information criterion.....	71
Bayesian information criterion.....	71
Silhouette Visualization.....	72

Dunn's partition coefficient	72
Output structure.....	73
Time Serie Decomposition.....	73
Table 31.: Lists of parameters in Time Serie Decomposition.....	74
Model Fitting.....	74
Table 31.: Lists of parameters in model fitting.....	75
Hierarchical Cluster Analysis.....	75
Table 32.: List of parameters in hierarchical cluster analysis.....	75
K-Means Cluster Analysis.....	75
Table 33.: List of parameters in k-means cluster analysis.....	75
Partitioning Around Medoids (Pam).....	75
Table 34.: List of parameters in partitioning around medoids analysis.....	76
Silhouette Visualization.....	76
Table 35.: List of parameters in Silhouette Visualization.....	76
Clustering Large Applications Algorithm.....	76
Table 36.: List of parameters in Clustering Large Applications.....	76
Fuzzy Analysis Clustering.....	76
Table 37.: List of parameters in Fuzzy Analysis Clustering.....	77
Model Based Clustering.....	77
Table 38.: Lists of parameters in Model Based Clustering.....	78
3.4.3 Sample case.....	78
Data.....	78
Figure 22.: Fiscal dataset of Athens and Thessaloniki.....	79
Descriptive measures.....	79
Figure 23.: Summary table of basic descriptive measures of Executed Expenditure amounts of Athens and Thessaloniki in 2011-2015 period.....	79
Boxplot	80
Figure 24.: Boxplots of executed expenditures in Athens and Thessaloniki in 2004-2015 period.....	80
Frequencies- Bar graph.....	81
Figure 25.: Executed amounts in Athens and Thessaloniki from 2011-2015.....	81
Time Series.....	81
Figure 26.: Time series of Executed Expenditure amounts in Athens and Thessaloniki.....	82
Clusters Analysis evaluation.....	82
Figure 27.: Cluster silhouette plot.....	82
3.5 Rule/pattern mining.....	82
3.5.1 General description.....	82
Table 39.: Rule-mining packages fulfil three requirements in D2.3.....	84
3.5.2 GUHA (complex) association rules.....	84
Task results.....	84
Table 40. List of parameters used in rule/pattern mining.....	85
Table 41. List of measures in rule/pattern mining.....	85

3.5.3 Input & output.....	85
User input.....	85
Pre-processing of input.....	86
Task definition.....	86
Output structure.....	88
Visualization.....	89
3.5.4 Sample case.....	90
Analyzed dataset.....	90
Data preprocessing.....	90
Preprocessed dataset description.....	91
Table. 42. List of parameters of pre-processed input data.....	91
Data mining using EasyMiner API.....	91
Example data mining task.....	92
Figure 28. Graphical UI of association rule pattern in EasyMiner system.....	92
Figure 29. Result of the rule-mining.....	93
Interpretation of results - association rules.....	93
Figure 30. Interpretation of a simple rule.....	94
Figure 31. Interpretation of a longer rule.....	94
Figure 32. Interpretation of several rules.....	94
3.6 Outlier/anomaly detection	95
3.6.1 General description.....	95
Table 43.: Outlier-detection packages fulfil six requirements in D2.3.....	96
3.6.1.1 Local Outlier Factors based on Subpopulation.....	96
Generating possible constraints	96
Finding subpopulations	97
Figure 33. A lattice of subpopulation.	97
Outlier detection within a subpopulation and outlier scores	97
Figure. 34.. The density of A is much lower than densities of its neighbors, so A is an outlier	98
Outlier score and its interpretation	99
Figure 35. Data-items are clustered based on ratio of densities.....	99
3.6.1.2 Frequent patterns.....	99
3.6.1.3 Financial ratios.....	100
3.6.2 input & output.....	101
3.6.2.1 Local Outlier Factors based on Subpopulation.....	101
User input.....	101
Pre-processing of input.....	101
Output structure.....	101
3.6.2.2 Frequent patterns.....	101
User input.....	101
Pre-processing of input.....	101
3.6.2.3 Financial ratios.....	102
3.6.3 Sample case.....	102

3.6.3.1 Local Outlier Factors based on Subpopulation	102
User input	102
Figure 36.: A sample user interface for subpopulation-based LOF outlier-detection.....	102
Pre-processing and the input to the core algorithm.....	102
Figure 37.: Automatically generated Sparql query to extract data items from selected RDF files.....	103
Figure 38.: A CSV file is automatically generated for the input of the algorithm	104
Output of the core algorithm.....	104
Figure 39. The top 25 outlier data-items are saved in a csv file.....	105
3.6.3.2 Frequent patterns.....	105
Figure 40: Visualization of the anomaly instance - it is composed from less frequent items (red bars in the middle of each individual bar plot).....	107
Figure 41: Visualization of the regular instance - it is composed from more more frequent items (red bars in each individual bar plot).....	107
3.6.3.3 Financial ratios	108
Figure 42: Financial ratios visualization.....	109
Figure 43: Detail of the EAFRD fund for Netherlands.....	109
Figure 44: Detail of the EAFRD fund for Luxembourg.....	110
4 Guidance on methods and comparison to requirements.....	110
4.1 Relevant situations for applying the mining methods.....	111
4.2 Coverage of end-user requirements.....	112
Table 44. List of 37 needs in D2.3, with an evaluation to the coverage by data-mining tools.....	118
5 Conclusions and Ongoing work.....	118
Table 45. A statistics of the status of coverage, and needs.....	119
References.....	120
Referenced Internet Links:.....	122
7. Appendix.....	123
7.1 lists of open-source data-mining tools developed till now.....	123
7.2 Installation (Ubuntu) and start the server.....	123
Figure A1.: Screenshot of a running redis-server.....	125
Figure A2.: Screenshot of a running DAM back-end.....	125

List of Tables

1	Introduction.....	26
2	Short Guideline of Data-mining Modules.....	27
2.1	Installation	27
2.2	How is the data-mining request processed?.....	27
2.3	Data mining system EasyMiner	28
3	Implemented Data-Mining requests.....	29
3.1	Descriptive statistics	29
3.1.1	General description.....	29
Table 1.	Three identified needs in D2.3 can be satisfied by the “DescriptiveStats.OBeu” package.....	29
3.1.2	Input & output.....	30
User input.....	30	
Table 2.	Input of the descriptive statistics algorithm.....	30
Pre-processing of input.....	30	
Figure 1 -	Workflow of the Descriptive Analysis Module.....	31
Central Tendency Measures.....	31	
Mean.....	31	
Median.....	32	
Dispersion Measures (Measures of Spread).....	32	
Range.....	32	
Quartiles and Interquartile range (IQR).....	32	
Figure 2:	An illustration of Quantiles.....	32
Figure 3:	An illustration of Interquartile range.....	33
Variance.....	33	
Standard Deviation.....	33	
Skewness	34	
Figure 4:	An illustration of Skewness.....	34
Kurtosis.....	35	
Figure 5:	An illustration of types of kurtosis.....	35
Boxplot.....	35	
Figure 6:	An illustration of Boxplot and a probability density function of a Normal Population.....	36
Histogram.....	36	
Bar graph	37	
Correlation.....	37	
Pearson's correlation coefficient (rank correlation coefficient).....	38	
Spearman's rank correlation coefficient		
Spearman's rank correlation coefficient measures the (statistical) dependence		

between the ranking of two variables. It is used to assess if the relationship between two variables can be described using a monotonic function. The Spearman's coefficient is equal to Pearson's coefficient, but applied to the rank variables. A perfect Spearman correlation of +1 or -1 occurs when each of the variables is a perfect monotone function of the other.....	38
Kendall's Tau b.....	39
Output structure.....	39
Table 3: The main return components of descriptive statistics.....	40
Descriptives.....	40
Table 4: Interpretation of variables in descriptive statistics.....	40
Boxplot.....	41
Table 5: Interpretation of variables in boxplot.....	41
Histogram.....	41
Table 6: Interpretation of variables in histogram.....	41
Frequencies.....	41
Table 7: Interpretation of variables in frequencies.....	41
Correlation.....	41
Table 8: Interpretation of variables in correlation.....	42
3.1.3 Sample case.....	42
Data.....	42
Figure 7.: Fiscal dataset of Municipality of Athens.....	43
Descriptive measures.....	43
Figure 8.: Summary table of basic descriptive measures of Athens in 2004-2015 period.....	43
Boxplot	43
Figure 9.: Snapshot of Executed Expenditure Amounts of Athens in 2004-2015 period.....	44
Histogram.....	44
Figure 10. Histogram representation.....	45
Frequencies.....	46
Figure 11. Frequency representation.....	46
Correlation	46
Figure 12. Correlation representation.....	47
3.2 Time series analysis, predictions.....	47
3.2.1 General description.....	47
Table 9. List of needs covered in Time-Series Analysis.....	49
3.2.2 Input & output.....	49
User input.....	49
Table 10. Table of input parameters for time series analysis.....	49
Pre-processing of input.....	49
Figure 13.- Workflow of Time Series Analysis	50
Stationary tests.....	51

Autocorrelation function (ACF).....	51
Partial autocorrelation function (PACF).....	52
Kwiatkowski-Phillips-Schmidt-Shin test (KPSS).....	52
Augmented Dickey-Fuller test (ADF).....	53
Phillips-Perron test (PP).....	53
Cox Stuart test (CS).....	53
Mann-Kendall Test For Monotonic Trend (MK).....	54
Model Fit- Forecasts.....	55
Output structure.....	56
 Table 11.: The main return components of time series analysis.....	57
acf.param.....	57
Table 12. Table of output variables for autocorrelation function.....	57
Table 13. Table of output variables for partial autocorrelation function.....	58
Table 14. Table of output variables for autocorrelation function for the residual model.....	58
Table 15. Table of output variables for partial autocorrelation function for the residual model.....	58
Decomposition- stl.plot:.....	58
Table 16. Table of output variables in trend analysis.....	59
Forecasts:.....	59
Table 17. Table of output variables in forecast.....	59
3.2.3 Sample case.....	59
Input Data.....	59
Figure 13.: Time Series Data of Revised Expenditure Time Series Data of Municipality of Athens.....	60
Figure 14.: Time Series of Revised Expenditure Time Series Data of Municipality of Athens.....	60
Autocorrelation and Partial autocorrelation.....	60
Figure 15.: An illustration of autocorrelation and partial autocorrelation.....	60
Decomposition.....	61
Figure 16.: Decomposition of Revised Budget Phase Time Series.....	61
ARIMA Model Fit-Forecasts.....	61
Figure 17. Forecasts for 10 years forward.....	62
3.3 Clustering and Similarity learning.....	62
3.3.1 General description.....	62
Table 18. List of needs covered by clustering and similarity analysis.....	63
3.3.2 Input & output.....	63
User input.....	63
Table 19. Table of user input parameters for clustering and similarity analysis...64	64
Pre-processing of input.....	64
Figure 18.: Cluster Analysis Process.....	64
Hierarchical clustering.....	64

Table 20. Table of parameters used in hierarchical clustering.....	65
Table 21. Table of parameters used in linkage clustering.....	66
k-means clustering	66
Partitioning Around Medoids (PAM).....	67
Clustering for Large Applications (CLARA).....	67
Fuzzy clustering.....	68
Model Based Clustering.....	68
Principal Component Analysis	69
Figure. 19: Example of convex hulls and ellipses that visualize borders of clusters.....	70
Output structure.....	70
Table 22. List of output components of clustering analysis.....	70
Hierarchical Clustering.....	71
Table 23. List of output components of hierarchical clustering analysis.....	71
K-Means.....	71
Table 24. List of output components of k-means clustering analysis.....	71
Partitioning Around Medoids (Pam).....	71
Table 25. List of output components of Partitioning Around Medoids (clustering analysis).....	72
Clustering Large Applications (Clara).....	72
Table 26. List of output components of Clustering Large Applications (clustering analysis).....	72
Fuzzy Analysis Clustering (Fanny).....	72
Table 27. List of output components of Fuzzy Analysis Clustering (clustering analysis).....	73
Model Based Clustering.....	73
Table 28. List of output components of Model Based Clustering (clustering analysis).....	73
3.3.3 Sample case.....	73
Data.....	73
Figure 20.: Fiscal dataset of Athens and Thessaloniki.....	74
Figure 21.: Partitioning Around Medoids Visualization.....	75
Table 29.: The most representative expenditure budget phases amounts of Municipalities of Athens and Thessaloniki.....	75
3.4 Comparative analysis.....	76
3.4.1 General description.....	76
Table 30.: Comparative packages fulfills eight needs in D2.3.....	77
3.4.2 input & output.....	77
User input.....	77
Pre-processing of input.....	77
Log-likelihood.....	77
Akaike information criterion.....	78
Bayesian information criterion.....	78

Silhouette Visualization.....	79
Dunn's partition coefficient	79
Output structure.....	80
Time Serie Decomposition.....	80
Table 31.: Lists of parameters in Time Serie Decomposition.....	81
Model Fitting.....	81
Table 31.: Lists of parameters in model fitting.....	82
Hierarchical Cluster Analysis.....	82
Table 32.: List of parameters in hierarchical cluster analysis.....	82
K-Means Cluster Analysis.....	82
Table 33.: List of parameters in k-means cluster analysis.....	82
Partitioning Around Medoids (Pam).....	82
Table 34.: List of parameters in partitioning around medoids analysis.....	83
Silhouette Visualization.....	83
Table 35.: List of parameters in Silhouette Visualization.....	83
Clustering Large Applications Algorithm.....	83
Table 36.: List of parameters in Clustering Large Applications.....	83
Fuzzy Analysis Clustering.....	83
Table 37.: List of parameters in Fuzzy Analysis Clustering.....	84
Model Based Clustering.....	84
Table 38.: Lists of parameters in Model Based Clustering.....	85
3.4.3 Sample case.....	85
Data.....	85
Figure 22.: Fiscal dataset of Athens and Thessaloniki.....	86
Descriptive measures.....	86
Figure 23.: Summary table of basic descriptive measures of Executed Expenditure amounts of Athens and Thessaloniki in 2011-2015 period.....	86
Boxplot	87
Figure 24.: Boxplots of executed expenditures in Athens and Thessaloniki in 2004-2015 period.....	87
Frequencies- Bar graph.....	88
Figure 25.: Executed amounts in Athens and Thessaloniki from 2011-2015.....	88
Time Series.....	88
Figure 26.: Time series of Executed Expenditure amounts in Athens and Thessaloniki.....	89
Clusters Analysis evaluation.....	89
Figure 27.: Cluster silhouette plot.....	89
3.5 Rule/pattern mining.....	89
3.5.1 General description.....	89
Table 39.: Rule-mining packages fulfil three requirements in D2.3.....	91
3.5.2 GUHA (complex) association rules.....	91
Task results.....	91
Table 40. List of parameters used in rule/pattern mining.....	92

Table 41. List of measures in rule/pattern mining.....	92
3.5.3 Input & output.....	92
User input.....	92
Pre-processing of input.....	93
Task definition.....	93
Output structure.....	95
Visualization.....	96
3.5.4 Sample case.....	97
Analyzed dataset.....	97
Data preprocessing.....	97
Preprocessed dataset description.....	98
Table. 42. List of parameters of pre-processed input data.....	98
Data mining using EasyMiner API.....	98
Example data mining task.....	99
Figure 28. Graphical UI of association rule pattern in EasyMiner system.....	99
Figure 29. Result of the rule-mining.....	100
Interpretation of results - association rules.....	100
Figure 30. Interpretation of a simple rule.....	101
Figure 31. Interpretation of a longer rule.....	101
Figure 32. Interpretation of several rules.....	101
3.6 Outlier/anomaly detection	102
3.6.1 General description.....	102
Table 43.: Outlier-detection packages fulfil six requirements in D2.3.....	103
3.6.1.1 Local Outlier Factors based on Subpopulation.....	103
Generating possible constraints	103
Finding subpopulations	104
Figure 33. A lattice of subpopulation.	104
Outlier detection within a subpopulation and outlier scores	104
Figure. 34.. The density of A is much lower than densities of its neighbors, so A is an outlier	105
Outlier score and its interpretation	106
Figure 35. Data-items are clustered based on ratio of densities.....	107
3.6.1.2 Frequent patterns.....	107
3.6.1.3 Financial ratios.....	108
3.6.2 input & output.....	108
3.6.2.1 Local Outlier Factors based on Subpopulation.....	108
User input.....	108
Pre-processing of input.....	108
Output structure.....	109
3.6.2.2 Frequent patterns.....	109
User input.....	109
Pre-processing of input.....	109
3.6.2.3 Financial ratios.....	109

3.6.3 Sample case.....	109
3.6.3.1 Local Outlier Factors based on Subpopulation	109
User input	109
Figure 36.: A sample user interface for subpopulation-based LOF outlier-detection.....	110
Pre-processing and the input to the core algorithm.....	110
Figure 37.: Automatically generated Sparql query to extract data items from selected RDF files.....	110
Figure 38.: A CSV file is automatically generated for the input of the algorithm.....	111
Output of the core algorithm.....	111
Figure 39. The top 25 outlier data-items are saved in a csv file.....	112
3.6.3.2 Frequent patterns.....	112
Figure 40: Visualization of the anomaly instance - it is composed from less frequent items (red bars in the middle of each individual bar plot).....	114
Figure 41: Visualization of the regular instance - it is composed from more frequent items (red bars in each individual bar plot).....	114
3.6.3.3 Financial ratios	115
Figure 42: Financial ratios visualization.....	116
Figure 43: Detail of the EAFRD fund for Netherlands.....	116
Figure 44: Detail of the EAFRD fund for Luxembourg.....	117
4 Guidance on methods and comparison to requirements.....	117
4.1 Relevant situations for applying the mining methods.....	118
4.2 Coverage of end-user requirements.....	119
Table 44. List of 37 needs in D2.3, with an evaluation to the coverage by data-mining tools.....	125
5 Conclusions and Ongoing work.....	125
Table 45. A statistics of the status of coverage, and needs.....	126
References.....	127
Referenced Internet Links:.....	129
7. Appendix.....	130
7.1 lists of open-source data-mining tools developed till now.....	130
7.2 Installation (Ubuntu) and start the server.....	130
Figure A1.: Screenshot of a running redis-server.....	132
Figure A2.: Screenshot of a running DAM back-end.....	132

1 Introduction

The development of Task 2.4 Data Mining and Statistical Analytics Techniques mainly follows the Deliverable 2.3 Requirements for Statistical Analysis and Data Mining, updated with the Deliverable 5.3 and available datasets. We integrated and adapted existing data-mining tools, and implemented new data-mining methods which fit the financial domain within OBEU. Softwares developed in this working package can be downloaded from Github and are listed in Appendix 7.1.

The rest of this document is structured as follows: Section 2 is a short guide-line to install these software tools, and a quickstart to use them. Section 3 describes the detailed implementation of several data-mining tasks, including descriptive statistics, comparative analysis, time series analysis and prediction, rule/pattern mining, clustering and similarity learning, outlier/anomaly detection. Section 4 summarizes how many requirements have been fulfilled or partially fulfilled by these data-mining algorithms, based on the Deliverable 5.3. Section 5 summarizes the whole deliverable, and describes some on-going research works.

2 Short Guideline of Data-mining Modules

In this section, we list web locations of the open source softwares developed for data analysis and mining for the OBEU project, and present a gentle guideline to install, update, and use these tools.

Software modules developed for data analysis and mining are available at
<https://github.com/openbudgets>
<https://github.com/kizi/easyminer>
<https://github.com/okgreece>

2.1 Installation

The installation of the data-mining base module on a local Ubuntu platform is described in the README.md file at

https://github.com/openbudgets/DAM/tree/staging_indigo.

and also described in the Appendix 7.2 of this document.

2.2 How is the data-mining request processed?

When users send a data-mining request to the backend, they shall send three pieces of information: (1) dataset(s), (2) the name of data-mining function, (3) parameters to the function.

From the function name and parameters, the backend server decides what kind of pre-processing is going to be done. The pre-processing task is conducted by the **preprocessing_dm** module¹.

From the function name, the backend server decides where this task will be processed -- at UEP data-mining server, or at OKFGR data-mining server², or locally. Communicating with the UEP data-mining server is carried out by **uep_dm** module³; Communicating with the OKFGR data-mining server is carried out by **okfgr_dm** module⁴. If the task shall be processed locally, the backend-server import a local module. For example, the task of outlier-detection based on LOF (local outlier factor) is processed by the local module **outlier_dm**⁵.

2.3 Data mining system EasyMiner

Data mining system EasyMiner for association rules mining is developed on the University of Economics, Prague. The full system is based on composition of RESTful web services. The main parts are:

- *EasyMinerCenter* (central, main user access component, front-end and API endpoint)
- *Data service* (service for upload and management of user data)

¹https://github.com/openbudgets/preprocessing_dm

²<http://okfnrg.math.auth.gr/ocpu/test/>

³https://github.com/openbudgets/uep_dm

⁴https://github.com/openbudgets/okfgr_dm

⁵https://github.com/openbudgets/outlier_dm

- *Preprocessing service* (service with implementation of data preprocessing)
- *Mining service* (service for association rule mining [using algorithms *apriori* and *fp-growth*] and association rule pruning [using algorithm *rCBA*])
- *EasyMiner-Scorer* (component for evaluation of classification models)

These web services are connected to one functional complex system. Each user registers a custom user account and then it is equivalent to use both – REST API or graphical user interface. The main endpoint for the user is the component EasyMinerCenter. The API of this component is used with integration components of the project OpenBudgets (mainly with the component DAM).

EasyMiner components are licensed under Apache License, Version 2.0. The source code of the version EasyMiner/R (there is another version with LISp-Miner backend, currently under the development, see Section 3.5.2) are public available in the GitHub repository <https://github.com/kizi/easyminer> and for the installation purposes are available also Docker images.

API usage example is available on: <https://github.com/KIZI/EasyMiner-EasyMinerCenter/wiki/API-usage-manual>

3 Implemented Data-Mining requests

3.1 Descriptive statistics

Kleanthis, Aikaterini, Charalampos

3.1.1 General description

It may not be easy to understand visualizations of raw data. Descriptive statistics is the data analysis that describes, shows, or summarizes data in a meaningful way: it captures simple patterns that could be hidden in the data. This kind of analysis is very important since it allows simpler interpretation of the data.

We developed “*DescriptiveStats.OBeu*” package to enable the calculation of descriptive statistical measures in Budget data of municipalities across Europe, form the basis of the their structures and provide simple visualization in order to meet users’ needs and eventually the tasks described in detail in Deliverable 2.3- “*Requirements for Statistical Analytics and Data Mining Techniques*” and summarized in the following table. This package was built in R Software environment and it is available in Github⁶.

⁶<https://github.com/okgreece/DescriptiveStats.OBeu>

Need	Description	Discussion	Task No.
N08	Perform aggregations and simple statistics	As this need refers to users unexperienced in budgeting, the focus for the aggregations and statistics performed lies on a user-friendly interface.	T07
N17	Consider fiscal indicators like error, performance and absorption rates	After calculating these fiscal indicator we will apply statistics with a focus on trends.	T16
N30	Include actual statistics	This need extends the already formulated requirement (R03) extracted from needs (N19) and (N27) to also include actual statistics. These statistics can be incorporated in OBEU in two ways: First providing the statistics as additional information to the data and second directly in the analysis to enhance the results.	-

Table 1. Three identified needs in D2.3 can be satisfied by the “*DescriptiveStats.OBeu*” package

This package includes functions for measuring central tendency and dispersion of numeric variables along with their distributions and correlations and the frequencies of categorical variables for a given dataset that are used in OpenBudgets.eu (OBEU) fiscal datasets. “*DescriptiveStats.OBeu*” is based on *jsonlite*⁷ and *reshape*⁸ R libraries.

3.1.2 Input & output

User input

The user should define the “*dimensions*”, “*measured.dimensions*” and “*amounts*” parameters to form the dataset. Then there is an automated process that calculates the basic descriptive measures of tendency and spread, boxplot and histogram parameters in order to describe and visualize the distribution characteristics of the desired dataset.

The user can also interact and select whether or not outliers should be considered and if so, the level of the coefficient outliers in the boxplot can further be defined. In addition the correlation coefficient can be also selected for the correlation matrix returns- available coefficients are “pearson” (default), “kendall” or “spearman”.

⁷<https://cran.r-project.org/web/packages/jsonlite/>

⁸<https://cran.r-project.org/web/packages/reshape/>

Input	Description
json_data	The json string, URL or file from Open Spending API
dimensions	The dimensions of the input data
amounts	The measures of the input data
measured.dimensions	The dimensions to which correspond amount/numeric variables
coef.outl	Determines the length of the "whiskers" plot. If it is equal to zero no outliers will be returned. Default is 1.5.
box.outliers	If TRUE the outliers will be computed at the selected "coef.outl" level
box.wdth	The width level is determined 0.15 times the square root of the size of the input data.
cor.method	The correlation coefficient method to compute: "pearson" (default), "kendall" or "spearman".
freq.select	One or more nominal variables to calculate their corresponding frequencies.

Table 2. Input of the descriptive statistics algorithm

Pre-processing of input

“*DescriptiveStats.OBeu*” package includes functions that automatically calculates the central tendency and spread measures, the boxplot, histogram and barplot visualization parameters and the correlation matrix of the input fiscal dataset.

The final returns are the parameters needed for forming summary tables of central tendency and dispersion measures and visualizing boxplot, histogram, barplot and correlation matrix of the input data.

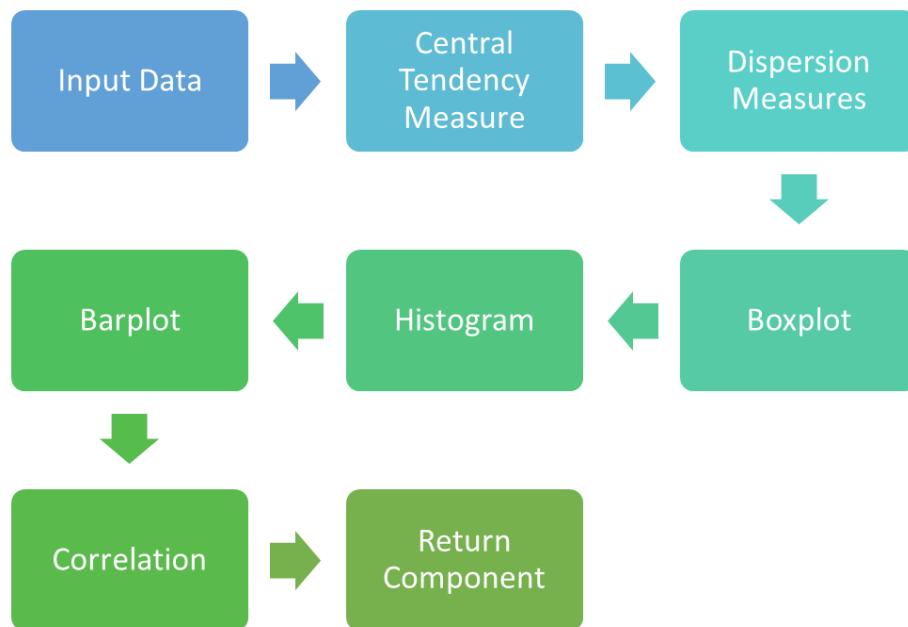


Figure 1 - Workflow of the Descriptive Analysis Module

Central Tendency Measures

Central Tendency Measures describe the central position of a distribution for a group of data. The basic measures are the mean and the median.

Mean

The mean (average) is the most popular measure of central tendency and can be used with discrete and continuous data. An important property of the mean is that it includes every value of the data set in the calculation process. For a set of n observations with x_1, x_2, \dots, x_n values, the sample mean is defined as:

$$\bar{x} = \frac{(x_1 + x_2 + \dots + x_n)}{n}$$

Median

The median is the value that separates the higher half from the lower half of the data that has been ranked in order of magnitude. In other words it is the middle value of the data. In contrast with mean, this measure is less affected by outliers and skewed data.

Dispersion Measures (Measures of Spread)

Dispersion measures describe how similar or varied the data is. The range, quartiles and the interquartile range, variance and standard deviation are measures of spread.

Range

The range is defined as the difference between the largest and smallest values. For a set of n observations with x_1, x_2, \dots, x_n values, the range is:

$$\text{range} = x_{\max} - x_{\min}$$

Quartiles and Interquartile range (IQR)

Quantiles are the values that divide the data, which should be ordered, into four equal parts. These values are called the first, second, and third quartiles and they are denoted by Q1, Q2, and Q3, respectively.

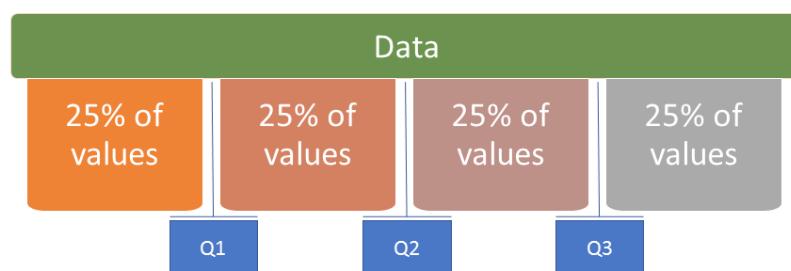


Figure 2: An illustration of Quantiles

The first quantile corresponds to the 25% of the data that lie below the Q1 value and the rest 75% lie above Q1. The second quantile corresponds to the 50% of the data that lie below the Q2 value and the rest 50% lie above Q2, in other words Q2 is the median value of the data. The third quantile corresponds to the 75% of the data that lie below the Q3 value and the rest 25% lie above Q3.

Interquartile range is a measure of variability that is not affected by outliers and it is defined as the difference between the third (Q3) and first (Q1) quartiles, and describes the middle 50% of ordered values.

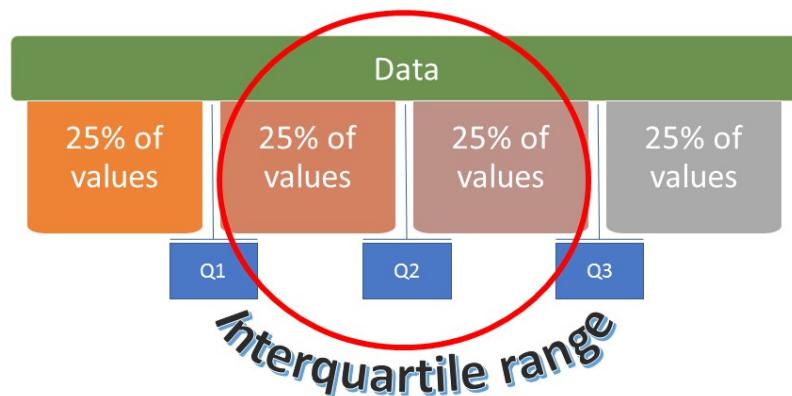


Figure 3: An illustration of Interquartile range

These quartiles can be clearly seen on a box plot of the data and also can be used to identify outliers (explained below in the boxplot section).

Variance

Variance known as second central moment of a distribution, is used to measure how far the data are spread, taking into account each individual value of the dataset. We use sample variance which is denoted by s^2 and concerning a data set of n observations of x_1, x_2, \dots, x_n values is defined as:

$$s^2 = \frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2$$

Standard Deviation

Standard deviation is defined as the square root of its variance. Unlike the variance, it is expressed in the same units as the data. A low standard deviation indicates that the data points tend to be close to the mean (expected value) of the set, while a high standard deviation indicates that the data points are spread out over a wider range of values than expected.

We use the corrected sample standard deviation, denoted by s and concerning a data set of n observations of x_1, x_2, \dots, x_n values is defined as:

$$s = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2}$$

Skewness

Skewness is the third central moment and is a measure of the asymmetry (describes the shape) of the probability distribution of a real-valued random variable about its mean. The skewness is shows (not directly) the relationship between the mean and median. In a distribution with negative skew the mean is smaller than the median and in a positive skew the mean is larger than the median.

When the skew is negative, the left tail is longer of a distribution making the mass of the distribution concentrate on the right of the figure. Having a positive skew, the right tail is longer of a distribution making the mass of the distribution concentrate on the left of the figure.

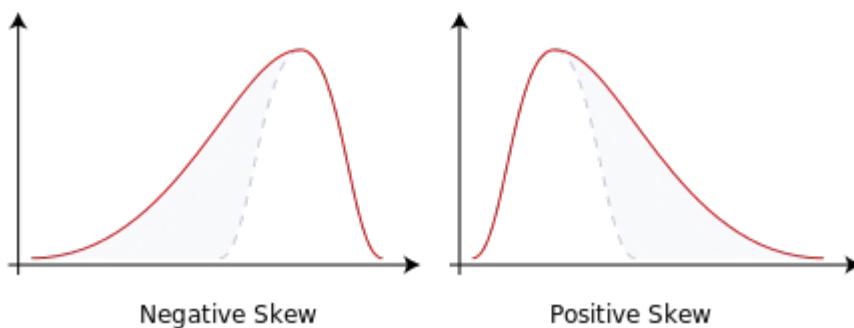


Figure 4: An illustration of Skewness

For a data set of n observations of x_1, x_2, \dots, x_n values and the sample mean, the sample skewness is defined as:

$$\text{skewness} = \frac{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^3}{\sqrt{\left(\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2\right)^3}}$$

Kurtosis

Kurtosis is a measure to describe the distribution, or skewness, of observed data around the mean, also referred to as the volatility of volatility and generally describes trends in charts.

Kurtosis can be identified in a histogram chart as heavy-tailed or light-tailed distributed data relative to a normal distribution. Datasets with high kurtosis value tend to have heavy tails, or outliers. In contrast to datasets with low kurtosis value that tend to have light tails and lack of outliers.

The kurtosis of data that are normally distributed are close to 3. Distributions with kurtosis less than 3 are called platykurtic and means the data probably have fewer and less extreme outliers than does the normal distribution. Distributions with kurtosis greater than 3 are said to be leptokurtic.

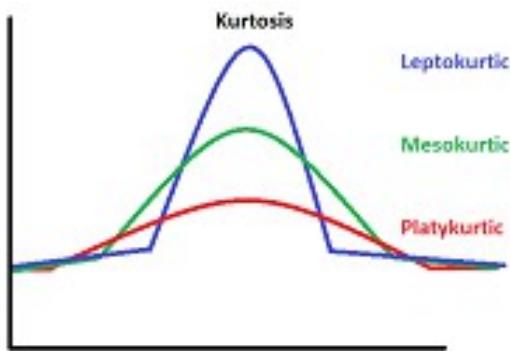


Figure 5: An illustration of types of kurtosis

We used the adjusted version of Pearson's kurtosis, the excess kurtosis, which is the kurtosis minus 3, in order to provide the comparison to the normal distribution. For a data set of n observations of x_1, x_2, \dots, x_n values and the sample mean, the sample excess kurtosis is defined as:

$$\text{kurtosis} = \frac{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^4}{\left(\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2\right)^2} - 3$$

The histogram is an effective graphical technique for showing both the skewness and kurtosis of data set.

Boxplot

The boxplot is the visualization that depicts groups of numerical data through their quartiles and have lines extending vertically from the boxes (whiskers) indicating variability outside the upper and lower quartiles. In this visualization outliers (extreme values) can be seen as individual points. The degree of spread and skewness in the data is shown by the spacings between the different parts of the box indicate.

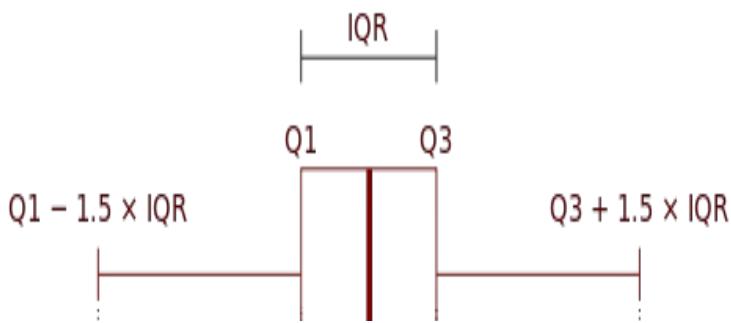


Figure 6: An illustration of Boxplot and a probability density function of a Normal Population

Box plots show variation in samples of a statistical population without making any assumptions of the underlying statistical distribution.

The interquartile range is used to find outliers in data. Outliers are observations that fall below $Q1 - 1.5 \times IQR$ or above $Q3 + 1.5 \times IQR$. The coefficient level 1.5 defines what is considered as outlier and can be defined by the user. The limits defined by this formula are the highest and lowest value that are drawn as bar of the whiskers, and outside these limits are the outliers as individual points.

Histogram

The histogram is another way to represent the distribution (shape) of numerical data. To construct a histogram the data should be divided into equal, non-overlapping and adjacent intervals called bins and then count how many values fall into each interval.

In other words the histogram is a function m_i that counts the number of observations that fall into each of the bins. Let n be the total number of observations and k be the total number of bins, the histogram m_i meets the following conditions:

$$n = \sum_{i=1}^k m_i.$$

There is no best number of bins, and different bin sizes can reveal different features of the data but usually equal width bins are used. Let k the number of bins and h the bin size of a continuous variable x of length n , some ways to determine the number of bins are the following:

Through the ceiling function of:

$$k = \left\lceil \frac{\max x - \min x}{h} \right\rceil$$

Sturges' formula:

Sturges' formula is derived from a binomial distribution and assumes an approximately normal distribution.

$$k = \lceil \log_2 n \rceil + 1$$

Scott's normal reference rule

$$h = \frac{3.5\hat{\sigma}}{n^{1/3}}$$

where $\hat{\sigma}$ is the sample standard deviation. Scott's normal reference rule minimizes the integrated mean squared error of the density estimate which is a good approach for random samples of normally distributed data.

Freedman–Diaconis' rule:

$$h = 2 \frac{\text{IQR}(x)}{n^{1/3}}$$

Where IQR is the interquartile range. This rule is less sensitive to outliers in data.

Bar graph

A bar chart or bar graph is a representation of categorical data with rectangular bars. Its lengths are proportional to the values they represent. Each bar in the visualization contains the frequency or count of the occurrences of values within a particular group or interval, and in this way, this visualization shows the distribution of values in the sample.

Correlation

Correlation is a statistical technique that shows the relationship between two random variables. Correlations provide useful indications of a predictive relationship between these variables that can be exploited in practice.

Pearson's correlation coefficient (rank correlation coefficient)

Pearson's correlation coefficient measures the linear dependence (correlation) between two variables X and Y. It accepts values between -1 and +1, where values near 1 mean that the two variables are positive linear correlated, -1 are negative linear correlated and 0 are non linear correlated.

We use the sample Pearson's correlation coefficient (denoted with r). For two variables of $\{x_1, x_2, \dots, x_n\}$ and $\{y_1, y_2, \dots, y_n\}$ values and length n, the coefficient is defined as:

$$r = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}}$$

where the sample means of variables x and y respectively.

Spearman's rank correlation coefficient

Spearman's rank correlation coefficient measures the (statistical) dependence between the ranking of two variables. It is used to assess if the relationship between two variables can be described using a monotonic function. The Spearman's coefficient is equal to Pearson's coefficient, but applied to the rank variables. A perfect Spearman correlation of +1 or -1 occurs when each of the variables is a perfect monotone function of the other.

For a sample of size n, the n raw scores X_i, Y_i are converted to ranks is computed from:

$$r_s = \rho_{rg_X, rg_Y} = \frac{\text{cov}(rg_X, rg_Y)}{\sigma_{rg_X} \sigma_{rg_Y}}$$

where

ρ is the Pearson correlation coefficient, but applied to the rank variables,

$\text{cov}(rg_X, rg_Y)$ is the covariance of the rank variables

σ_{rg_X} and σ_{rg_Y} are the standard deviations of the rank variables

Kendall's Tau b

Kendall's tau b statistic measures the ordinal association between two measured quantities and makes the appropriate adjustments for ties and similarly with Pearson's Correlation Coefficient it's range is from -1 (total negative association) to +1 (total positive association) and zero correlation means that there is no association.

The Kendall Tau-b coefficient is defined as:

$$\tau_B = \frac{n_c - n_d}{\sqrt{(n_0 - n_1)(n_0 - n_2)}}$$

where

$$n_0 = n(n - 1)/2$$

$$n_1 = \sum_i t_i(t_i - 1)/2$$

$$n_2 = \sum_j u_j(u_j - 1)/2$$

n_c the number of concordant pairs,

n_d the number of discordant pairs,

t_i the number of tied values in the i -th group of ties for the first quantity and

u_j the number of tied values in the j -th group of ties for the second quantity.

Output structure

The output of this process is a list in json format divided into four components of parameters and results with the first subcomponents (for further details about the package see `ds.analysis` function⁹):

descriptives	Min	Quantiles
	Max	Variance
	Range	StandardDeviation
	Mean	Skewness
	Median	Kurtosis
boxplot	lo.whisker	box.width
	lo.hinge	lo.out

⁹ <https://github.com/okgreece/DescriptiveStats.OBeu/blob/master/R/ds.analysis.R>

	median	up.out
	up.hinge	n
	up.whisker	
histogram	cuts	mean
	counts	median
	normal.curve	
frequencies	frequencies	
	relative.frequencies	
correlation	cor.matrix	

Table 3: The main return components of descriptive statistics

The component *descriptives* includes the information about the central tendency and dispersion measures as well as the third and fourth central moments of the input data.

In *boxplot*, *histogram*, *frequencies* and *correlation* components there are all the details concerning the needed parameters to visualize their corresponding parameters. These components can be used for Comparative Analysis.

Descriptives

Output	Description
Min	The minimum observed value of the input data
Max	The maximum observed value of the input data
Range	The range, defined as the difference of the maximum and the minimum value
Mean	The average value of the input data
Median	The median value of the input data
Quantiles	The 25%, 75% percentiles
Variance	The variance of the input data
StandardDeviation	The standard deviation of the input data
Skewness	The Skewness of the input data
Kurtosis	The Kurtosis of the input data

Table 4: Interpretation of variables in descriptive statistics

Boxplot

Output	Description
lo.whisker	Lower horizontal line out of the box
lo.hinge	Lower horizontal line of the box
median	Horizontal line in the box
up.hinge	Upper horizontal line of the box
up.whisker	Upper horizontal line out of the box
box.width	The box width of each variable
lo.out	Lower outliers
up.out	Upper outliers
n	The number of non-NA observations

Table 5: Interpretation of variables in boxplot

Histogram

Output	Description
cuts	The boundaries of the histogram classes
counts	The frequency of each histogram class
normal.curve	The normal curve
mean	The average value of the input vector
median	The median value of the input data

Table 6: Interpretation of variables in histogram

Frequencies

Output	Description
Variable name	The name of the calculated variable
frequencies	The frequency value
“_row”	Name of the categories of the variable
relative.frequencies	Relative frequency values

Table 7: Interpretation of variables in frequencies

Correlation

Output	Description

Variable name	The name of the calculated variable
Correlation value	The correlation value
“_row”	The corresponding correlation variable

Table 8: Interpretation of variables in correlation

3.1.3 Sample case

This section is a sample case of Descriptive Analysis using the “*DescriptiveStats.OBeu*” package providing custom visualizations for Municipality of Athens. It describes the descriptive measures along with their corresponding visualizations. This package can be used for comparison visualizations matrices that included in Comparison Analysis Section.

Data

We selected the data set of expenditure budget phase amounts for Municipalities of Athens from 2004 to 2015. This dataset includes the administrative units of the municipality, the year of recorded expenditure activity (2004-2015), the description of the expenditure amounts, the expenditure budget phase amount of draft ,revised, reserved,executed and approved money of Athens for the selected time window.

year	administrative_unit	description	draft	revised	reserved	approved	executed
2004	ΜΙΣΘΟΔΟΣΙΑ ΑΝΤΑΠΟΔΟΤΙΚΩΝ ΥΠΗΡΕΣΙΩΝ	Προμήθεια λοιπών ειδών ένδυσης & υπόδησης & ειδ...	18000	18000	18000	8977	5000.0
2004	Δ/ΝΣΗ ΔΗΜΟΣΙΩΝ ΣΧΕΣΕΩΝ & ΔΙΕΘΝΟΥΣ ΣΥΝΕΡΓΑΣΙΑΣ	Αμοιβές μεταφραστών, στενοδακτυλογράφων & διερμ...	50000	50000	34000	9601	16223.0
2004	Δ/ΝΣΗ ΔΗΜΟΣΙΩΝ ΣΧΕΣΕΩΝ & ΔΙΕΘΝΟΥΣ ΣΥΝΕΡΓΑΣΙΑΣ	Οδοιπορικά έξοδα & ημερ. αποζημ. μετακιν. Αιρετών ...	350000	350000	140200	41075	41477.0
2004	Δ/ΝΣΗ ΔΗΜΟΣΙΩΝ ΣΧΕΣΕΩΝ & ΔΙΕΘΝΟΥΣ ΣΥΝΕΡΓΑΣΙΑΣ	Οδοιπορικά έξοδα & ημερ. αποζημ. μετακιν. υπαλλήλ...	160000	240000	139653	62286	50000.0
2004	Δ/ΝΣΗ ΔΗΜΟΣΙΩΝ ΣΧΕΣΕΩΝ & ΔΙΕΘΝΟΥΣ ΣΥΝΕΡΓΑΣΙΑΣ	Οδοιπορικά έξοδα & ημερ. αποζημ. μετακιν. που δεν...	150000	210000	87800	74275	75393.0
2004	Δ/ΝΣΗ ΔΗΜΟΣΙΩΝ ΣΧΕΣΕΩΝ & ΔΙΕΘΝΟΥΣ ΣΥΝΕΡΓΑΣΙΑΣ	Λοιπές επικοινωνίες	40000	40000	19500	2032	2032.0
2004	Δ/ΝΣΗ ΔΗΜΟΣΙΩΝ ΣΧΕΣΕΩΝ & ΔΙΕΘΝΟΥΣ ΣΥΝΕΡΓΑΣΙΑΣ	Δαπάνες υποδοχής & φιλοξενίας	300000	300000	202000	44484	42554.9
2004	Δ/ΝΣΗ ΔΗΜΟΣΙΩΝ ΣΧΕΣΕΩΝ & ΔΙΕΘΝΟΥΣ ΣΥΝΕΡΓΑΣΙΑΣ	Δαπάνες συνεδρίων, δεξιώσεων & άλλων εκδηλώσεων	500000	500000	365420	90351	106342.1
2004	Δ/ΝΣΗ ΔΗΜΟΣΙΩΝ ΣΧΕΣΕΩΝ & ΔΙΕΘΝΟΥΣ ΣΥΝΕΡΓΑΣΙΑΣ	Λοιπές δαπάνες δημοσίων σχέσεων	400000	410000	207300	58905	40000.0
2004	Δ/ΝΣΗ ΔΗΜΟΣΙΩΝ ΣΧΕΣΕΩΝ & ΔΙΕΘΝΟΥΣ ΣΥΝΕΡΓΑΣΙΑΣ	Εκτυπώσεις εκδόσεις & βιβλιοδετήσεις	150000	150000	84000	47790	60000.0
2004	Δ/ΝΣΗ ΔΗΜΟΣΙΩΝ ΣΧΕΣΕΩΝ & ΔΙΕΘΝΟΥΣ ΣΥΝΕΡΓΑΣΙΑΣ	Λοιπές δαπάνες γενικής φύσεως	100000	100000	57000	20178	20000.0
2004	Δ/ΝΣΗ ΔΗΜΟΣΙΩΝ ΣΧΕΣΕΩΝ & ΔΙΕΘΝΟΥΣ ΣΥΝΕΡΓΑΣΙΑΣ	Μεταφορές προσώπων	200000	200000	64500	15050	17750.0
2004	Δ/ΝΣΗ ΔΗΜΟΣΙΩΝ ΣΧΕΣΕΩΝ & ΔΙΕΘΝΟΥΣ ΣΥΝΕΡΓΑΣΙΑΣ	Λοιπές Προμήθειες βιβλίων & εντύπων	100000	100000	12000	1500	3000.0
2004	Δ/ΝΣΗ ΔΗΜΟΣΙΩΝ ΣΧΕΣΕΩΝ & ΔΙΕΘΝΟΥΣ ΣΥΝΕΡΓΑΣΙΑΣ	Εισφορές στο EUROCITIES	15000	15000	14379	14379	14379.0
2004	Δ/ΝΣΗ ΑΠΟΧΕΤΕΥΣΗΣ	Έξοδα κινήσεως υπαλλήλων εντός έδρας	55000	55000	28000	27340	30000.0
2004	Δ/ΝΣΗ ΑΠΟΧΕΤΕΥΣΗΣ	Κατασκευή αγγυών δικτύου ακαθάρτων	3096000	3096000	3095399	1300474	1538336.2
2004	Δ/ΝΣΗ ΑΠΟΧΕΤΕΥΣΗΣ	Κατασκευή αγγυών σύνθετης οικίων με δίκτυο ακαθά...	1056000	1056000	277961	294816	335216.1
2004	Δ/ΝΣΗ ΑΠΟΧΕΤΕΥΣΗΣ	Λοιπά έργα αποχέτευσης	65000	65000	10430	10422	10421.8
2004	Δ/ΝΣΗ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΕΛΕΓΧΟΥ Κ ΕΠΙΘΕΩΡΗΣΗΣ	Έξοδα κινήσεως υπαλλήλων εντός έδρας	26000	30000	0	18136	70000.0
2004	Δ/ΝΣΗ ΑΡΧΙΤΕΚΤΟΝΙΚΟΥ	Έξοδα κινήσεως υπαλλήλων εντός έδρας	120000	150000	120000	67041	178053.8
2004	Δ/ΝΣΗ ΑΡΧΙΤΕΚΤΟΝΙΚΟΥ	Εκτυπώσεις εκδόσεις & βιβλιοδετήσεις	10000	10000	3460	3339	3000.0

Figure 7.: Fiscal dataset of Municipality of Athens

Descriptive measures

We calculated the main descriptive measures for each variable of the budget phase (draft, revised, reserved, approved, executed) in order to get a clearer view of the expenditure management in this municipality from 2004 to 2015.

	mean	sd	median	min	max	range	skew	kurtosis	var	Q1	Q3
draft	1551908.8	5730706.3	130000.0	480.0	120000000.0	119999520.0	10.3	155.7	32840994822550.5	30000.0	644400.0
revised	1594889.5	5648765.1	146000.0	480.0	127273717.0	127273237.0	9.6	140.7	31908546923488.8	30000.0	700000.0
reserved	1253866.9	4332557.0	78141.0	30.0	61200000.0	61199970.0	7.2	68.0	18771050394602.1	14690.0	450000.0
approved	1115545.5	4075113.3	42514.0	30.0	55990810.7	55990780.7	7.3	68.7	16606548658410.9	7490.7	306505.6
executed	1146727.4	4159006.7	43653.8	20.0	55990810.7	55990790.7	7.2	65.7	17297336500165.1	7654.3	317055.0

Figure 8.: Summary table of basic descriptive measures of Athens in 2004-2015 period

The above Table presents the basic descriptive statistics of the amounts of budget phases in Athens from 2004 to 2015. In this time window, an average draft amount of 1,551,908.8€ was revised to 1,594,889.5€, reserved to 1,253,866.9€, approved to 1,115,545.5€ and finally executed an amount of 1146727.4€.

The first quartile(Q1) value is 7,654.3€ which means that the 25% of the executed expenditure amounts lie below and the rest 75% lie above that value.

The third quartile(Q3) value is 317,055€ which means that the 75% of the executed expenditure amounts lie below and the rest 25% lie above that value. The middle value of these amounts 43653.8€ making the 50% of the expenditures lie above and below that value.

The skew values show that all these budget phases are positively skewed (see boxplot and histogram below) forming the right tail longer and making the mass of the distribution is concentrated on smaller amounts and the large values of kurtosis making the distributions leptokurtic, providing an indication of many and large outlier values.

Boxplot

The boxplots visualization below show the executed expenditure amounts of Athens for each year from 2004 to 2015 through their quartiles. Using the standard coefficient level (1.5) we can see some outliers (extreme values) which are depicted as individual points. The degree of spread and skewness in the data verify the table of the calculated descriptive measures in the previous section. The boxes height (Interquartile range -IQR) varies over the years showing that the municipality changed the executed expenditure amounts of some services to the citizens.

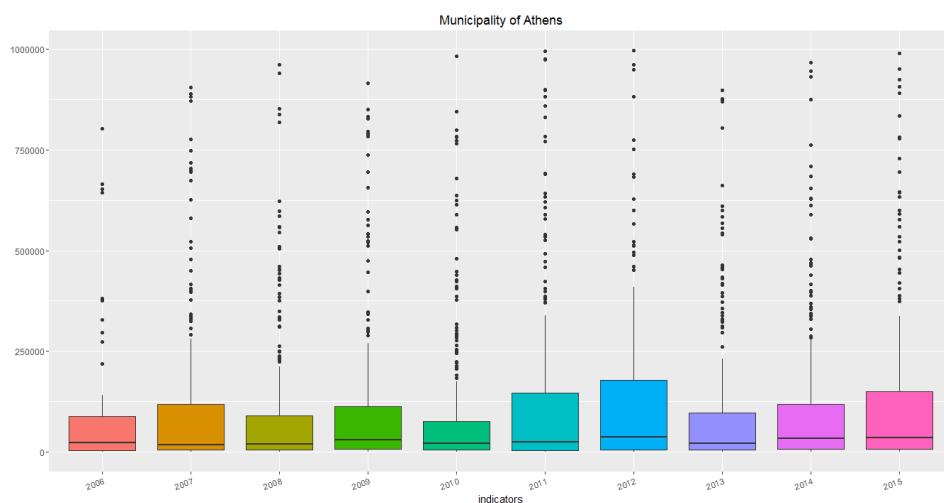


Figure 9.: Snapshot of Executed Expenditure Amounts of Athens in 2004-2015 period

Histogram

Figure 10 shows the distribution (shape) (histogram) visualizations of budget phase amounts, using Scott's normal reference rule. The skewness and kurtosis values from the descriptive measures table, explained in previous section, verify the nature of the following histograms where the mass of the distribution is concentrated on smaller amounts and the large values of kurtosis making the distributions leptokurtic. These distributions have heavy tails indicating the existence of many and large outlier values (which were also shown in boxplots).

draft	revised
-------	---------

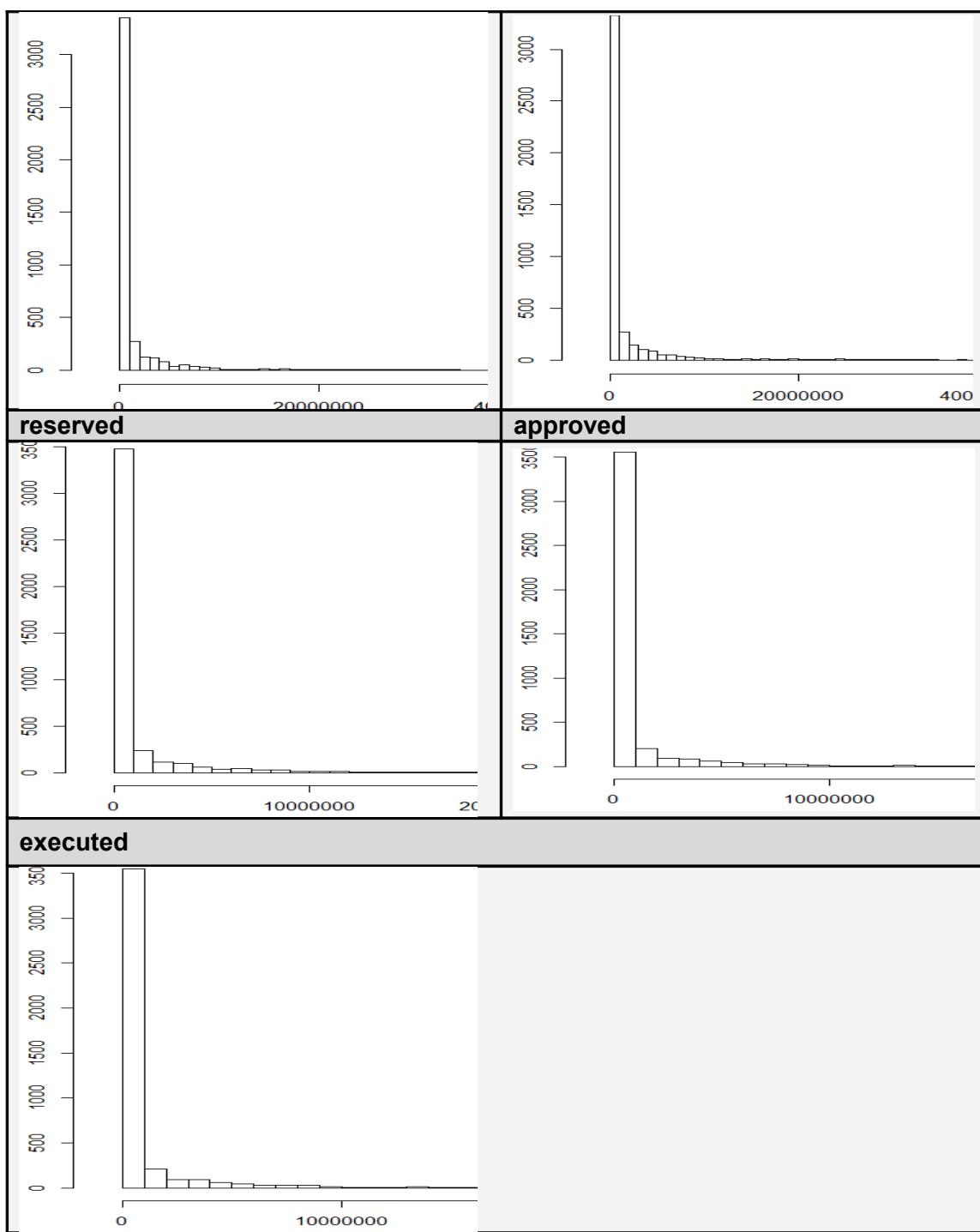


Figure 10. Histogram representation

Frequencies

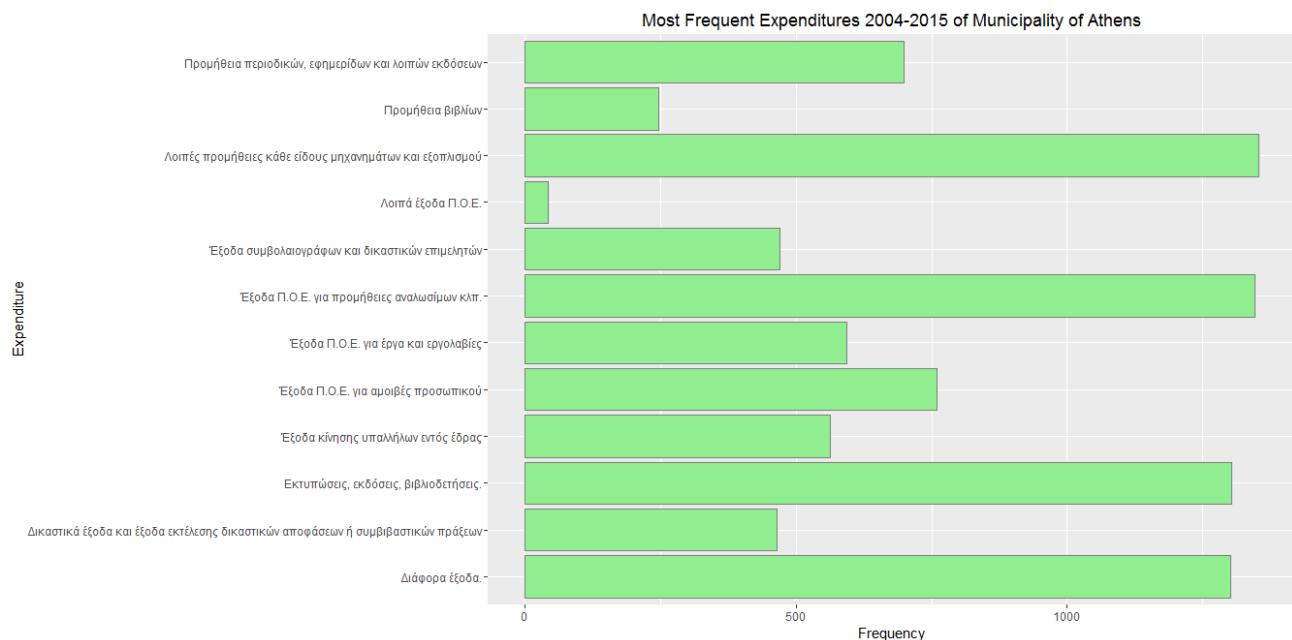


Figure 11. Frequency representation

The Figure above shows the 12 most frequent expenditure amounts that were executed by the administrative units of municipality of Athens from 2004 to 2015.

Correlation

The Figure below shows the correlation matrix among expenditure budget phase amounts of Athens. We used Pearson's correlation coefficient that assess the existence of linear relationship among budget phase amounts. All these budget phase amounts are positively correlated (correlation near 0.8) and the lower correlation is observed at draft and executed amounts and draft and approved, which means that the draft amounts are less linearly dependent with executed and approved amounts than the other budget phases.

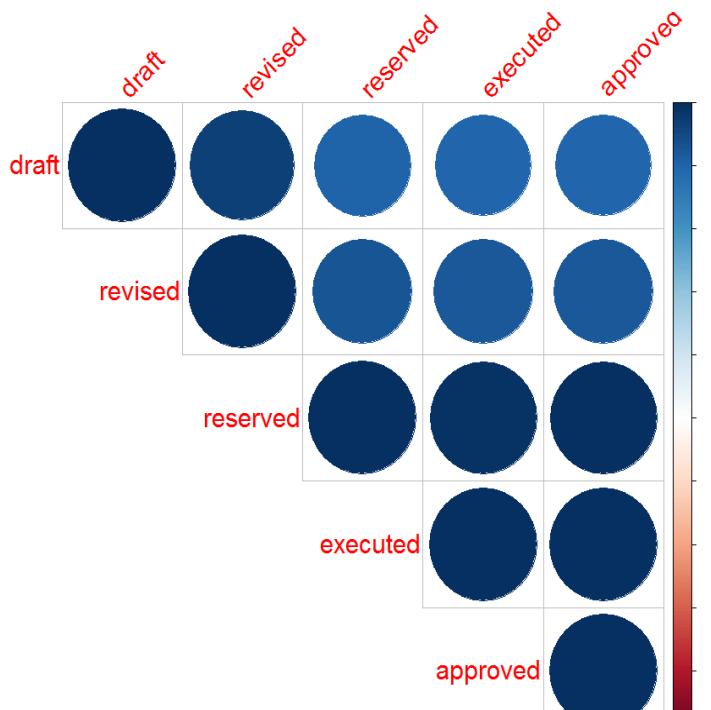


Figure 12. Correlation representation

Based on the nature of the selected data and the method this sample case is considered also as part of comparative analysis. For example in the summary table of basic descriptive measures of Athens in 2004-2015 period could be considered as comparative table of the budget phases amounts as there are the basic descriptive measures of each phase.

3.2 Time series analysis, predictions

Kleanthis

3.2.1 General description

Time series data are values sampled as an ordered sequence of equally spaced time intervals. Time series analysis involves methods and techniques that take into account the internal structure of the data (such as autocorrelation, trend or seasonal variance), in order to extract meaningful characteristics and fit a model to predict future behavior of such data.

Budget data of municipalities across Europe do have that structure in different levels. In order to meet user needs and eventually the tasks described in detail in Deliverable 2.3- “*Requirements for Statistical Analytics and Data Mining Techniques*” and summarized in the following table, we developed “*TimeSeries.OBeu*” package, which was built in R Software environment and it is available in github¹⁰:

Need	Description	Discussion	Task No.
N02	Version tracking of budgets	This data mining and analytics need refers to a comparative analysis of budget lines along the different budget phases and can be extended to find and measure trends in doing so.	T02
N05	Extrapolations on data	This data mining and analytics need extends (N02) in two aspects: first to perform predictions for future budgets and second to incorporate budget data from different fiscal periods in the analysis.	T04
N07	Temporal trend of the difference between planned and actual spending	This data mining and analytics need is related to (N02) and extends it with a temporal dimension involving budget data from several years and incorporating corresponding spending data. Another aspect is to investigate and analyze the reasons for the detected trends.	T06
N13	Analyze larger trends over time and in different funding areas	This need matches with (N02) and (N07) and extends it to a general trend analysis on the temporal dimension in budget and spending data.	T10
N15	Pay special focus on analyzing the spending and management of EU budget funds	For the analysis of EU budget funds we will use several of the already mentioned methods: Time series analysis (T02), comparative analysis (T09), rule/pattern mining (T11), and outlier detection (T12). The focus on analyzing EU budget funds is formulated as requirement (R04).	T02
N17	Consider fiscal indicators like error,	After calculating these fiscal indicator we will apply statistics with a focus on trends.	T16

10 <https://github.com/okgreece/TimeSeries.OBeu>

	performance and absorption rates	
--	----------------------------------	--

Table 9. List of needs covered in Time-Series Analysis

This package includes functions for Time Series Analysis techniques that are used in OpenBudgets.eu (OBEU) fiscal datasets. `TimeSeries.OBeu` is based on `forecast`¹¹, `locfit`¹², `jsonlite`¹³, `trend`¹⁴, `tseries`¹⁵ libraries. Local regressions and models from `arima` family have been selected to decompose, fit and predict the input fiscal datasets.

3.2.2 Input & output

User input

The user should define the “*time*” and “*amount*” parameters to form the time series data. The “*prediction_steps*” parameter should be defined to predict the future steps, the default value is set to 1 step forward. The user can also interact with the selection of the ARIMA model’s order to fit the data and specify the “*order*” parameter. The default order of the model, is fixed to fit the best model through some conditions and diagnostic tests.

The following table summarizes the input parameters:

Input	Description
<code>json_data</code>	The json string, URL or file from Open Spending API
<code>time</code>	The time label of the json time series data
<code>amount</code>	The amount label of the json time series data
<code>order</code>	An integer vector of length 3 specifying the order of the Arima model (optional)
<code>prediction_steps</code>	The future steps forward to predict

Table 10. Table of input parameters for time series analysis

Pre-processing of input

“`TimeSeries.OBeu`” package includes functions that automatically analyze the input univariate time series data. A set of tests are implemented in the input time series data in

11 <https://cran.r-project.org/web/packages/forecast/>

12 <https://cran.r-project.org/web/packages/locfit/>

13 <https://cran.r-project.org/web/packages/jsonlite/>

14 <https://cran.r-project.org/web/packages/trend/>

15 <https://cran.r-project.org/web/packages/tseries/>

order to assess the stationarity for further analysis. Autocorrelation and partial autocorrelation functions (ACF and PACF respectively), Phillips Perron, Augmented Dickey Fuller (ADF), Kwiatkowski Phillips Schmidt Shin (KPSS), Mann Kendall for Monotonic Trend and Cox Stuart trend tests are used to assess the stochastic or deterministic trend of the input time series data (see “`ts.stationary.test`” function for further details).

Depending the nature of the time series data and the stationary tests there are four branches of analysis that handles different types of time series data structures and return the most appropriate forecasts for:

- Short non seasonal time series
- Short seasonal time series
- Long non seasonal time series and
- Long seasonal time series

The final returns are the parameters needed for visualizing the time series data with the specified predictions as long as the decomposition components and some comparison measure matrices or measure plots parameters (see section for Comparison Analysis).

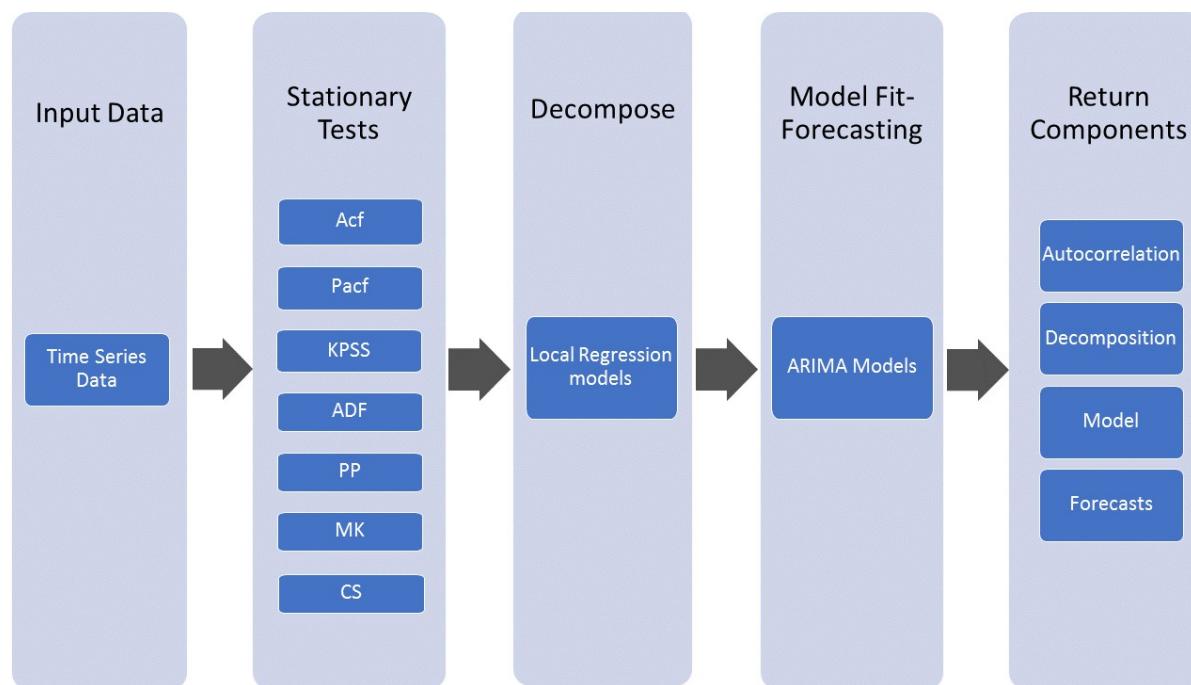


Figure 13.- Workflow of Time Series Analysis

To decompose time series data to trend, seasonal and irregular components, local regression models are used. Local regression is a non-parametric regression that merge

multiple regression models in a k-nearest-neighbor-based meta-model. Which is used to model a relation between a predictor variable and response variable.

The the model is defined as :

Where \hat{y}_t is the regression function which relates the t -th measurement of the response y with the corresponding measurement of the vector of p predictors x_t and ϵ_t is a random error.

Local regression assumes that near x_0 , the value of a function in some specified parametric class can approximate the regression function $g(x)$, locally. By fitting a regression surface to the data points within a chosen neighborhood of the point x_0 , we can obtain such a local approximation.

Stationary tests

Autocorrelation function (ACF)

Autocorrelation function is used to detect seasonality, repeating patterns or missing of fundamental frequency. Some forms of processes with autocorrelation are the unit root processes, trend stationary processes, autoregressive processes, and moving average processes. Let X be a stochastic process, t be any point in time, μ_t be the mean of this process and σ_t^2 the variance at time t . Then the definition of the autocorrelation between times s and t is:

$$R(s, t) = \frac{E[(X_t - \mu_t)(X_s - \mu_s)]}{\sigma_t \sigma_s}$$

where " E " is the expected value operator. This equation is not well defined as the means may not exist, and the variance may be zero or infinite. If we can calculate function R , its value must be between -1 and 1 , with 1 indicating perfect correlation and -1 indicating perfect anti-correlation.

If X_t is a wide-sense stationary process the mean and the variance are time-independent, so the autocorrelation depends only on the lag between t and s : the position in time has not any influence in correlation, correlation depends only on the time-distance between the pair of values. We can express the autocorrelation as a function of the time-lag, and that this would be an even function of the lag $\tau=s-t$.

$$R(\tau) = \frac{E[(X_t - \mu)(X_{t-\tau} - \mu)]}{\sigma^2}$$

Partial autocorrelation function (PACF)

Partial autocorrelation function gives the partial correlation (a conditional correlation) of a time series with its own lagged values, controlling for the values of the time series at all shorter lags. This function aims at identifying the extent of the lag in an autoregressive model, hence, it plays an important role in data analyses. The determination of the appropriate lags p in an AR(p) model or in an extended ARIMA (p,d,q) model, could be determined by plotting the partial autocorrelation functions, for this reason, the function was introduced as part of the Box-Jenkins approach to time series modeling.

Given a time series z_t , the partial autocorrelation of lag k , denoted $a(k)$, is the autocorrelation between z_t and z_{t+k} with the linear dependence of z_t on z_{t+1} through z_{t+k-1} removed; equivalently, it is the autocorrelation between z_t and z_{t+k} that is not accounted for by lags 1 to $k - 1$, inclusive.

$$a(1) = \text{Cor}(z_{t+1}, z_t),$$

$$a(k) = \text{Cor}(z_{t+k} - P_{t,k}(z_{t+k}), z_t - P_{t,k}(z_t)) \text{ for } k \geq 2$$

Where denotes the projection of x onto the space spanned

To identify the order of an autoregressive model, we use partial autocorrelation plots.

Kwiatkowski-Phillips-Schmidt-Shin test (KPSS)

Kwiatkowski-Phillips-Schmidt-Shin test is used for testing a null hypothesis that an observable time series is stationary around a deterministic trend (i.e. trend-stationary) against the alternative of a unit root.

There are three-component representations of the observed time series X_t the sum of a deterministic time trend, a random walk and a stationary residual:

$$X_t = \beta t + (r_t + \alpha) + e_t$$

where

$r_t = r_{t-1} + u_t$ is a random walk, the initial value $r_0 = a$ serves as an intercept,
 t is the time index,
 u_t are independent identically distributed $(0,)$

The null and alternative hypothesis:

$H_0: X_t$ is trend (or level) stationary or $= 0$

$H_1: X_t$ is a unit root process

In contrast, with the other tests, the hypothesis of a unit root is the alternative. If a unit root does not exist there is a proof of trend stationarity.

Augmented Dickey-Fuller test (ADF)

Augmented Dickey-Fuller test is used in time series to test the null hypothesis of a unit root. There are different versions of the test that can be used, and depending on the version we have the alternative hypothesis, in this package we use the stationarity or trend-stationarity. ADF test is a version of Dickey Fuller test, which is used for large and complicated sets of time series model.

ADF statistic is a negative number. The more negative it is, the stronger the rejection of the hypothesis that there is a unit root at some level of confidence.

The equation of the model is:

$$\Delta y_t = \alpha + \beta t + \gamma y_{t-1} + \delta_1 \Delta y_{t-1} + \dots + \delta_{p-1} \Delta y_{t-p+1} + \varepsilon_t$$

where α is a constant, β the coefficient on a time trend and p the lag order of the autoregressive process. Imposing the constraints $\alpha = 0$ and $\beta = 0$ corresponds to modelling a random walk and using the constraint $\beta = 0$ corresponds to modeling a random walk with a drift. By including lags of the order p the ADF formulation allows for higher-order autoregressive processes. This means that the lag length p has to be determined when applying the test.

Phillips-Perron test (PP)

Phillips-Perron test is a unit root test that is used to test the null hypothesis that the order of integration in a time series equals with 1. It is based on the Dickey-Fuller test. PP test deals with the problem that the process generating data for y_t might have a higher order of autocorrelation than is admitted in the test equation-making y_{t-1} endogenous. It also makes a non-parametric correction to the t-test statistic.

$$\Delta y_t = \rho_{t-1} + u_t$$

With respect to unspecified autocorrelation and heteroscedasticity in the disturbance process of the test equation, PP test is robust. Two advantages for PP test are: it's robust to general forms of heteroskedasticity in the error term u_t and that we do not have to specify a lag length for the test regression.

Cox Stuart test (CS)

The Cox- Stuart test is a non-parametric test. It is used to test the null hypothesis that no monotonic trend exists in the series against the alternative that the trend is monotonic. The alternative hypothesis provide three alternatives:

1. An upward or downward trend exists
2. A downward trend exists
3. An upward trend exists

The 3rd one is symmetric to the 2nd one, and has p-value twice smaller than the 1st.

H_0 : No monotonic trend exists in the series

H_1 : The series is characterized by a monotonic trend

Let x_i , $i=1,\dots,n$ be a series of data. We suppose that n is even (if it's not we remove the middle value of the data) and divide the data into two equal groups. So we have the pairs

$$\left(x_j, x_{j+\frac{n}{2}} \right), j = 1, \dots, \frac{n}{2}$$

The test statistics T equals the number of pairs in which

$$x_j < x_{j+\frac{n}{2}}$$

If the null hypothesis is true, then the statistics T is binomially distributed with parameters,

$$T \sim B\left(\frac{n}{2}, \frac{1}{2}\right)$$

[Reference: Properties of the Cox-Stuart Test for Trend in Application to Hydrological Series: The Simulation Study]

Mann-Kendall Test For Monotonic Trend (MK)

Mann-Kendall test estimates if a monotonic upward or downward trend of the variable of interest exist over time. A monotonic upward (downward) trend means that the variable increases (decreases) through time, but it's not necessary for the trend to be linear. We can replace the parametric linear regression analysis with MK test, which can be used to test if the slope of the estimated linear regression line is different from zero. MK test does not require the residuals from the fitted regression line be normally distributed, as in regression analysis. The MK test is a nonparametric (distribution-free) test.

Three assumptions hold in MK test are: The measurements (observations or data) obtained over time are independent (there is no correlation over time) and identically distributed, when no trend is present. The true conditions at sampling times are represented by the observations obtained over time. The sample collection, handling, and measurement methods provide unbiased and representative observations of the underlying populations over time.

H_0 : no monotonic trend

H_1 : monotonic trend is present

Model Fit- Forecasts

An autoregressive integrated moving average model, in time series analysis, is a generalization of autoregressive moving average model. ARIMA models are considered to be the most general class of models for forecasting a time series. These models are being used in order to understand the data or to predict future values of the series, by fitting those models into the time series. ARIMA models handles also data that show signs of non stationarity. The autoregressive means that the evolving variable of interest is regressed on its own lagged values. The moving average points out that the regression error consists of a linear combination of error terms. And the integrated means that the data have been replaced with the difference between their present and previous values. An ARIMA(p,d,q) process expresses this polynomial factorization property, parameters p, d, and q are non-negative integers, p is the number of time lags of the autoregressive model, d is the degree of differencing, and q is the order of the moving-average model. the equation of the model (forecasting equation) is:

$$\left(1 - \sum_{i=1}^p \varphi_i L^i\right) \left(1 - L\right)^d X_t = \left(1 + \sum_{i=1}^q \theta_i L^i\right) \varepsilon_t$$

$$(1 - \varphi_1 L - \dots - \varphi_p L^p) \quad (1 - L)^d X_t = (1 + \theta_1 L + \dots + \theta_q L^q) \varepsilon_t$$

↑ ↑ ↑
 $AR(p)$ d differences $MA(q)$

Where

L is the lag operator,

the φ_i are the parameters of the moving average part,

the ε_t are error terms,

and X_t is a time series of data where t is an integer index and the X_t are real numbers.

In “*TimeSeries.OBeu*” package there is an automated process that selects the appropriate model to fit the time series data after testing different models and meeting some specified conditions.

For example the forecasting equation of a first-order autoregressive model (ARIMA(1,0,0)) is:

$$\hat{X}_t = \mu + \varphi_1 X_{t-1}$$

which is X regressed to itself lagged by one period. If the mean of X is zero, then the constant term would not be included. For positive and less than one slope coefficient φ_1 , the model describes mean-reverting behavior in which next period's value should be predicted to be φ_1 times as far away from the mean as this period's value. For negative φ_1 , it predicts mean-reverting behavior with alternation of signs, i.e., it also predicts that X will be below the mean next period if it is above the mean this period.

There are models in ARMA family that handles the problem of autocorrelated errors of a random walk, such as a differenced first-order autoregressive model (ARIMA(1,1,0)), which is a first-order autoregressive model with one order of nonseasonal differencing and a constant term. This model is defined as:

$$\hat{X}_t = \mu + X_{t-1} + \varphi_1 (X_{t-1} + X_{t-2})$$

It's possible that the problem of autocorrelated errors of a random walk model can be fixed by regressing the first difference of Y on itself lagged by one period.

Another model that handles could be used as a strategy for correcting autocorrelated errors in a random walk model is the ARIMA(0,1,1) without constant which is a simple exponential smoothing model. Sometimes, random walk model does not perform as well as a moving average of past values when we have nonstationary time series, so it is better to use an average of the last few observations in order to filter out the noise and more accurately estimate the local mean.

The simple exponential smoothing model uses an exponentially weighted moving average of past values to achieve this effect. The prediction equation for the simple exponential smoothing model can be written in a form, known as "error correction" form, in which the previous forecast is adjusted in the direction of the error it made:

$$\hat{X}_t = X_{t-1} - \theta_1 e_{t-1}, \text{ with } \theta_1 = 1 - \alpha$$

Output structure

The output of this process is a list in json format divided into four components of parameters and results with the first subcomponents (for further details about the package see the function `ts.analysis`¹⁶) :

acf.param	acf.parameters
	pacf.parameters
	acf.residuals.parameters
	pacf.residuals.parameters
	stl.plot
	stl.general

16 <https://github.com/okgreece/TimeSeries.OBeu/blob/master/R/ts.analysis.R>

	residuals_fitted
	compare
forecasts	forecasts

Table 11.: The main return components of time series analysis

The component `acf.param` includes the information about the autocorrelation and partial autocorrelation function of the input data and the residuals after fitting a model. In decomposition component there are all the details concerning the decomposition of time series data. Subcomponent `stl.plot` has the values of the seasonal,trend and residuals components and `stl.general,residuals_fitted` and `compare` components are described in Section 3.4.

The `model.param` component also described in Section 3.4 and in `forecast` component there are the predicted values of the fitted ARIMA model along with their confidence intervals.

acf.param

acf.parameters	
Output	Description
acf	The estimated acf values of the input time series
acf.lag	The lags at which the acf is estimated
confidence.interval.up	The upper limit of the confidence interval
confidence.interval.low	The lower limit of the confidence interval

Table 12. Table of output variables for autocorrelation function

pacf.parameters	
Output	Description
pacf	The estimated pacf values of the input time series

pacf.lag	The lags at which the pacf is estimated
confidence.interval.up	The upper limit of the confidence interval
confidence.interval.low	The lower limit of the confidence interval

Table 13. Table of output variables for partial autocorrelation function

acf.residuals.parameters	
Output	Description
acf.res	The estimated acf values of the model residuals
acf.res.lag	The lags at which the acf is estimated
confidence.interval.up	The upper limit of the confidence interval
confidence.interval.low	The lower limit of the confidence interval

Table 14. Table of output variables for autocorrelation function for the residual model

pacf.residuals.parameters	
Output	Description
pacf.res	The estimated pacf values of the model residuals
pacf.res.lag	The lags at which the pacf is estimated
confidence.interval.up	The upper limit of the confidence interval
confidence.interval.low	The lower limit of the confidence interval

Table 15. Table of output variables for partial autocorrelation function for the residual model

Decomposition- stl.plot:

Output	Description
trend	The estimated trend component
trend.ci.up	The estimated up limit for trend component (for non seasonal time)

	series)
trend.ci.low	The estimated low limit for trend component (for non seasonal time series)
seasonal	The estimated seasonal component
remainder	The estimated remainder component
time	The time of the series was sampled

Table 16. Table of output variables in trend analysis

Forecasts:

Output	Description
data	The time series data values
data_year	The time/year that time series data were sampled
predict_values	The predicted values that defined by the prediction_steps parameter
predict_time	The time/years forward that defined by the prediction_steps parameter
up80	The upper limit of the 80% predicted confidence interval
low80	The lower limit of the 80% predicted confidence interval
up95	The upper limit of the 95% predicted confidence interval
low95	The lower limit of the 95% predicted confidence interval

Table 17. Table of output variables in forecast

3.2.3 Sample case

This section is a sample case of Time Series Analysis using the “*TimeSeries.OBeu*” package providing custom visualizations for Municipality of Athens. It describes the time series tasks under their corresponding conditions and example visualizations of the results. In this package there are comparison visualization matrices that included and described in Section 3.4.

Input Data

We selected to study time series data of revised expenditure budget phase for 2004 to 2015 of Municipality of Athens.

```
Time Series:
Start = 2004
End = 2015
Frequency = 1
[1] 741600221 642063819 665848153 745142370 885773956 999818946
[7] 1026427000 931404942 864990287 911073390 815403479 833076206
```

Figure 13.: Time Series Data of Revised Expenditure Time Series Data of Municipality of Athens

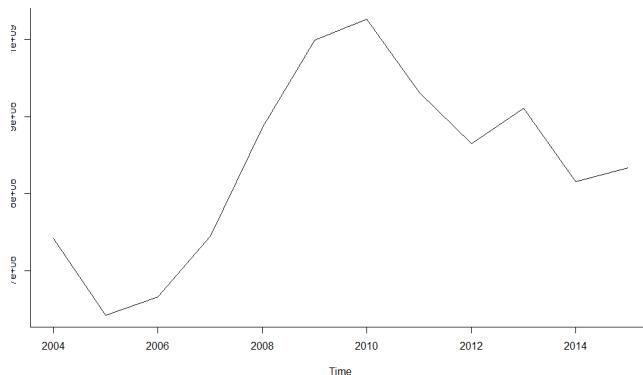


Figure 14.: Time Series of Revised Expenditure Time Series Data of Municipality of Athens

Autocorrelation and Partial autocorrelation

The corresponding autocorrelation and partial autocorrelation function visualizations of Revised time series data are shown below:

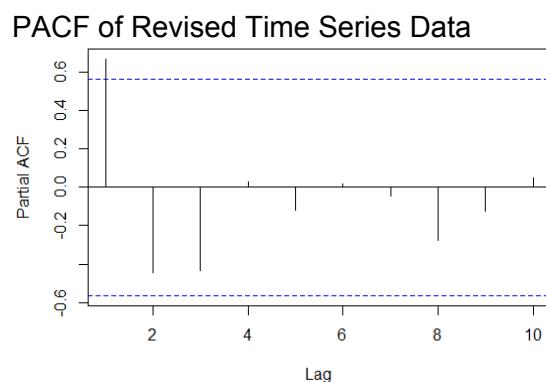
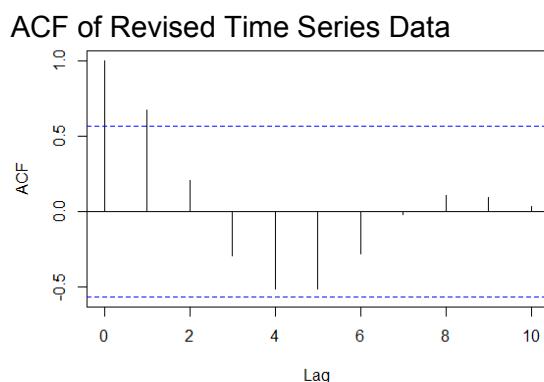


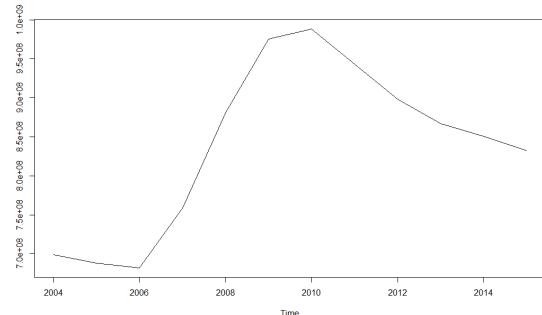
Figure 15.: An illustration of autocorrelation and partial autocorrelation

The ACF of the Revised time series data visualization slowly tailing off and its PACF cuts off after lag 1, which means that the time series data are not stationary and should be made stationary proposing through the geometric pattern an appropriate order of the model (in this case the 1rst order of AR). Nevertheless the algorithm will auto select the appropriate model after making the time series data stationary and will take into account the Akaike's Information Criterion values among the tested models.

Decomposition

The following visualizations show the decomposition components of the input time series data:

Trend



Remainder

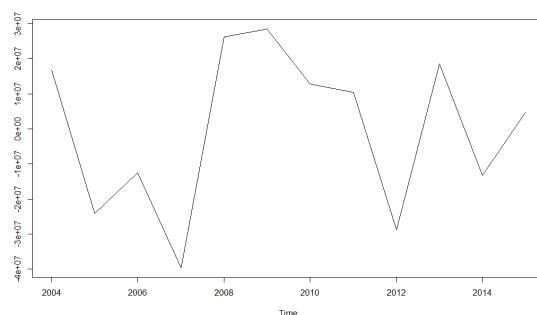


Figure 16.: Decomposition of Revised Budget Phase Time Series

There are trend and irregular components and the sampled frequency is 1 so there is no seasonal component in these data. There is an upward trend until 2010 and after that year the trend slowly decreases.

ARIMA Model Fit-Forecasts

The process of the identification of best fit ARIMA model and the autocorrelation and partial autocorrelation functions of the residuals are included in Comparison Analysis Section.

The user will have access only in the autocorrelation and partial autocorrelation visualizations in order to evaluate the stationarity of the time series data as these visualizations are easily readable in contrast with the other stationary tests that need mathematical background and that is the reason that they consist background work to the algorithm in order to provide reliable results.

Next visualization shows the forecast values of 10 years forward after fitting an ARIMA model along with the corresponding confidence intervals for each predicted value.

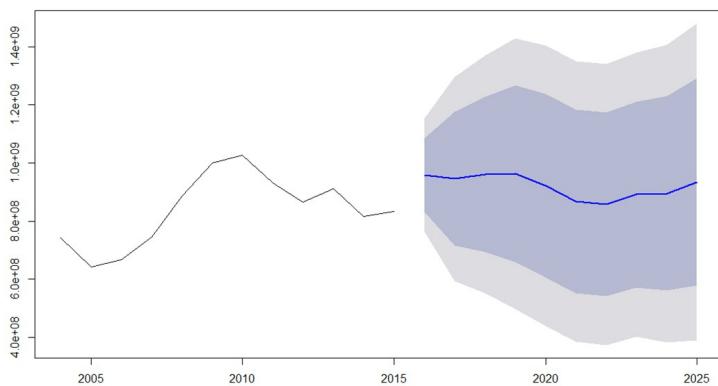


Figure 17. Forecasts for 10 years forward

3.3 Clustering and Similarity learning

Kleanthis, Jaroslav

3.3.1 General description

Cluster Analysis is used as an iterative process of knowledge discovery and involves a set of techniques and algorithms used to find divide the data into groups of similar observations (clusters). There are various algorithms depending the nature of the problem to be solved that differ significantly in their notion of how to form and define a cluster. The appropriate clustering algorithm, the distance function, density threshold and other parameter selection depend on the dataset.

Budget data of municipalities across Europe do have such internal patterns in different levels. In order to meet user needs and eventually the tasks described in detail in Deliverable 2.3-“*Requirements for Statistical Analytics and Data Mining Techniques*” and summarized in the following table, we developed “*Cluster.OBeu*” package which was built in R Software environment and it is available in github¹⁷.

Need	Description	Discussion	Task No.
N01	Filtering commensurable objects	We will implement this as a pre-processing step for the other data mining and analytics tasks, e.g. pattern mining and outlier	T01

17 <https://github.com/okgreece/Cluster.OBeu>

		<p>detection. To make this an interesting data mining task as well, we will not limit to e.g. municipalities of similar population size but also incorporate additional features like geospatial classifications, social and economic information, and consider multiple dimensions at once.</p> <p>Therefore the corresponding task involves introducing a similarity measure on different entities like locations and organizations (i.e. similarity learning) and a clustering grouping comparable items together.</p> <p>The corresponding demand to enrich the data with those additional features listed above is formulated as requirement (R19).</p>	
--	--	--	--

Table 18. List of needs covered by clustering and similarity analysis

This package includes functions for Cluster Analysis techniques that are used in OpenBudgets.eu (OBEU) fiscal datasets. “*Cluster.OBeu*” is based on car, cluster, clValid, dendextend¹⁸, jsonlite¹⁹, mclust²⁰, RCurl²¹, reshape²² and stringr²³ libraries. This package provides different clustering models to be selected through an evaluation process or to be defined by the end user.

3.3.2 Input & output

User input

The user should define the “*dimensions*”, “*measured.dim*” and “*amount*” parameters to form the cluster data. The parameter “*cl.aggregate*” can be used to aggregate the data with a different way. The user can also interact with the selection of the clustering algorithm, the number of clusters and the metric distance by specifying the “*cl.method*”, “*cl.num*” and “*cl.dist*” parameters respectively. The default order of the model, is fixed to select the best clustering model and number of clusters by evaluating internal and stability measures.

The following table summarizes the input parameters:

18 <https://cran.r-project.org/web/packages/dendextend/>

19 <https://cran.r-project.org/web/packages/jsonlite/>

20 <https://cran.r-project.org/web/packages/mclust/>

21 <https://cran.r-project.org/web/packages/RCurl/>

22 <https://cran.r-project.org/web/packages/reshape/>

23 <https://cran.r-project.org/web/packages/stringr/>

Input	Description
json_data	The json string, URL or file from Open Spending API
dimensions	The time label of the json cluster data
amounts	The amount label of the json cluster data
measured.dim	The dimensions to which correspond amount/numeric variables
cl.aggregate	The aggregation function
cl.method	The clustering method algorithm (optional)
cl.num	The number of clusters (optional)
cl.dist	The distance metric (optional)

Table 19. Table of user input parameters for clustering and similarity analysis

Pre-processing of input

“Cluster.OBeu” package includes functions that automatically analyzes the input cluster data. The clustering algorithm, the number of clusters and the distance metric of the clustering model are set to the best selection using internal and stability measures. The end user can also interact with the cluster analysis and specify these parameters.

The final returns are the parameters needed for visualizing the cluster data depending on the selected algorithm and the specification parameters, as long as some comparison measure matrices (see section for Comparison Analysis).

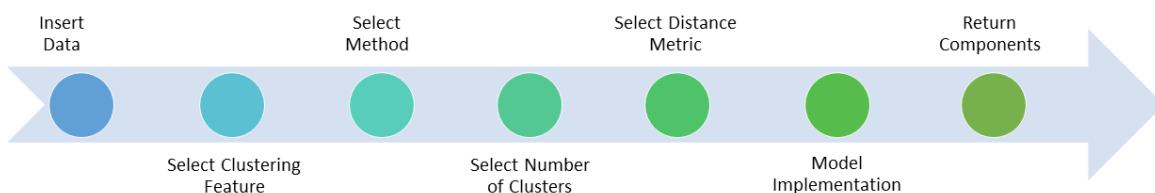


Figure 18.: Cluster Analysis Process

Hierarchical clustering

Hierarchical clustering is a method of cluster analysis in order to find and build a hierarchy of clusters. There are two common types of strategies for hierarchical clustering:

- Agglomerative clustering
- Divisive

clustering

Agglomerative clustering is a "bottom-up" approach, specifically each observation starts in its own cluster, then the similarity distance is computed between each of the clusters and the

most similar clusters are merged. The complexity of these clustering makes them too slow for large data sets.

Divisive clustering is a "top-down" approach in which all observations are assigned to a single cluster, and the splits are performed recursively and moving down the hierarchy there is one cluster for each observation.

The results of hierarchical clustering are presented in a visualization called dendrogram. In order to form clusters in these two algorithms, a measure of dissimilarity between groups of observations is needed. As a measure is used a metric between pairs of observations and a linkage criterion in order to specify the dissimilarity of sets as a function of the pairwise distances of observations in the sets. The most common metrics are Euclidean or squared Euclidean distance.

The advantages of hierarchical clustering are that any valid metric can be used and that requires either the observations to compute the distance matrix or only the distance matrix.

The choice of an appropriate metric influences the shape of the clusters, as some elements may be close to one another according to one distance and farther away according to another. For example, in a 2-dimensional space, the distance between the point (1,0) and the origin (0,0) is always 1 according to the usual norms, but the distance between the point (1,1) and the origin (0,0) can be 2 under Manhattan distance, under Euclidean distance, or 1 under maximum distance.

The following table shows some metrics for hierarchical clustering that included in the package:

Distance	Formula
Euclidean	$\ a - b\ _2 = \sqrt{\sum_i (a_i - b_i)^2}$
Squared Euclidean	$\ a - b\ _2^2 = \sum_i (a_i - b_i)^2$
Manhattan	$\ a - b\ _1 = \sum_i a_i - b_i $
Maximum	$\ a - b\ _\infty = \max_i a_i - b_i $

Table 20. Table of parameters used in hierarchical clustering

The linkage criterion that used in the package “*Cluster.OBeu*” and determines the distance between sets of observations are:

Linkage Clustering	Formula
---------------------------	----------------

Maximum or complete	$\max \{ d(a, b) : a \in A, b \in B \}$
Minimum or single	$\min \{ d(a, b) : a \in A, b \in B \}$
Mean or average	$\frac{1}{ A B } \sum_{a \in A} \sum_{b \in B} d(a, b)$
Centroid	$\ c_s - c_t\ $, where c_s and c_t are the centroids of clusters s and t , respectively.

Table 21. Table of parameters used in linkage clustering

[https://en.wikipedia.org/wiki/Hierarchical_clustering
http://www.saedsayad.com/clustering_hierarchical.htm]

k-means clustering

The k-means clustering is a procedure of vector quantization, which divides n observations into k clusters. Each observation is considered to belong to the cluster with the closest mean. The given results are in a division of the data space into Voronoi cells.

There is a variety of efficient algorithms that converge quickly to a local optimum. The K-means clustering aim to partition a given set of n observations (x_1, x_2, \dots, x_n) , where each observation is a d -dimensional real vector, into k ($\leq n$) sets $S = \{S_1, S_2, \dots, S_k\}$ so as to minimize the sum of distance functions of each point in the cluster to the K center. In other words, its objective is to find:

$$\arg \min_S \sum_{i=1}^k \sum_{x \in S_i} \|x - \mu_i\|^2$$

where μ_i is the mean of points in S_i .

The standard algorithm (Lloyd's algorithm) is the default approach of the k-means clustering that involves two steps the:

- Assignment step and update step

Given a dataset of x_1, x_2, \dots, x_n and an initial set of k means $m_1^{(1)}, \dots, m_k^{(1)}$, in the first step, each observation is assigned to the cluster whose mean produce the least within-cluster sum of squares. The squared Euclidean distance is used to divide the observations into groups according to the Voronoi diagram generated by the means.

$$S_i^{(t)} = \{x_p : \|x_p - m_i^{(t)}\|^2 \leq \|x_p - m_j^{(t)}\|^2 \forall j, 1 \leq j \leq k\}$$

where each x_p is assigned to exactly one $S^{(t)}$, even if it could be assigned to more of them.

In the second step new means are calculated to be the centroids of the observations in the new clusters.

$$m_i^{(t+1)} = \frac{1}{|S_i^{(t)}|} \sum_{x_j \in S_i^{(t)}} x_j$$

Partitioning Around Medoids (PAM)

Partitioning Around Medoids algorithm is the most common variation of k-means clustering that uses the medoid instead of the mean. The k-medoids is partitional and attempt to minimize the distance between points labeled to be in a cluster and a point defined as the center of that cluster. The difference is that k-medoids chooses specific observations as clusters centers and uses distance metrics to determine the distances among the observations of the dataset.

A medoid is the object of a cluster whose average dissimilarity to all the objects in the cluster is minimal, in other words is the most central located observation in the cluster. PAM intend to find a sequence of medoids. The goal is to minimize the average dissimilarity of the objects to their closest selected objects.

For a given dataset, the algorithm first selects k of the n observations as medoids and associates the rest observations to the closest medoid. While the cost of formation decreases, for each medoid m and for each other non medoid observation o swaps m and o to recompute the sum of distances of observations to their medoid (cost) and if this sum/cost increased keeps the previous observation as medoid.

Clustering for Large Applications (CLARA)

Clustering for Large Applications is used to cluster much bigger datasets also in high dimensions than k-means and Partition Around Medoids algorithm.

This algorithm has a different approach than the other algorithms as it focuses on a sample of the dataset and clusters the sample instead of the entire dataset assigning the rest observations of the dataset to these clusters.

First a sample of 40+2k of the dataset is drawn and PAM algorithm is applied on this sample in order to determine the medoids of the sample. If the sample is drawn in a sufficiently random way, we can assume that the medoids of the sample can sufficiently represent the medoids of the entire data set. This algorithm continues by drawing another four samples, in order to select the most representative observations (medoids) and eventually return the best clustering results. If the average dissimilarity of the clustering is less than the last minimum value, the new value will be used as the new minimum value, and retain the corresponding k medoids as the best set of medoids obtained until the algorithm finishes its iterations.

Fuzzy clustering

Fuzzy clustering is a type of clustering in which each observation of the data can belong to more than one cluster in contrast with other methods. Fuzzy C-Means (FCM) Algorithm is used in “*Cluster.OBeu*” package and is similar to the k-means algorithm.

This algorithm attempts to divide a finite collection of n elements x_1, x_2, \dots, x_n into c fuzzy clusters.

Given a finite set of data, the algorithm returns a list of c cluster centres $C = \{c_1, c_2, \dots, c_c\}$ and a partition matrix $W = w_{i,j}$ with $i=1,2,\dots,n$ and $j=1,2,\dots,c$ where each element $w_{i,j}$ is the degree to which element x_i belongs to cluster c_j .

The aim of FCM is to minimize an objective function:

$$\arg \min_C \sum_{i=1}^n \sum_{j=1}^c w_{i,j}^m \|x_i - c_j\|^2$$

where:

$$w_{i,j} = \frac{1}{\sum_{k=1}^c \left(\frac{\|x_i - c_j\|}{\|x_i - c_k\|} \right)^{\frac{2}{m-1}}}$$

Model Based Clustering

In this algorithm, the data is assumed to be generated by a model. The model-based clustering tries to reach the original model from the data. This model defines clusters and an assignment of objects to clusters. Sample observations arise from a distribution that is a mixture of two or more components (denoted by G). Each component k (i.e. group or cluster) is modeled by the normal or Gaussian distribution.

EM (Expectation-Maximization) algorithm is used to estimate the model parameters by hierarchical model based clustering. The clusters are centered around their means with increased density for points near their means.

The covariance matrix determines the shape, volume and orientation of each cluster. The model options used are based in mclust package, and they are represented by three identifiers of three possible values (E,V,I). The first identifier refers to volume, the second to shape and the third to orientation, for example EII, VII, EEI, VEI, EVI, VVI, EEE, EEV, VEV and VVV. Where E stands for equal, V for variable and I for coordinate axes.

For example, EEE means that the clusters have the same volume, shape and orientation in p-dimensional space.

Principal Component Analysis

Used internally to determine the components in axes mainly for visualization of clusters with convex hulls or ellipses enclosing each identified cluster. PCA together with convex hulls or ellipses allow better understanding, visualization and separation of clusters in 2D space.

PCA is a statistical technique for finding patterns in high dimensional data. The main advantage is extracting important dimensions and reducing the number of dimensions, without loss of information. It thus allows the projection of multidimensional data to lower dimensions. The main concept of PCA is to find principle components of data. They are the directions where there is the most variance, the directions where the data is most spread out - they are independent linear combinations of the original dimensions. It uses correlation or covariance matrix, eigenvectors and eigenvalues as the underlying concept. Each principal component is calculated as a linear combination of an eigenvector of the correlation matrix with the variables. The number of principal components is less than or equal to the number of original variables. The first principal component has the largest possible variance and all subsequent components have lower variances, where computed eigen value represents the variance. In order to visualize the clusters in 2D space, we use two top principle components with top eigenvalues that should represent main variance of the data for the 2D space.

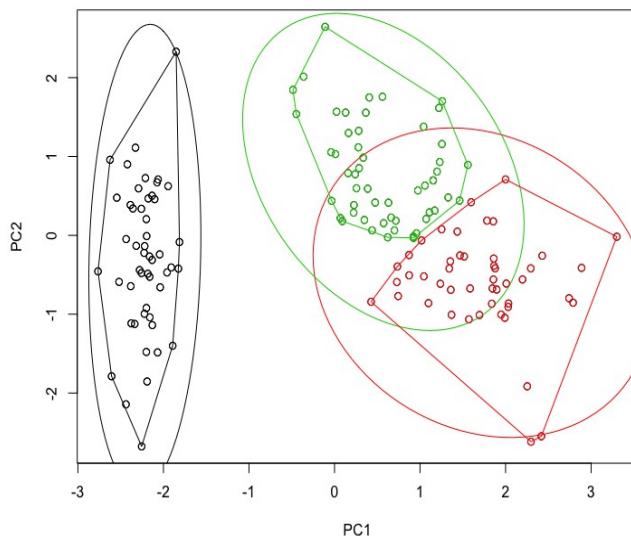


Figure. 19: Example of convex hulls and ellipses that visualize borders of clusters.

Figure presents an example (clustering of the standard iris dataset) of convex hulls and ellipses that visualize the borders between clusters. The example uses only top two principal components (PC1 and PC2) to reduce dimensionality (originally 4 dimensions).

Convex hulls and ellipses are computed on the two main principal components of input data. To compute PCA in R we use `prcomp` function from the core stats package. The computation of convex hulls is based on `chull` function available from the core package `grDevices`. For computing of ellipses we use `dataEllipse` function from the `car` package [<https://cran.r-project.org/package=car>].

Output structure

The output of this process is a list in json format divided into four components of parameters and results with subcomponents:

Component	Subcomponent
<code>cl.meth</code>	-
<code>clust.numb</code>	-
<code>data.pca</code>	-
<code>modelparam</code>	Model parameters (depends on the implemented model)
	<code>compare</code>

Table 22. List of output components of clustering analysis

The component “*cl.meth*” includes the name of the clustering algorithm under the selected number of clusters “*clust.numb*”. “*data.pca*” are the data after the Principal Component Analysis and “*model.param*” component consists of the model parameters and the comparative measure matrices are included in “*compare*” subcomponent and described in Section 3.4.

The output of each clustering algorithm implementation depends on the selected model and described below:

Hierarchical Clustering

Output	Description
<i>data</i>	Input Data
<i>clustered.data</i>	Coordinates and node names of data

Table 23. List of output components of hierarchical clustering analysis

K-Means

Output	Description
<i>data.pca</i>	Data after implementation of Principal Component Analysis
<i>cluster.ellipses</i>	Ellipses
<i>cluster.convex.hulls</i>	Convex Hulls

<i>data</i>	Initial Data
<i>clusters</i>	Clusters
<i>cluster.centers</i>	Cluster Centers

Table 24. List of output components of k-means clustering analysis

Partitioning Around Medoids (Pam)

Output	Description
<i>data.pca</i>	Data after implementation of Principal

	Component Analysis
cluster.ellipses	Ellipses
cluster.convex.hulls	Convex Hulls
data	Initial Data
medoids	Medoids Coordinates
medoids.id	Medoids Identification
clusters	Clusters

Table 25. List of output components of Partitioning Around Medoids (clustering analysis)

Clustering Large Applications (Clara)

Output	Description
data.pca	Data after implementation of Principal Component Analysis
cluster.ellipses	Ellipses
cluster.convex.hulls	Convex Hulls
data	Initial Data
medoids	Medoids
medoids.id	Medoids Identification
clusters	Clusters

Table 26. List of output components of Clustering Large Applications (clustering analysis)

Fuzzy Analysis Clustering (Fanny)

Output	Description
data.pca	Data after implementation of Principal Component Analysis
cluster.ellipses	Ellipses

cluster.convex.hulls	Convex Hulls
data	Initial Data
clusters	Clusters

Table 27. List of output components of Fuzzy Analysis Clustering (clustering analysis)

Model Based Clustering

Output	Description
data.list	A list of 2 dimension data frames containing all combinations of variables of the data
data.list.colnames	Variable Names
classification	Classification
data	Initial Data
clusters	Number of Clusters

Table 28. List of output components of Model Based Clustering (clustering analysis)

3.3.3 Sample case

This section is a sample case of Cluster Analysis using the “*Cluster.OBeu*” package providing custom visualizations for Municipalities of Athens and Thessaloniki. Cluster analysis was used to discover the groups of the administrative units in these Municipalities in accordance with their budget expenditure phase amounts.

Data

We selected the data set of expenditure budget phase amounts for Municipalities of Athens and Thessaloniki from 2011 to 2015.

This dataset includes the names of the administrative unit (label), the year, the expenditure amount of draft, revised, reserved, executed and approved money of Athens and Thessaloniki for the selected time window.

The identification used for each observation in the dataset was the name of the administrative unit the year and the city. The observations that had more than one expenditure code in the same year were summarized.

administrative_unit	year	city	draft	revised	reserved	approved	executed
Δ/ΝΣΗ ΣΧΕΔΙΟΥ ΠΟΛΕΩΣ & ΠΟΛΕΟΔΟΜΙΑΣ	2012	Athens	54000	67000	4750	4750	4750
Δ/ΝΣΗ ΚΕΝΤΡΟΥ ΕΣΥΠΗΡΕΤΗΣΗΣ ΠΟΛΙΤΩΝ	2012	Athens	1150000	1150000	1149445	1149445	1149445
Δ/ΝΣΗ ΑΣΤΙΚΗΣ ΚΑΤΑΣΤΑΣΗΣ	2012	Athens	76000	76000	10578	10578	10578
Δ/ΝΣΗ ΔΗΜΟΤΙΚΩΝ ΠΡΟΣΟΔΩΝ	2012	Athens	309000	309000	201057	174836	174836
Δ/ΝΣΗ ΔΗΜΟΤΙΚΩΝ ΠΡΟΣΟΔΩΝ	2013	Athens	525000	205000	113541	91391	90040
Δ/ΝΣΗ ΔΗΜΟΤΙΚΩΝ ΠΡΟΣΟΔΩΝ	2014	Athens	200000	154000	130356	124362	124362
Δ/ΝΣΗ ΔΗΜΟΤΙΚΩΝ ΠΡΟΣΟΔΩΝ	2015	Athens	1380000	1400697	587903	586339	584889
Δ/ΝΣΗ ΑΠΟΚΕΝΤΡΩΣΗΣ & ΔΙΟΙΚΗΣΗΣ	2013	Athens	712830	971155	594470	594470	594470
Δ/ΝΣΗ ΑΠΟΚΕΝΤΡΩΣΗΣ & ΔΙΟΙΚΗΣΗΣ	2014	Athens	890450	976450	514403	514403	514364
Δ/ΝΣΗ ΑΠΟΚΕΝΤΡΩΣΗΣ & ΔΙΟΙΚΗΣΗΣ	2015	Athens	629200	867155	414607	412588	412547
Δ/ΝΣΗ ΣΧΕΔΙΟΥ ΠΟΛΕΩΣ & ΔΟΜΗΣΗΣ	2013	Athens	10000	10000	5000	5000	5000
Δ/ΝΣΗ ΣΧΕΔΙΟΥ ΠΟΛΕΩΣ & ΔΟΜΗΣΗΣ	2014	Athens	103000	74000	73822	73822	73822
Δ/ΝΣΗ ΣΧΕΔΙΟΥ ΠΟΛΕΩΣ & ΔΟΜΗΣΗΣ	2015	Athens	60000	90000	49917	49917	49917
Δ/ΝΣΗ ΚΕΝΤΡΟΥ ΕΣΥΠΗΡΕΤΗΣΗΣ ΠΟΛΙΤΩΝ (Κ.Ε.Π.)	2013	Athens	1150000	1150000	805000	805000	805000
Δ/ΝΣΗ ΚΕΝΤΡΟΥ ΕΣΥΠΗΡΕΤΗΣΗΣ ΠΟΛΙΤΩΝ (Κ.Ε.Π.)	2014	Athens	1145000	1137000	1029000	1029000	1029000
Δ/ΝΣΗ ΚΕΝΤΡΟΥ ΕΣΥΠΗΡΕΤΗΣΗΣ ΠΟΛΙΤΩΝ (Κ.Ε.Π.)	2015	Athens	708000	708000	708000	708000	708000
Γενικές υπηρεσίες	2011	Thessaloniki	136634453	138244125	113208354	113208354	113208354
Γενικές υπηρεσίες	2012	Thessaloniki	149249594	151252316	111457477	111457477	111457477
Γενικές υπηρεσίες	2013	Thessaloniki	37121281	36492152	31305799	31305799	31305799
Γενικές υπηρεσίες	2014	Thessaloniki	33232291	35793857	31501129	31501129	31501129
Γενικές υπηρεσίες	2015	Thessaloniki	34381569	33643478	25189823	25189823	25189823
Δημοτική Αστυνομία	2011	Thessaloniki	4243882	4514582	4046914	4046914	4046914

Figure 20.: Fiscal dataset of Athens and Thessaloniki

We define as variables of the dataset the column names (label, year, draft, revised, reserved, executed and approved).

Partitioning Around Medoids algorithm was selected to cluster the expenditure budget phase data and the number of clusters was auto selected according to the internal and stability measures results. PAM selected because is robust to outliers and in these data there are a lot of outlier values described in sample case in section of Descriptive Statistics.

PAM clustering algorithm is based on the search for k-representative observations of the dataset, the medoids, by minimizing the overall dissimilarity between the representative observation of each cluster and its members. The sum of dissimilarities in this method was calculated using the Manhattan distance. The Figure below shows the resulting visualization of this algorithm:

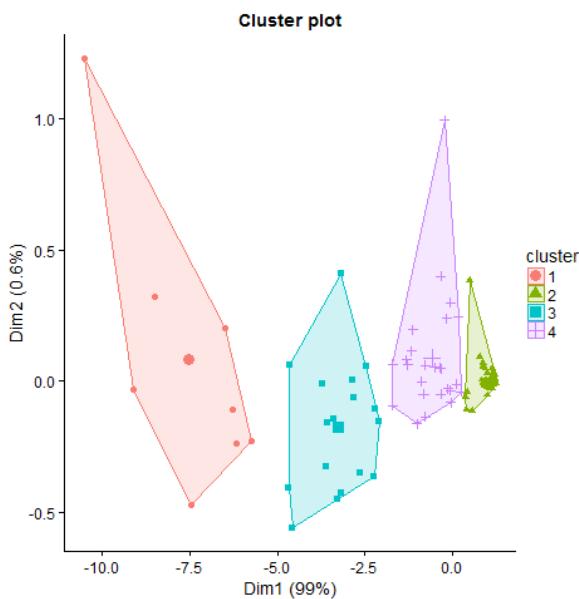


Figure 21.: Partitioning Around Medoids Visualization

These four groups differ with respect to their budget phase amounts. Every administrative unit that belongs to a specific group has similar budget phase amounts with the corresponding representative unit. As medoid or representative called every observation that could represent the features of the clusters it belongs. Table below shows the medoids of the expenditure budget phase amounts, selected by the PAM algorithm:

Medoids	Cluster
General Services Thessaloniki, 2011	1 (red)
Decentralization and Administration Athens, 2014	2 (blue)
Social Welfare Athens, 2011	3 (purple)
General Services Thessaloniki, 2015	4 (green)

Table 29.: The most representative expenditure budget phases amounts of Municipalities of Athens and Thessaloniki

The administrative unit of Thessaloniki that is responsible for General Services represents two different clusters. This is a strong indication, as it is selected as representative, that this administrative unit made important changes in its strategy in this time window to adapt the provided services to the citizens according to the available budget.

Based on the nature of the selected data this sample case is considered also as part of comparative analysis.

3.4 Comparative analysis

3.4.1 General description

Comparative Analysis is the process of comparison of two or more comparable alternatives, processes, sets of data, models and algorithms. In the three aforementioned packages there are measures and visualizations parameters that were estimated in order to meet user needs and eventually the tasks described in detail in Deliverable 2.3- "*Requirements for Statistical Analytics and Data Mining Techniques*" and summarized in the following table.

Need	Description	Discussion	Task No.
N02	Version tracking of budgets	This data mining and analytics need refers to a comparative analysis of budget lines along the different budget phases and can be extended to find and measure trends in doing so.	T02
N07	Temporal trend of the difference between planned and actual spending	This data mining and analytics need is related to (N02) and extends it with a temporal dimension involving budget data from several years and incorporating corresponding spending data. Another aspect is to investigate and analyze the reasons for the detected trends.	T06
N12	Perform comparisons measuring how the data has changed when a data set has been updated	This need refers to a comparative analysis of different versions of uploaded datasets.	T09
N15	Pay special focus on analyzing the spending and management of EU budget funds	For the analysis of EU budget funds we will use several of the already mentioned methods: Time series analysis (T02), comparative analysis (T09), rule/pattern mining (T11), and outlier detection (T12). The focus on analyzing EU budget funds is formulated as requirement (R04).	T02, T09
N18	Perform comparative analysis of certain budget and expenditure areas through the use of timelines;	This need refers to a broad analysis of budget data with special interest to the temporal, geographical and thematic dimension.	T18

	geographically; and by sector		
N20	Comparisons of previous years' budgets with the current one.	This need refers to a comparative analysis of different years' budgets.	T19
N22	Comparative analysis	This need is extending (N02), (N12), (N18) and (N20) to perform comparative analysis of budget and spending data in general.	T20
N32	Compare the same budget line across countries and cities	Comparative analysis along the geospatial dimension is a special case of (T18).	T29

Table 30.: Comparative packages fulfills eight needs in D2.3

This package includes functions to calculate some comparative indices and matrices that evaluate the implemented model and compare to other in terms of time series cluster and descriptive analysis. In OpenBudgets.eu platform the comparisons will also be provided with visualizations.

3.4.2 input & output

User input

User input were described in previous sections 3.1, 3.2, 3.3.

Pre-processing of input

This section describes some comparative techniques that were not included in sections 3.1, 3.2, 3.3.

Log-likelihood

The log-likelihood is being used instead of the likelihood function for computational convenient since logarithm is a monotonically increasing function. The log-likelihood function can also be used in maximum likelihood estimation since it achieves at the same point with the likelihood function its maximum value.

For example, if we have a set of statistically independent observations with the log-likelihood function we are going to have a sum of individual logarithms, and derivate of the sum of terms is easier to compute, the most times, comparatively with the derivative of the product of these terms (that we will have if we had used the likelihood function).

Because the maximum of the log-likelihood is required, the higher value the better.

Akaike information criterion

The AIC or Akaike information criterion constitutes a criterion that we can select between statistical models that concern a given dataset. Specifically, there is the ability of estimate the quality of each model, for the data, comparatively to each of the other models. Therefore, AIC provides a means for model selection. AIC provides, in information theory, a relative estimate of the information lost when a given model is used to define the process that generates the data. This criterion has to do with the exchange between the goodness of fit and the complexity of the model. AIC will not give any warning if all the candidate models do not fit perfectly. Suppose that we have a statistical model of some data. Let L be the maximum value of the likelihood function for the model; let k be the number of estimated parameters in the model. Then the AIC value of the model is the following:

$$AIC = 2k - 2 \ln(L)$$

Given a set of nominee models for the data, the preferred model is the one with the minimum AIC value.

AICc provides an alteration of AIC when we consider the correction for finite sample sizes. The formula for AICc depends on the statistical model. Assuming that the model is univariate, linear, and has normally distributed residuals (conditional upon regressors), the formula for AICc is defined as follows:

$$AICc = AIC + \frac{2(k+1)(k+2)}{n-k-2}$$

where n denotes the sample size and k denotes the number of parameters. The difference between AICc and AIC is that the first one has greater penalty for extra parameters.

If we use AIC, instead of AICc:

- The probability of models with too many parameters to be selected is increasing when n is not many times larger than k^2 , i.e. of overfitting. The probability of AIC overfitting can be essential, in some cases.
- The two criterions will give identical (relative) valuations If all the candidate models have the same k ; hence, there will be no disadvantage in using AIC instead of AICc.
- If n is many times larger than k^2 , then the correction will be insignificant; consequently, there will be a negligible disadvantage in using AIC instead of AICc.

AIC has many advantages, it can be used between models with different error distribution, it is valid for both nested and non nested models. Differently, some issues that might denote problems are that AIC cannot be used to compare models of different datasets, for all models the response variable should be the same. Also, there is a confusion between AIC and null hypothesis test, so there must be a careful use of language (e.g. rejected, significant).

Bayesian information criterion

Bayesian information criterion (BIC) or Schwarz information criterion (SIC) is a criterion of choosing statistical models but among a finite set of models. It has been introduced as a

contender to the AIC. As AIC, the preferred model is the one with the lowest BIC. In some cases, when BIC has to deal with overfitting, it inserts a penalty test for the number of parameters in the model. The BIC value of the model is the following:

$$\text{BIC} = \ln(n)k - 2\ln(\hat{L}).$$

Where \hat{L} is the maximum likelihood function of the model, n is the number of observations and k is the number of free parameters to be estimated or the number of regressors if the model is linear regression. BIC provides a large-sample estimator of a transformation of the Bayesian posterior probability associated with the approximating model.

There are two main limitations that must be considered when we use this criterion. First, BIC cannot deal with a complex set of models, it has a preference for simpler models compare to AIC. Second, the approximation for BIC is valid only when the sample size n is much larger than k.

Moreover, if we want to compare estimated models with this criterion the numerical values of the dependent variable must be identical for all estimates being compared and the models being compared need to be non-nested.

Silhouette Visualization

Silhouette analysis is a cluster validation approach that measures how well an observation is clustered through the average distance between clusters.

The silhouette value is considered as a measure of how similar an object is to its own cluster (cohesion) compared to other clusters (separation). The silhouette ranges from -1 to 1, where a value near +1 indicates that the object is well clustered to its own cluster and poorly matched to neighboring clusters. If most objects have a high value, then the clustering configuration is appropriate, unlike the case where many points have a low or negative value where the clustering results have too many or too few clusters.

The Euclidean distance or the Manhattan distance, or any other distance metric can be used in silhouette.

Dunn's partition coefficient

Dunn's partition coefficient measures how close the fuzzy solution is to the corresponding hard solution. The classification of each object with largest membership forms the hard clustering solution.

Dunn's partition coefficient $F_c(k)$ of the clustering, where k is the number of clusters. $F_c(k)$ is the sum of all squared membership coefficients, divided by the number of observations. Its value is between $1/k$ and 1. The normalized form of the coefficient is also given. It is defined as

$$Fc(U) = \frac{F(U) - (1/K)}{1 - (1/K)}$$

and ranges between 0 and 1.

A low value of Dunn's coefficient indicates a very fuzzy clustering, whereas a value close to 1 indicates a near-crisp clustering.

Output structure

The output of this process is a list in json format divided into components and subcomponents of parameters depending the type of analysis. For further details see three previous sections. Tables below show the output structures depending on the implemented analysis. Every descriptive task described in section 3.1 can be considered as comparative technique.

Time Serie Decomposition

Output- Seasonal	Description
stl.win	Spans Smoothers
stl.degree	Smoothers' Degrees
-	-

Output- Non seasonal	Description
stl.degree	Degree of Fit
degfr	Effective degrees of freedom
degfr.fitted	Fitted degrees of freedom

Output- residuals_fitted	Description
residuals	Residual Values
fitted	Fitted Values
time	Time/ Years
line	The y=0 line

Output- Seasonal	Description
arima.order	Arima Orders
arima.coef	Arima Coefficients
arima.coef.se	Coefficients Standard Error

Output- Non seasonal	Description
-	-
-	-
-	-

covariance.coef	Estimated Coefficients Variance	-	-
resid.variance	Residuals Variance	resid.variance	Residuals Variance
not.used.obs	Not Used Observations	-	-
used.obs	Used Observations	used.obs	Used Observations
loglik	Log-Likelihood	loglik	Log-Likelihood
aic	Akaike information criterion	aic	Akaike information criterion
bic	Bayesian information criterion	bic	Bayesian information criterion
aiacc	Corrected Akaike information criterion	gcv	Generalized cross validation

Table 31.: Lists of parameters in Time Serie Decomposition

Model Fitting

Output- Seasonal and Nonseasonal	Description
arima.order	Arima Orders
arima.coef	Arima Coefficients
arima.coef.se	Coefficients Standard Error

Output-Seasonal and Nonseasonal	Description
resid.variance	Residuals Variance
variance.coef	Estimated Coefficients Variance
not.used.obs	Not Used Observations
used.obs	Used Observations
loglik	Log-Likelihood

aic	Akaike information criterion
bic	Bayesian information criterion
aicc	Corrected Akaike information criterion

Table 31.: Lists of parameters in model fitting

Hierarchical Cluster Analysis

Output	Description
compare	The total sum of squares

Table 32.: List of parameters in hierarchical cluster analysis

K-Means Cluster Analysis

Output	Description
total.sumOfsquares	The total sum of squares
within.sumofsquares	Vector of within-cluster sum of squares-One component per cluster
total.within.sumofsquares	Total within-cluster sum of squares
between.sumofsquares	The between-cluster sum of squares
cluster.size	Size of Clusters- one size per cluster

Table 33.: List of parameters in k-means cluster analysis

Partitioning Around Medoids (Pam)

Output	Description
cluster.size	Size of Clusters-One size per cluster
cluster.max_diss	The total sum of squares
cluster.av_diss	Vector of within-cluster sum of squares-One component per cluster
cluster.diameter	Total within-cluster sum of squares
cluster.separation	The between-cluster sum of squares

Table 34.: List of parameters in partitioning around medoids analysis

Silhouette Visualization

Output-silhouette.info	Description
widths	Width of each bar
clus.avg.widths	the average silhouette width per cluster
avg.width	the average silhouette width for the dataset

Table 35.: List of parameters in *Silhouette Visualization**Clustering Large Applications Algorithm*

Output	Description
cluster.size	Size of Clusters-One size per cluster
cluster.max_diss	The total sum of squares
cluster.av_diss	Vector of within-cluster sum of squares-One component per cluster
cluster.diameter	Total within-cluster sum of squares
cluster.separation	The between-cluster sum of squares

Table 36.: List of parameters in *Clustering Large Applications**Fuzzy Analysis Clustering*

Output	Description
membership	A matrix containing the memberships for each pair consisting of an observation and a cluster.
coeff	Dunn's partition coefficient
memb.exp	The membership exponent used in the fitting criterion
convergence	A named vector with iterations, the number of iterations needed and converged indicating if the algorithm converged (in maxit iterations within convergence tolerance tol).
objective	Value of criterion maximized during the partitioning algorithm

Table 37.: List of parameters in *Fuzzy Analysis Clustering**Model Based Clustering*

Output	Description
model.name	A character string denoting the model at which the optimal BIC occurs
observations	The number of observations in the data.
data.dimension	The data variables
clust.numb	The number of clusters
all.Bics	All BIC values
optimal.bic	Optimal BIC value
optimal.loglik	The log-likelihood corresponding to the optimal BIC.
numb.estimated.parameters	The number of estimated parameters
hypervolume.parameter	The hypervolume parameter for the noise component if required, otherwise set to NULL
mixing.proportion	A vector whose kth component is the mixing proportion for the kth component of the mixture model. If missing, equal proportions are assumed.
mean.component	The mean for each component. If there is more than one component, this is a matrix whose kth column is the mean of the kth component of the mixture model.
uncertainty	The uncertainty associated with the classification.

Output-variance.components	Description
modelName	A character string indicating the model.
d	The dimension of the data.
G	An integer vector specifying the numbers of mixture components (clusters) for which the BIC is to be calculated
sigma	For all multidimensional mixture models. A d by d by G matrix array whose [,,k]th entry is the covariance matrix for the kth component of the mixture model.
scale	For diagonal models "EEI", "EVI", "VEI", "VVI" and constant-

	shape models "EEV" and "VEV". Either a G-vector giving the scale of the covariance (the dth root of its determinant) for each component in the mixture model, or a single numeric value if the scale is the same for each component.
shape	For diagonal models "EEI", "EVI", "VEI", "VVI" and constant-shape models "EEV" and "VEV". Either a G by d matrix in which the kth column is the shape of the covariance matrix (normalized to have determinant 1) for the kth component, or a d-vector giving a common shape for all components
orientation	For the constant-shape models "EEV" and "VEV". Either a d by d by G array whose [,,k]th entry is the a matrix whose columns are the eigenvectors of the covariance matrix of the kth component, or a d by d orthonormal matrix if the mixture components have a common orientation. The orientation component is not needed in spherical and diagonal models, since the principal components are parallel to the coordinate axes so that the orientation matrix is the identity.

Table 38.: Lists of parameters in *Model Based Clustering*

3.4.3 Sample case

This section is a sample case of Comparative Analysis using the “TimeSeries.OBeu”, “DescriptiveStats.OBeu” and “Cluster.OBeu” packages providing custom visualizations for Municipalities of Athens and Thessaloniki. There are some sample comparative visualization and measure tables in order to meet the user needs. Descriptive tasks used to understand better the data that will be compared, time series visualization showed the progress of the selected variable and cluster analysis was used to discover the groups of the administrative units in these Municipalities in accordance with their budget expenditure phase amounts along with the model’s evaluation.

Data

We selected the data set of expenditure budget phase amounts for Municipalities of Athens and Thessaloniki from 2011 to 2015.

This dataset includes the names of the administrative unit (label), the year, the expenditure amount of draft ,revised, reserved,executed and approved money of Athens and Thessaloniki for the selected time window.

The identification used for each observation in the dataset was the name of the administrative unit the year and the city. The observations that had more than one expenditure code in the same year were summarized.

administrative_unit	year	city	draft	revised	reserved	approved	executed
Δ/ΝΣΗ ΣΧΕΔΙΟΥ ΠΟΛΕΩΣ & ΠΟΛΕΟΔΟΜΙΑΣ	2012	Athens	54000	67000	4750	4750	4750
Δ/ΝΣΗ ΚΕΝΤΡΩΝ ΕΞΥΠΗΡΕΤΗΣΗΣ ΠΟΛΙΤΩΝ	2012	Athens	1150000	1150000	1149445	1149445	1149445
Δ/ΝΣΗ ΑΣΤΙΚΗΣ ΚΑΤΑΣΤΑΣΗΣ	2012	Athens	76000	76000	10578	10578	10578
Δ/ΝΣΗ ΔΗΜΟΤΙΚΩΝ ΠΡΟΣΟΔΩΝ	2012	Athens	309000	309000	201057	174836	174836
Δ/ΝΣΗ ΔΗΜΟΤΙΚΩΝ ΠΡΟΣΟΔΩΝ	2013	Athens	525000	205000	113541	91391	90040
Δ/ΝΣΗ ΔΗΜΟΤΙΚΩΝ ΠΡΟΣΟΔΩΝ	2014	Athens	200000	154000	130356	124362	124362
Δ/ΝΣΗ ΔΗΜΟΤΙΚΩΝ ΠΡΟΣΟΔΩΝ	2015	Athens	1380000	1400697	587903	586339	584889
Δ/ΝΣΗ ΑΠΟΚΕΝΤΡΩΣΗΣ & ΔΙΟΙΚΗΣΗΣ	2013	Athens	712830	971155	594470	594470	594470
Δ/ΝΣΗ ΑΠΟΚΕΝΤΡΩΣΗΣ & ΔΙΟΙΚΗΣΗΣ	2014	Athens	890450	976450	514403	514403	514364
Δ/ΝΣΗ ΑΠΟΚΕΝΤΡΩΣΗΣ & ΔΙΟΙΚΗΣΗΣ	2015	Athens	629200	867155	414607	412588	412547
Δ/ΝΣΗ ΣΧΕΔΙΟΥ ΠΟΛΕΩΣ & ΔΟΜΗΣΗΣ	2013	Athens	10000	10000	5000	5000	5000
Δ/ΝΣΗ ΣΧΕΔΙΟΥ ΠΟΛΕΩΣ & ΔΟΜΗΣΗΣ	2014	Athens	103000	74000	73822	73822	73822
Δ/ΝΣΗ ΣΧΕΔΙΟΥ ΠΟΛΕΩΣ & ΔΟΜΗΣΗΣ	2015	Athens	60000	90000	49917	49917	49917
Δ/ΝΣΗ ΚΕΝΤΡΩΝ ΕΞΥΠΗΡΕΤΗΣΗΣ ΠΟΛΙΤΩΝ (Κ.Ε.Π.)	2013	Athens	1150000	1150000	805000	805000	805000
Δ/ΝΣΗ ΚΕΝΤΡΩΝ ΕΞΥΠΗΡΕΤΗΣΗΣ ΠΟΛΙΤΩΝ (Κ.Ε.Π.)	2014	Athens	1145000	1137000	1029000	1029000	1029000
Δ/ΝΣΗ ΚΕΝΤΡΩΝ ΕΞΥΠΗΡΕΤΗΣΗΣ ΠΟΛΙΤΩΝ (Κ.Ε.Π.)	2015	Athens	708000	708000	708000	708000	708000
Γενικές υπηρεσίες	2011	Thessaloniki	136634453	138244125	113208354	113208354	113208354
Γενικές υπηρεσίες	2012	Thessaloniki	149249594	151252316	111457477	111457477	111457477
Γενικές υπηρεσίες	2013	Thessaloniki	37121281	36492152	31305799	31305799	31305799
Γενικές υπηρεσίες	2014	Thessaloniki	33232291	35793857	31501129	31501129	31501129
Γενικές υπηρεσίες	2015	Thessaloniki	34381569	33643478	25189823	25189823	25189823
Δημοτική Αστυνομία	2011	Thessaloniki	4243882	4514582	4046914	4046914	4046914

Figure 22.: Fiscal dataset of Athens and Thessaloniki

Descriptive measures

We calculated the main descriptive measures for each variable of the budget phase (draft, revised, reserved, approved, executed) in order to get a clearer view of the expenditure management in these two Municipalities from 2011 to 2015.

year	city	number of observations	mean	standard deviation	median	min	max	range
2011	Athens	321	1.377.634,89	4.638.545,64	47.026,98	39,38	52.806.988,51	52.806.949,13
	Thessaloniki	587	486.622,13	2.302.548,21	21.198,44	10,53	45.743.102,12	45.743.091,59
2012	Athens	212	1.715.052,91	5.547.311,09	82.483,66	43,05	54.447.267,13	54.447.224,08
	Thessaloniki	530	509.412,99	2.357.063,10	18.273,71	18,44	43.672.237,20	43.672.218,76
2013	Athens	302	1.132.976,84	4.382.008,81	37.013,80	40,00	49.178.283,98	49.178.243,98
	Thessaloniki	429	418.567,01	1.281.992,59	39.064,51	21,00	11.392.695,38	11.392.674,38
2014	Athens	331	1.073.462,85	3.863.463,96	53.719,13	66,42	39.488.535,74	39.488.469,32
	Thessaloniki	476	488.920,01	1.981.732,65	28.517,68	6,42	33.376.967,48	33.376.961,06
2015	Athens	341	947.742,82	3.462.318,32	55.350,00	60,00	33.413.994,93	33.413.934,93
	Thessaloniki	443	529.717,89	2.084.801,98	23.826,87	11,53	33.366.291,37	33.366.279,84

Figure 23.: Summary table of basic descriptive measures of Executed Expenditure amounts of Athens and Thessaloniki in 2011-2015 period

The above Table presents the basic descriptive statistics of the amounts of budget phases in Athens and Thessaloniki for each year from 2011 to 2015. In 2011, 321 administrative units in Athens executed an average amount of 1,377,634.89€ with the minimum observed expenditure of 39.38€ and maximum of 52,806,988.51€. Whereas, the same year, 587 administrative units in Thessaloniki spent an average amount of 486,622.13€ with the minimum observed expenditure of 43.05€ and maximum of 45,743,102.12€.

From 2011 to 2015 there is a decrease of the number of administrative units that execute their services in contrast to Athens where there is a slight increase.

Despite the fact that there are fewer administrative units that spend comparing 2011 and 2015, the average amount of executed money in 2015 increased at 529,717.89€, meaning that Thessaloniki improved the provided services to the citizens, unlike Athens.

Boxplot

The boxplots visualization below show the executed expenditure amounts of Athens and Thessaloniki for each year from 2011 to 2015 through their quartiles. Using the standard coefficient level (1.5) we can see some outliers (extreme values) which are depicted as individual points. The boxes height (Interquartile Range) varies over the years showing that these municipalities changed the executed expenditure amounts of some services to the citizens.

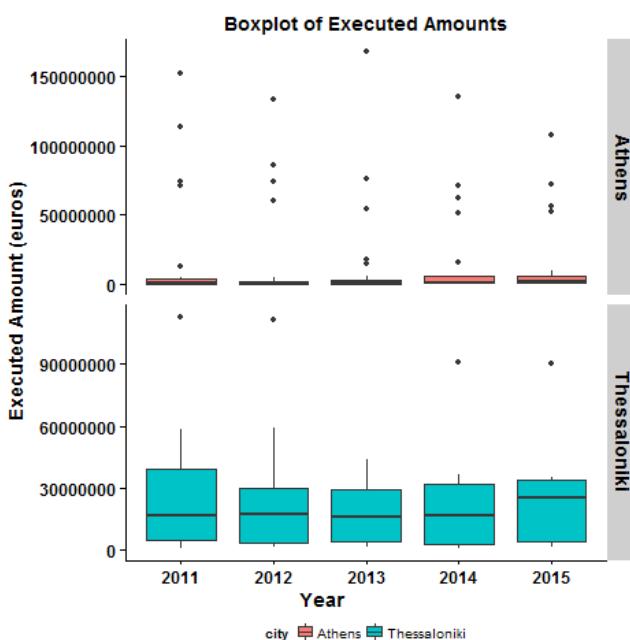


Figure 24.: Boxplots of executed expenditures in Athens and Thessaloniki in 2004-2015 period

Frequencies- Bar graph

The Figure below shows the difference of expenditure amounts that were executed by the administrative units of municipality of Athens and Thessaloniki for each year of the selected time window. It confirms that Athens, as the capital and largest city of Greece, executes more expenditures than Thessaloniki which is the second largest city.



Figure 25.: Executed amounts in Athens and Thessaloniki from 2011-2015

Time Series

This following figure shows the advantages of time series comparison visualizations, where different aspects of comparison and conclusions are depicted.

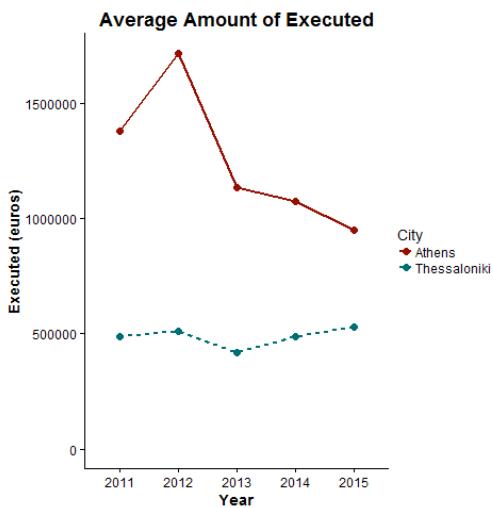


Figure 26.: Time series of Executed Expenditure amounts in Athens and Thessaloniki

This visualization shows clearly the difference of executed expenditure amounts of Athens and Thessaloniki over the years. The basic characteristic of time series of Athens is that after 2012 there is a great decrease of the executed amounts whereas Thessaloniki is more stable.

Clusters Analysis evaluation

We fixed the dataset to 2011-2015 period and used Partition Around Medoids clustering algorithm which provides the same visualization as in Section 3.3.3.

The following figure shows the average silhouette width which indicates that the administrative units were well matched to its own clusters and the clustering configuration is appropriate. In other words, this silhouette visualization interprets and validates the consistency of an administrative unit to its own cluster compared to other clusters.

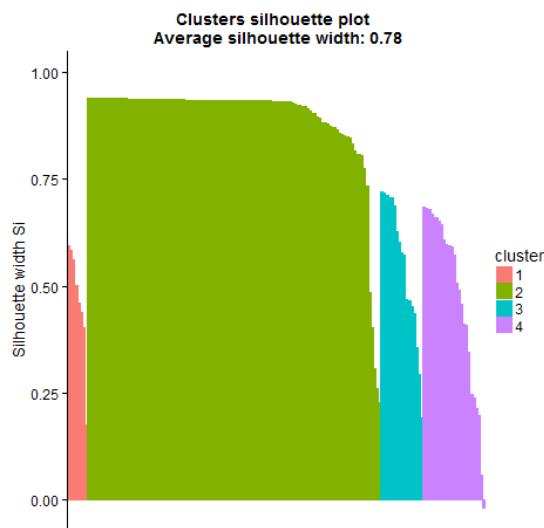


Figure 27.: Cluster silhouette plot

3.5 Rule/pattern mining

Stanislav, David, Václav

3.5.1 General description

The rule mining tasks are implemented using data mining of association rules. The association rules are suitable for description of relations between data attributes also as for creation of classification data mining models. The association rule mining is implemented in

the web data mining system EasyMiner (<http://easyminer.eu>), described in the following paragraphs.

Process of association rules mining belongs among basic tasks of the data mining field. This process leads to discovery of new knowledge about input data. A rule is thus one of the means for knowledge representation which we write either as an IF-THEN statement or as an implication representing a conditional statement of mathematical logic:

This rule has a left and right side (antecedent and consequent). It can be interpreted such as "if the A statement is true, then the B statement is also true". We can use this knowledge for lots of other cases, e.g. predictive analysis, exploration tasks, anomaly detection, business rules composition etc.

The problem of association rules mining is NP-hard, but we can get an approximate result much faster if we want to search only frequent association rules by some thresholds. Frequent association rules are usually mined by support and confidence measures and the final state space is therefore pruned. For this purpose we can use lots of algorithms (apriori, FP-growth, Eclat etc.) and obtain wanted rules very fast if we chose suitable thresholds for the mining process. Output frequent rules with high value of interest measures like support, confidence, lift etc. may be considered as interesting rules and sometimes as useful discovered knowledge for some business problems.

Need	Description	Discussion	Task No.
N14	Identify both good and bad examples of financial management	This need can be address by learning patterns of such good and bad examples and applying a classifier on the learned patterns/rules to the remaining data.	T11
N16	Identify systematic problems of project implementation in different funds and programmes, rather than in-depth engagement with individual projects	This need combines (N14) with performing aggregations on the data to achieve a general, systematic overview on the funds' and programmes' spending data, and incorporating the temporal dimension.	T13
N15	Pay special focus on analyzing the spending and management of EU budget funds	For the analysis of EU budget funds we will use several of the already mentioned methods: Time series analysis (T02), comparative analysis (T09), rule/pattern mining (T11), and outlier detection (T12). The focus on analyzing EU budget funds is formulated as requirement (R04).	T02

Table 39.: Rule-mining packages fulfil three requirements in D2.3

3.5.2 GUHA (complex) association rules

GUHA association rules are the special case of association rules based on complex mathematical theory, GUHA (General Unary Hypotheses Automaton) method, focusing on exploratory data analysis and developed in Prague from the mid-1960s. The GUHA method can be considered as an early example of knowledge discovery in databases and data mining systems. The described concepts are very close to the original association rules mining presented by Agrawal in 1993 (Agrawal, 1993).

The description of the complex theoretical background is beyond the scope of this deliverable, it can be found for example in (Rauch, 2013).

Due to the age of the method, many implementations have been realised, but the most have been suspended. Development and continuous improvement currently runs on system Lisp-Miner, an academic project for support research and teaching of knowledge discovery in databases. Lisp-Miner is composed from several procedures and not all of them function with association rules. The procedure which function with association rules is called 4ft-Miner.

Lets focus on the functionality of GUHA association rules implemented in LISp-Miner system compared to the “traditional” association rules:

- Negation of values - it is possible to include all values of specific attribute with the exception of one
- Conditional rules - the template for generating the rules can contain the optional condition part in addition to antecedent and consequent
- The possibility of using disjunction as the connective between the attributes in antecedent and consequent of the rule
- Partial cedent as the possibility of grouping attributes and further work with these groups in data mining tasks
- The use of coefficient, the multi-element sets. This functionality allows the runtime grouping of values for example to intervals with specified length.
- The number of possible interest measures - GUHA AR offers 17 different interest measures, for example statistical tests such as Fisher or Chi-square tests.

The integration of GUHA association rules and EasyMiner system is currently under development.

Task results

The results of the data mining tasks are association rules in the form:

The rule describes a relation (association) between the antecedent and the consequent. Using the EasyMiner/R system, antecedent and consequent part of the rule are conjunctions of attributes with specific values. Antecedent can contain zero or more attributes. In the case

of classification task, the consequent part of the rule has to contain exactly one attribute (with different values), in other cases the consequent can contain one or more attributes.

The relation between antecedent and consequent part of each rule is described using values of interest measures. The interest measures are calculated using formulas defined on a four field table:

	consequent	not consequent
antecedent	a	b
not antecedent	c	d

Table 40. List of parameters used in rule/pattern mining

Most used interest measures are:

Interest measure	Formula
confidence	
support	
lift	

Table 41. List of measures in rule/pattern mining.

Example rule is explained in the section 3.5.4.

3.5.3 Input & output

The data mining process in EasyMiner consists of the following steps (individually described in the subsections): 1. upload of data, 2. data preprocessing, 3. task definition, 4. processing of results. The steps follow each other and the results of each step are stored in the database of the system EasyMiner. It is suitable for the possibility of repetition of a single step with different parameters (for example solution of more data mining tasks on a single preprocessed dataset).

User input

The first user input for data mining of association rules is a dataset in a form of single table, saved in CSV file. In the table each column presents a different data field, each row presents a single “instance” of data.

In the terms of integration into the system OpenBudgets, the CSV file can be generated using pipelines with appropriate SPARQL query.

The user has to upload the CSV file (optionally zipped) into EasyMiner system using REST API. For small files there is a simple possibility to upload the full file using single POST request. Large files can be uploaded using the sequence of upload requests using the EasyMiner data service, or using the graphical user interface. Input CSV data are, within uploading process, transformed into the transactional form which is more suitable for association mining task.

The CSV file has to contain the names of data fields in the first row. The user configures the upload with parameters for encoding, separator and type of database. For normal datasets

the database is called “limited”, for really large datasets the database is called “unlimited”. For the most dataset processed in the OpenBudgets.eu project the “limited” database is more suitable. From the technical point of view, the database is MariaDB.

Pre-processing of input

Basic algorithms for association rules mining, which are also used in the EasyMiner system, use the transactional form for data representation. Each transaction is one record that consists of a set of items. Algorithms mine frequent itemsets by a minimal support and then make rules from itemsets by a minimal confidence. The mining process thus works only with sets of items where every item is supposed to have a value which is often written as a key-value pair (a column of the table as the key and a value of the record for the column as the value). These types of algorithms do not deal with numeric kinds of value; an item is considered to be a nominal value even if it contains a number. The mining complexity also depends on the total number of items therefore it is desirable to pre-process data by reduction of items amount.

We often want to have few items that have a minimal support and are easily readable and valuable for our business problem. Let we have a numeric column with many distinct numeric values. Probably most of these values are infrequent and a mining tool will not find any frequent association rule with this column. The solution is to discretize this column by merging of numbers and the creation of intervals. Then we get only several items from many, the result will contain more objective rules and it is more likely to find frequent rules because most of items are also frequent.

Within the OpenBudgets.eu project we implement some pre-processing algorithms directly into the EasyMiner system. Now EasyMiner supports these types of discretization and merging methods:

- **Nominal Enumeration:** Merging of items into a user-specified item.
- **Interval Enumeration:** Merging of numeric items into a user-specified intervals.
- **Equidistant Intervals:** Automatic creation of intervals where each interval has an identical range (user input: number of intervals).
- **Equifrequent Intervals:** Automatic creation of intervals where all intervals have similar frequencies (user input: number of intervals).
- **Equisized Intervals:** Automatic creation of intervals where all intervals have similar frequencies by a minimal support (user input: a support threshold).

The EasyMiner system allows us to use these pre-processing algorithms for data transformation of any column of an input table. We can run these tasks within the EasyMiner-Preprocessing API as a RESTful operation. The definition of a pre-processing task is specified by a PMML document which has the XML format. Processed columns are then prepared for the running of an association rules mining task by the EasyMiner-Miner API.

Task definition

The task definition consists from definition of:

- antecedent,

- consequent,
- min. threshold values of selected interest measures,
- limit of max founded association rules.

Antecedent and consequent in the task definition presents a “pattern” for the search of association rules. In the task definition the user specifies the list of attributes, which may occur in the antecedent/consequent part of association rules. Optionally, if the user is interesting only in rules with attribute with a concrete value, the task definition can fix the founded value of an attribute only to a one selected.

The user has also specified the threshold values of selected interest measures. The supported interest measures are min. confidence, support, lift and max rule length. The required interest measures are confidence and support; lift is only optional. If not specified, the max rule length can be calculated automatically as the count of attributes in antecedent and consequent.

For the building of classification models, there is one special “interest measure” called “cba”. For this use case, the task definition has to contain only one attribute in the consequent pattern (with “any value”) and one or more attributes in antecedent pattern. Then if the user attaches the special interest measure “cba”, the system applies the algorithm rCBA on the data mining results. The output is suitable for preparation of classification models, but the user can use it also for quicker exploration analysis (the results contain significantly fewer association rules covering most of the data rows).

Example of structure of the task definition:

```
{  
    "miner": 1000,  
    "name": "Miner name",  
    "antecedent": [  
        {"attribute": "attributeA", "fixedValue": "x"},  
        {"attribute": "attributeB"},  
        {"attribute": "attributeC"}  
    ],  
    "consequent": [  
        {"attribute": "attributeD"}  
    ],  
    "IMs": [  
        {"name": "confidence", "value": 0.5},  
        {"name": "support", "value": 0.01}  
    ],  
    "specialIMs": [  
        {"name": "cba"}  
    ],  
    "limitHits": 5000  
}
```

Output structure

The full results of the association rule mining tasks are available in forms of two XML based formats and in the really simple (basic) form of JSON output.

PMMI 4.3 – Association Rules

PMMI is technical standard supported by the majority of data mining system and tools. PMMI is a standard covering a large number of different data mining models. For the association rule mining the output model is called “Association Rules”. In the current version of PMMI standard (4.3) the Association Rules model can be newly based not only on transactional but also on tabular data, also it fully supports the task solvable in EasyMiner/R. The PMMI document consists of 3 main parts – *DataDictionary*, *TransformationDictionary* and the data mining model, in this case called *AssociationModel*.

The part *DataDictionary* contains the description of original data used for data mining – there is a list of data fields, optionally also with list of values and its frequencies.

The part *TransformationDictionary* contains description of applied preprocessing algorithms. For data mining of association rules, the data fields it is necessary to preprocess the data field into data attributes usable for data mining. The simplest preprocessing is a simple “copy” from data field to attributes, but the system supports also other preprocessing methods – described in previous paragraphs.

The part *AssociationModel* contains list of items, itemsets and list of association rules. Each rule contains a reference to definition of itemsets for its antecedent and consequent parts. Each itemset is a conjunction of one or more attributes with concrete values. Each rule contains also values of interest measures confidence, support and lift. The system EasyMiner complements these informations also with extensions with values from the four field table.

GUHA PMML

Extension of PMMI standard for support of more complex association rules. This output can describe both – simple association rules gained using algorithms Apriori or FP-Growth, also as complex association rules gained using procedure 4ft-Miner.

Similarly to the format PMMI Association Rules the output in GUHA PMML contains the full description of the data mining task. The GUHA PMML document consists of three main parts – *DataDictionary*, *TransformationDictionary* and *guha:AssociationModel*.

The parts *DataDictionary* and *TransformationDictionary* essentially match the same parts in the standard PMMI 4.3.

The part *guha:AssociationModel* contains the definition of data mining task and the founded results.

Basic JSON/XML output

For the purposes of simple exploration of task results the system EasyMiner supports simple export of rules in text form in combination with values from the four field table of the given association rule. Each founded association rules contains as properties the textual representation of the rule and values from the four field table (a, b, c, d). Other properties are id of the association rules in the system EasyMiner and the property “selected” (boolean property used for marking of interesting rules by the user).

For the representation of interest measures it is necessary to calculate its values from the values a, b, c and d using standard formulas for confidence, support and lift.

Example output:

```
{  
  "task": {  
    "id": 2530,  
    "miner": 357,  
    "name": "Example task",  
    "type": "cloud",  
    "state": "solved",  
    "importState": "done",  
    "rulesCount": 2,  
    "rulesOrder": "default"  
  },  
  "rules": [  
    {  
      "id": 1671253,  
      "text":  
        "competitiveness_of_smes((55;60]) →  
        technical_assistance((55;60))",  
      "a": 3, "b": 1, "c": 0, "d": 25,  
      "selected": "0"  
    },  
    {  
      "id": 1671254,  
      "text":  
        "low_carbon_economy((60;65]) → technical_assistance((55;60))",  
      "a": 3, "b": 1, "c": 0, "d": 25,  
      "selected": "0"  
    }  
  ]  
}
```

Visualization

For the visualization of the data mining results (founded association rules) is available a XSL transformation from GUHA PMML to HTML 5.

The visualization in the software stack of OpenBudgets project should be realized using the through the main integration using DAM. In the current version the integration script uses only the basic output in the JSON format.

From a user perspective, it is necessary to support at least a visualization with list of found rules with values of the basic interest measures (confidence, support, lift). The list of rules should be orderable using the values of the interest measures.

For a better understanding of the results the visualization should also support the display of the four field table. Beneficial would be the support of "search" in the results (for example display only rules containing the selected attribute).

3.5.4 Sample case

This sample case illustrates the use of data mining of association rules for data analysis of the list beneficiaries from all operational programs in program period 2007-2013 ([original dataset](#)).

Analyzed dataset

The list of beneficiaries was extended using data from the Czech Registry of economic entities (ARES). There was added information about the type of economic entities and about number of employees ([extended dataset](#)). In the preparation phase of the dataset for data mining, there were several changes:

- Truncated unnecessary text in string values - for better comprehensibility of association rules:
 - <http://data.openbudgets.eu/resource/dataset/esf-czech-projects/project/>
 - <http://linked.opendata.cz/resource/business-entity/>
 - <http://data.openbudgets.eu/codelist/cz-operational-programme/>
- Derived column "year" (derived from column "date")
- Derived column "numberOfEmployees" (string values with description of intervals defined using columns "numberOfEmployeesMin" and "numberOfEmployeesMax")
- Columns derived using calculations:
 - allocated2paidCz = allocatedCz/paidCz
 - allocated2paidEu = allocatedEu/paidEu,
 - certified2allocatedCz = certifiedCz/allocatedCz,
 - certified2allocatedEu = certifiedEu/allocatedEu,
 - certified2paidCz = certifiedCz/paidCz,
 - certified2paidEu = certifiedEu/paidEu

The above adjustments can be made using modification of the SPARQL query used for export data to CSV file. Other possibilities is using of SQL query in relational database or using of a tabular preprocessor (e.g. MS Excel).

Data preprocessing

As already described in the previous chapter ([3.5.3](#)), the algorithms for data mining of association rules can work only with nominal attributes. The continuous numerical values in data columns have to be preprocessed using intervals or named enumerations.

In the analyzed data, the used numerical columns (allocated2paidCz, allocated2paidEu, certified2allocatedCz, certified2allocatedEu, certified2paidCz, certified2paidEu) were preprocessed using equidistant intervals [0,05 ; 0,1), [0,1 ; 0,15)...

The numerical column year was preprocessed using the method each value - one bin (so the preprocessed attribute contains the same values like the used data column).

Preprocessed dataset description

The dataset has 107311 rows.

The attributes generated from data columns, used for data mining:

Attribute	Unique values count	Example value
allocated2paidCz_intervals	224	[1.00;1.05)
allocated2paid_Eu_intervals	227	[1.00;1.05)
certified2allocatedCz_intervals	71	[0.60;0.65)
certified2allocatedEu_intervals	71	[0.95;1.00)
certified2paidCz_intervals	61	[0.60;0.65)
certified2paidEu_intervals	64	[1.00;1.05)
numberOfEmployees	17	“[6-9]”
operationalProgramme	219	“3-2-2”
operationalProgrammeBroader	94	“3-2”
partnerTypeBroader	8	“Stát a jeho instituce a organizace”
year	10	“2014”

Table. 42. List of parameters of pre-processed input data

Data mining using EasyMiner API

For data mining using EasyMiner, there is available a complex RESTful API. In order to ensure the data security, all data using for data mining are connected with a specific user accounts. For usage of the API, the user has to have an existing user account. The user account is identified using API KEY send in all API requests – via GET variable called “apiKey={key}” or via a HTTP header “Authorization: ApiKey {key}”.

The API endpoint is described using swagger documentation available on the URL <http://{easyminer-server}/api>.

The data mining process using EasyMiner API is described on GitHub wiki page: <https://github.com/KIZI/EasyMiner-EasyMinerCenter/wiki/API-usage-manual>

In general, the data mining process should be:

1. Upload data to EasyMiner server
2. Initialize miner instance
3. Preprocess data columns to attributes
4. Define one or more data mining tasks and download the results
5. Delete preprocessed and original data from server

This data mining process was used on the described enriched ESF dataset. The following paragraphs contain a brief overview of two sample tasks and example of results interpretation.

Example data mining task

A definition of data mining task should be based on an analytical question. For example, the analytical question could be:

Is it possible to predict a ratio of allocated and paid money using the characteristics of a funded project partner, operational program and the year of funding?

The data mining task definition consists from a rule pattern and a list of required threshold of interest measures.

The consequent of the rule pattern should contain attributes certified2allocatedCz_intervals and certified2allocatedEu_intervals.

The antecedent of the rule should contain attributes numberOfEmployees, operationalProgramme, partnerTypeBroader and year.

The graphical UI is illustrated in Figure 28:

Association rule pattern

Antecedent	Interest measures	Consequent
numberOfEmployees (*) and operationalProgramme (*) and partnerTypeBroader (*) and year (*)	Confidence: 0.7 Support: 0.01 + Add interest measure	certified2allocatedCz_intervals (*) and certified2allocatedEu_intervals (*)

Figure 28. Graphical UI of association rule pattern in EasyMiner system

Using the API call, the task definition is:

```
{
  "miner": 1055,
  "name": "ESF miner",
  "antecedent": [
    {"attribute": "numberOfEmployees"},
    {"attribute": "operationalProgramme"},
    {"attribute": "partnerTypeBroader"},
    {"attribute": "year"}
  ],
  "consequent": [
    {"attribute": "certified2allocatedCz_intervals"},
    {"attribute": "certified2allocatedEu_intervals"}
  ],
  "IMs": [
    {"name": "confidence", "value": 0.7},
    {"name": "support", "value": 0.01}
  ],
}
```

```

    "limitHits": 1000
}

```

Using this task definition, the miner found 98 association rules, as shown in Figure 29:

`operationalProgramme(2-4-1) & year(2015) → certified2allocatedEu_intervals([0.00;0.05])`
 Confidence: 0.814 Support: 0.017

`operationalProgramme(2-4-1) & year(2015) → certified2allocatedCz_intervals([0.00;0.05])`
 Confidence: 0.814 Support: 0.017

`operationalProgramme(2-4-1) & year(2014) → certified2allocatedCz_intervals([1.00;1.05])`
 Confidence: 0.796 Support: 0.013

`operationalProgramme(2-4-1) & year(2014) → certified2allocatedEu_intervals([1.00;1.05])`
 Confidence: 0.792 Support: 0.013

`operationalProgramme(2-4-1) & partnerTypeBroader(Obce) & year(2015) → certified2allocatedEu_intervals([0.00;0.05])`
 Confidence: 0.802 Support: 0.013

`operationalProgramme(2-4-1) & partnerTypeBroader(Obce) & year(2015) → certified2allocatedCz_intervals([0.00;0.05])`
 Confidence: 0.802 Support: 0.013

`operationalProgramme(2-4-1) & partnerTypeBroader(Obce) & year(2014) → certified2allocatedCz_intervals([1.00;1.05])`
 Confidence: 0.805 Support: 0.01

Figure 29. Result of the rule-mining

Interpretation of results - association rules

The basic interpretation of simple association rules is the interpretation of the IF-THEN form: IF antecedent, THEN consequent. Consequently, it is necessary to take into account the interest measures used in the given data mining task.

Simple association rules found using the interest measure lift:

- `operationalProgramme(7-1-4) → allocated2paidCz_intervals([1.00;1.05])`
 Confidence: 0.987, Support: 0.052, Lift: 1.583
- `operationalProgramme(2-4-1) → allocated2paidEu_intervals([1.00;1.05])`
 Confidence: 0.973, Support: 0.054, Lift: 1.561
- `operationalProgramme(7-1-1) → certified2allocatedCz_intervals([0.00;0.05])`
 Confidence: 0.926, Support: 0.077, Lift: 3.521

How to interpret a rule?

`operationalProgramme(7-1-1) → certified2allocatedCz_intervals([0.00;0.05])`

Interest measure values

Interest Measure	Value
Support	0.0772
Confidence	0.9260
Lift	3.521

Four field contingency table

		Consequent	¬Consequent	
		Antecedent	8282	662
Antecedent	¬Antecedent	19938	78429	

Figure 30. Interpretation of a simple rule

- Confidence: If the operationalProgramme is “7-1-1”, then the value of certified2allocatedCz is in interval [0; 0.05) with confidence 92,6%.

- Lift: If the operationalProgramme is “7-1-1”, there is 3,521x higher probability, that the value of certified2allocatedCz in in interval [0; 0.05), than in the full dataset.

- Support: In the dataset, the rule is valid in 7,72% of rows.

When the end user is interested in concrete counts of rows (instances), the values can be read directly from the four field table. For example, the antecedent (operationalProgramme is “7-1-1”) is valid in 8944 rows, the full association rule is valid in 8282 rows.

Example of a bit longer association rule:

(numberOfEmployees([50-99]) & operationalProgramme(7-1-1)) & year(2015) → allocated2paidCz_intervals([1.00;1.05])

Interest measure values

Interest Measure	Value
Support	0.0178
Confidence	1.0000

Four field contingency table

		Consequent	¬Consequent	
		Antecedent	1913	0
Interest Measure	Value	¬Antecedent	64984	40414

Figure 31. Interpretation of a longer rule

If the numberOfEmployees is in the interval [50; 99], the operationalProgramme is “7-1-1” and the year is “2015”, then the value of allocated2paidCz is in the interval [1.00; 1.05), with the confidence 100%. The support of this association is 1,78, i.e. 1913 rows.

To understand all results, it is suitable to interpret not only individual association rules of each separately, but also analyze the combinations of them. For example, look on the following rules found using the [example task](#).

operationalProgramme(2-4-1) & year(2015) → certified2allocatedCz_intervals([0.00;0.05))
 Confidence: 0.814 | Support: 0.017

operationalProgramme(2-4-1) & year(2014) → certified2allocatedCz_intervals([1.00;1.05))
 Confidence: 0.796 | Support: 0.013

operationalProgramme(2-4-1) & partnerTypeBroader(Obce) & year(2015) → certified2allocatedCz_intervals([0.00;0.05))
 Confidence: 0.802 | Support: 0.013

Figure 32. Interpretation of several rules

About the first and second association rule:

If the operationalProgramme is “2-4-1”, then in the year “2015” is the value of certified2allocatedCz in interval [0 ; 0.05) with confidence 81,4%. But in the year “2014”, the value of certified2allocatedCz is in interval [1 ; 1.05) with confidence 79,6%.

About the first and third association rule:

If the partnerTypeBroader is “Obce” (in English is the value “municipality”), then in the case of operationalProgramme “2-4-1” and the year “2015” there is a bit lower probability that the value of certified2allocatedCz is in the interval [0 ; 0.05).

3.6 Outlier/anomaly detection

3.6.1 General description

Anomaly detection refers to the problem of finding patterns in data that do not conform to the expected normal behavior, i.e., Chandola et al. (2009). Different approaches share the same basic idea: to define certain *model* as normal cases, outliers are defined as those cases which do not fit the model. For example, we can define normal cases of heart beat rates²⁴ as follows: (1) for children 10 years and older, and adults (including seniors) is 60 - 100 beats per minute; (2) for well-trained athletes is 40 - 60 beats per minute. So, an adult whose heart beats 40 times per minute is abnormal, and should see the doctor. A well-trained athlete whose heart beats 100 times per minute is an abnormal case. This example introduces two important concepts in outlier-detection: subpopulation, and frequent pattern.

We developed two software packages for outlier-detection based on published papers (as no existing open-source libraries available), and two packages fulfil the requirements in D2.3 as follows. The Outlier-detection package fulfills six needs as listed in the following table.

Need	Description	Discussion	Task No.
N01	Filtering commensurable objects	Aggregate analytics can only operate on a pool of commensurable objects (i.e. objects with comparable “size”, in whatever terms).	T03
N04	Outlier detection	Outlier detection will be performed on the budget and spending data to find unusual values or patterns that may indicate errors in the data, irregular behavior like corruption or fraud, or point to regions/sectors of special interest.	T03
N14	Identify both good and bad examples of financial management	Another possible way <i>cf. rule/pattern mining for N14</i> to address this need is to apply outlier detection methods to identify unexpected behavior which may indicate a misspending of money (purposely or not, cf.	T12

24 http://www.heart.org/HEARTORG/HealthyLiving/PhysicalActivity/Target-Heart-Rates_UCM_434341_Article.jsp#.WHYq3JKBmnc

		(T03)) or on the other side an above-average financial management.	
N16	Identify systematic problems of project implementation in different funds and programmes, rather than in-depth engagement with individual projects	This need combines (N14) with performing aggregations on the data to achieve a general, systematic overview on the funds' and programmes' spending data, and incorporating the temporal dimension.	T14
N17	Consider fiscal indicators like error, performance and absorption rates	On top of that a separate outlier step might reveal deeper insights into the indicators.	T17
N29	Follow the state's money flows all the way down to transaction data and then questioning who was receiving the money and if this happened in a proper manner	To address this need, much effort is required on interlinking the budgets of the different government levels and the engaged ministries and councils (cf. requirement (R22)). Afterwards the resulting network will be analyzed using comparative statistics, graph analysis techniques, but also rule/pattern mining and outlier detection techniques (cf. (N14) and (N16)).	T28

Table 43.: Outlier-detection packages fulfil six requirements in D2.3

3.6.1.1 Local Outlier Factors based on Subpopulation

Whether an object is normal or abnormal within a group, largely relates to the way we define the group. A man with the height of 1.60 meter is abnormal in north Europe, but normal in the south of Asia. People die at the age of 40 is normal in some African countries, but abnormal in European countries. To identify abnormal data in a large dataset, we need to delineate groups to which this data belong. This introduces the subpopulation-based outlier detection method, i.e. Fleischhacker, et al. (2014). An open source data-mining tool has been developed following the methods described by Fleischhacker, et al. (2014), and downloadable at https://github.com/openbudgets/outlier_dm.

Generating possible constraints

Constraints can be set to the class or to the property. The descriptions “children 10 years and older, adults, well-trained athletes” in the heart-beating rate example contains classes as constraints: “children”, “adult”, and “athlete”, properties as constraints: “well-trained”, property values as constraints: “10 years and older”. Based on OBEU data model described in Klímek (2015a) and Klímek (2015b), constraints for OBEU dataset are limited to property values, such as “year = 2009”, “budgetPhase=approved”, “administrativeClassification = <value>”,

"economicClassification = <value>", "functionalClassification = <value>". Given one or more datasets, we first collect all these property-value pairs, as basic constraints.

Finding subpopulations

Each basic constraint partitions the whole set of measures into two subsets -- either satisfying the constraint, or not. Adding a new constraint to a subset might split it into two. Using the generated constraints, we can generate a lattice of subpopulations. Each node corresponds to a set of satisfied constraints (a one to one mapping). A child node satisfies more constraints than the parent nodes, as illustrated in Figure 33.

Figure 33. A lattice of subpopulation.

We can see that data-items in a node can be empty, if contradictory constraints are applied. It can also happen either the number of data-items are very small, or shrink very small compared to its parent node. For the former case, the number of data-items is not sufficient to define normal cases; for the latter case, the new constraint does not contain sufficient information to separate data-items in the parent node, which follows repeated computation for the parent node and the child node. For each case, we define some threshold values following Fleischhacker, et al. (2014): 'min_population_size': nodes with data items less than that value will be pruned; 'reduction_factor': the ration of data items in a child node to that of the parent node shall be less than this factor, otherwise, the constraint will not be applied.

Outlier detection within a subpopulation and outlier scores

One important perspective for outlier-detection, which has not been covered in other data-mining sub-tasks of this working package, is the method of LOF (Local Outlier Factor) by Breunig, et al. (2000). The basic idea of LOF is based on *local density*. The locality of a point A is represented by its k nearest neighbors. The density of A is estimated by the distance between A and these k neighbors. Points with similar density are grouped and form a region. Outliers are such points that their densities are significantly lower than their neighbors, as illustrated in Figure 34.

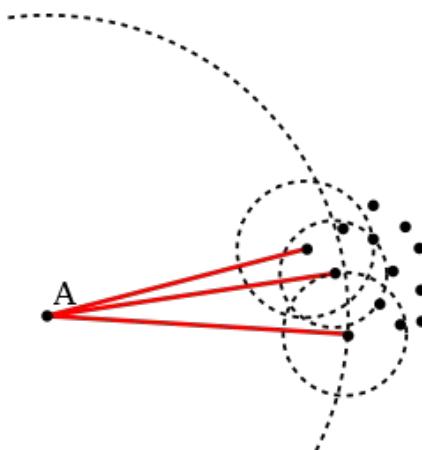


Figure. 34.. The density of A is much lower than densities of its neighbors, so A is an outlier

How can we estimate the density of A by the distance between A and these k neighbors? The distance from your office chair to your office door is 5 steps, which means that if you sit in your chair, you can reach to your office door in 5 steps. The distance from A to its k neighbors is 5 steps, which means that each neighbor of A can be reached within 5 steps. **k-distance(A)** is defined as the distance between A and its k-th nearest neighbor. **$N_k(A)$** is all the objects less than **k-distance(A)** to A. **$N_k(A)$** is called *the set of k nearest neighbors of A*. **$N_k(A)$** can be larger than k, if more than one object is exactly k-distance(A) away from A. The *reachability distance* from A to B is defined as k-distance(B), if A is in $N_k(B)$, or the distance between A and B.

$$\text{reachability-distance}_k(A,B) = \max\{k\text{-distance}(B), d(A,B)\}$$

Let A_1, A_2, \dots, A_k be k nearest neighbors of B, *reachability distance* from A to B can be rewritten as follows.

$$\text{reachability-distance}_k(A,B) = \max\{ d(A_1,B), d(A_2,B), \dots, d(A_k,B), d(A,B) \}$$

If A is among { A_1, A_2, \dots, A_k }, $\text{reachability-distance}_k(A,B) = k\text{-distance}(B)$, otherwise $\text{reachability-distance}_k(A,B) = d(A,B)$.

Let $k = 1$, A_1 be your office door, A be the public mobile phone on the floor (imagine there is a public mobile phone for all office members on the floor). If the phone is on your desk, the reachability distance will be the distance between you and the office door; if the phone is not in your office, the reachability distance will be the trace length from you to the mobile phone. Let $k = 2$, A_1, A_2 be office doors. If the phone is on your desk, the reachability distance will be the distance from you and the second nearest office door (the most nearest office should be your office door).

Intuitively, the density of an object is inversely proportional to distances to its neighbors. Using k-neighbors observation, the density of object A shall be inversely proportional to the reachability-distance from its neighbor. If A has more than one neighbor, we calculate the average value. Formally the local reachability density of A is defined as follows.

$$\text{lrd}(A) := 1 / \left(\frac{\sum_{B \in N_k(A)} \text{reachability-distance}_k(A, B)}{|N_k(A)|} \right)$$

$\text{lrd}(A)$ is inversely proportional to the average of the reachability distance from its k neighbor to A . The longer distance that A 's nearest neighbors take to reach A , the sparser of the density at A . To identify density-based outlier, we need to compare the density of A with densities of A 's neighbors. In the literature, this comparison is computed as the average ratio of lrd of A 's neighbor to lrd of A , $\text{LOF}_k(A)$, formally defined as follows.

$$\text{LOF}_k(A) := \frac{\sum_{B \in N_k(A)} \frac{\text{lrd}(B)}{\text{lrd}(A)}}{|N_k(A)|} = \frac{\sum_{B \in N_k(A)} \text{lrd}(B)}{|N_k(A)|} / \text{lrd}(A)$$

Outlier score and its interpretation

If $\text{LOF}_k(A)$ approximately equals to 1, A and its nearest k neighbors have the same local density. $\text{LOF}_k(A)$ below 1 means that the average $\text{lrd}(B)$ is less than $\text{lrd}(A)$. This can be interpreted as the average reachability distance of B is greater than the reachability distance of A . So, A is located in a denser region than B . $\text{LOF}_k(A)$ much larger than 1 means A is located in a rather sparse region, isolated from its neighbors. This indicates that A is an outlier, and $\text{LOF}_k(A)$ is call the outlier score, as illustrated in Figure 35.

In a lattice of sub-populations, an object may appear in several sub-populations. In each subpopulation, it has an outlier score. Its overall outlier score in the whole lattice can be computed differently. One simple way is to use the average value of all outlier scores in the subpopulations. More complicated way needs to analyze the constraints of an subpopulation, which may strengthen or weaken the influence its outlier score to the overall score. As a result, the overall outlier score will be the average of the weighted sum of all outlier scores in subpopulations.

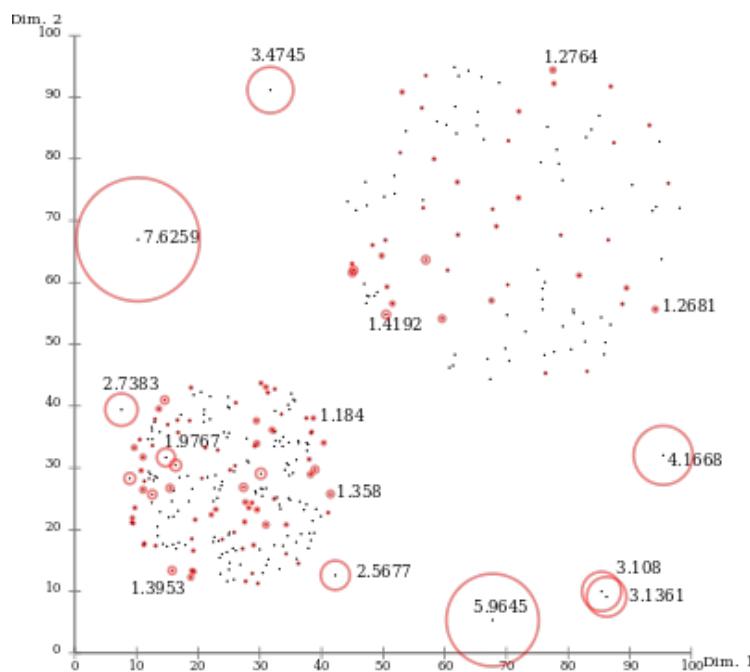


Figure 35. Data-items are clustered based on ratio of densities

The term ‘density’ is a concept in the spatial domain. Its paired concept in the temporal domain is ‘frequency’.

3.6.1.2 Frequent patterns

Description of the algorithm that uses a frequent pattern mining as an underlying concept. The main idea is: if an instance contains more frequent patterns, it means that this data instance is unlikely to be an anomaly. The algorithm is unsupervised and there is therefore no requirement for any training data with ground truth information. It can also work with a mixture of data types including binary, categorical or numeric variables where numeric values should be discretized. It is also focused on situations with multiple attributes (multivariate). One of the main motivations for an approach based on the frequent pattern mining is that we are not interested only in unusual amounts, but also in combinations with other descriptions of the data instance. We cannot rely only on statistical methods that analyse extreme values. Those extreme values can be dependent on associated contextual information such as the category of the spending.

Anomaly score computation

Algorithms that build on top of frequent patterns utilizes well known approaches for the frequent pattern mining, such as Apriori or FP-Growth. The main limitation is that they do not fully support numeric values and a discretization step is required as a preprocessing of the data.

There are several existing approaches in research that use mined frequent patterns to compute anomaly scores. We provide implementations for FPOF (He, et al. (2005)) that is

often considered as a baseline approach. It computes the anomaly score using only the availability of pattern in the instance together with its frequency. There are several known limitations including the issue with using of duplications coming from subsets or supersets of a frequent pattern. Other algorithms try to overcome those limitations with modifications of the anomaly score formula. Algorithm LFPOF/EFPOR (Zhang, et al.(2010)) decreases the influence of duplications using only the longest frequent patterns, MFPOF (Feng, et al. (2010)) uses maximal frequent patterns or WCFPOF (Jiadong, et al. (2009)) utilizes only closed frequent patterns. Algorithm FPCOF (Tang, et al.(2009)) measures how contradictory the existing patterns are and WFPOF (Zhou, et al. (2007)) extends the computation of the FPOF score about the influence of the length of pattern in contrast to the size of the data instance. Implementations of existing approaches are available in our R package²⁵.

We also extend the set of algorithms about a new formula that we call FPI (Frequent Pattern Isolation). It is inspired by an existing algorithm called Isolation Forests (IF). The algorithm starts with a mining of all frequent patterns that meets the standard predefined criteria: minimum relative support and maximal length of the pattern (a number of items in the pattern). The anomaly score is computed using following principle: the matching patterns that are more frequent (with higher support) and contains more items (higher length) produce significantly lower score than less frequent (lower support) and shorter patterns. If the data instance contains short but infrequent patterns, it is likely to be an anomaly. FPI includes also penalizations for situations when the limited amount of patterns is available and the data instance is not completely covered with the existing set of frequent patterns.

3.6.1.3 Financial ratios

It is available only as a dedicated experiment. Currently, there is no generic implementation.

3.6.2 input & output

3.6.2.1 Local Outlier Factors based on Subpopulation

User input

Users can select one or more RDF datasets as the input of the LOF-subpopulation-based outlier-detection. After the data-mining task is triggered, the names of the selected datasets will be sent to the DAM server at the backend.

Pre-processing of input

The outlier-detection algorithm accepts CSV file as input. When users select one or more RDF files, a pre-processing procedure must be carried out to extract the content from the select RDF files, and to integrate them into one CSV file.

25 <https://github.com/jaroslav-kuchar/fpmoutliers>

Output structure

The main procedure of LOF-subpopulation-based outlier-detection is to construct a lattice of subpopulations, compute outlier-score for each data-item in each subpopulation, and summarize an overall outlier-score for each data-item.

The output structure is one single CSV file, which is a list data-items with their overall outlier-scores, sorted in the decrease order.

3.6.2.2 Frequent patterns

User input

Data in tabular format (e.g. CSV file). Minimum support as a parameter of an underlying frequent pattern mining algorithm.

Pre-processing of input

Since frequent pattern mining algorithms do not interpret numeric values, all numeric attributes have to preprocessed using discretization algorithms (e.g. equal-width, equal frequency, clustering binning, ...)

Output structure

The algorithm assigns a numeric value (anomaly score) to each input instance/observation. Higher anomaly scores are assigned to instances with higher tendency to be anomalous.

Basic example of the R package:

```
#                                     load          library
library(fpmoutliers)
#           load      input      data      as      a      data.frame
dataFrame <- read.csv(system.file("extdata", "fp-outlier-customer-data.csv", package =
=                                         "fpmoutliers"))
#           compute      the      OD      model      and      scores
model     <- FPI(dataFrame, minSupport = 0.001)
#           print      scores      for      all      input      instances
print(model$scores)
```

3.6.2.3 Financial ratios

It is available only as a dedicated experiment. Currently, there is no generic implementation.

3.6.3 Sample case

3.6.3.1 Local Outlier Factors based on Subpopulation

User input

Suppose a user selects three RDF file names: budget-kilkis-expenditure-2012, budget-kilkis-expenditure-2013, and budget-kilkis-expenditure-2014, and chooses the LOF outlier-detection data-mining service. He specifies that a subpopulation must contain 30 items, and

he would like to have 25 top-outliers -- that is, their outlier-scores are ranked at top 25, as illustrated in Figure 36²⁶

Figure 36.: A sample user interface for subpopulation-based LOF outlier-detection

Pre-processing and the input to the core algorithm

The select three files are in the RDF format. A sparql query is automatically generated to extract the data items from the three files and construct a csv file, as illustrated in Figure 37, Figure 38.

```

PREFIX qb: <http://purl.org/linked-data/cube#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX gr-dimension: <http://data.openbudgets.eu/ontology/dsd/greek-municipalities/dimension/>
SELECT
  (MIN(?observation) AS ?ID)
  (SUM(?amount2) AS ?amount)
?economicClass
?adminClass
?year
?budgetPhase
FROM <http://data.openbudgets.eu/resource/dataset/budget-kilkis-expenditure-2012>
FROM <http://data.openbudgets.eu/resource/dataset/budget-kilkis-expenditure-2013>
FROM <http://data.openbudgets.eu/resource/dataset/budget-kilkis-expenditure-2014>
WHERE { ?slice qb:observation ?observation .
?observation obeu-measure:amount ?amount2 .
?slice ?economicClassification ?economicClass . filter(contains(str(?economicClassification), "economicClassification")) .
?slice ?administrativeClassification ?adminClass . filter(contains(str(?administrativeClassification), "administrativeClassification")) ?
?observation qb:dataSet/obeu-dimension:fiscalYear ?year .
?observation gr-dimension:budgetPhase ?budgetPhase . }
GROUP BY ?economicClass ?adminClass ?year ?budgetPhase
LIMIT 10000

```

Figure 37.: Automatically generated Sparql query to extract data items from selected RDF files

²⁶The user interface shown here is only for the testing case. A unified user interface for all data-mining tasks are being developed in Work Package 3.

	Result_top25.csv	2017-01-16_09-49-55-454449.csv
1	"ID","amount","economicClass","adminClass","year","budgetPhase"	
2	,target,nominal,nominal,nominal	
3	"http://data.openbudgets.eu/resource/dataset/budget-kilkis-expenditure-2013/observation/35.6011/revised",160600,"http://data.openbudgets.eu/resource/codelist/kae-ota-exodwn-2014/6011","http://data.openbudgets.eu/resource/codelist/kae-ota-administration-2014/35","http://reference.data.gov.uk/id/year/2013","http://data.openbudgets.eu/resource/codelist/budget-phase/revised"	
4	"http://data.openbudgets.eu/resource/dataset/budget-kilkis-expenditure-2012/observation/25.6414/executed",0,"http://data.openbudgets.eu/resource/codelist/kae-ota-exodwn-2014/6414","http://data.openbudgets.eu/resource/codelist/kae-ota-administration-2014/25","http://reference.data.gov.uk/id/year/2012","http://data.openbudgets.eu/resource/codelist/budget-phase/executed"	
5	"http://data.openbudgets.eu/resource/dataset/budget-kilkis-expenditure-2014/observation/30.7411.0001/reserved",0,"http://data.openbudgets.eu/resource/codelist/kae-ota-exodwn-2014/7411","http://data.openbudgets.eu/resource/codelist/kae-ota-administration-2014/30","http://reference.data.gov.uk/id/year/2014","http://data.openbudgets.eu/resource/dataset/greek-municipalities/codelist/budget-phase/reserved"	
6	"http://data.openbudgets.eu/resource/dataset/budget-kilkis-expenditure-2013/observation/00.6132/revised",600,"http://data.openbudgets.eu/resource/codelist/kae-ota-exodwn-2014/6132","http://data.openbudgets.eu/resource/codelist/kae-ota-administration-2014/00","http://reference.data.gov.uk/id/year/2013","http://data.openbudgets.eu/resource/codelist/budget-phase/revised"	
7	"http://data.openbudgets.eu/resource/dataset/budget-kilkis-expenditure-2013/observation/40.7311.0000/revised",0,"http://data.openbudgets.eu/resource/codelist/kae-ota-exodwn-2014/7311","http://data.openbudgets.eu/resource/codelist/kae-ota-administration-2014/40","http://reference.data.gov.uk/id/year/2013","http://data.openbudgets.eu/resource/codelist/budget-phase/revised"	
8	"http://data.openbudgets.eu/resource/dataset/budget-kilkis-expenditure-2014/observation/35.6612/approved",0,"http://data.openbudgets.eu/resource/codelist/kae-ota-exodwn-2014/6612","http://data.openbudgets.eu/resource/codelist/kae-ota-administration-2014/35","http://reference.data.gov.uk/id/year/2014","http://data.openbudgets.eu/resource/codelist/budget-phase/approved"	
9	"http://data.openbudgets.eu/resource/dataset/budget-kilkis-expenditure-2013/observation/10.6072/approved",0,"http://data.openbudgets.eu/resource/codelist/kae-ota-exodwn-2014/6072","http://data.openbudgets.eu/resource/codelist/kae-ota-administration-2014/10","http://reference.data.gov.uk/id/year/2013","http://data.openbudgets.eu/resource/codelist/budget-phase/approved"	
10	"http://data.openbudgets.eu/resource/dataset/budget-kilkis-expenditure-2014/observation/70.7513.0000/revised",0,"http://data.openbudgets.eu/resource/codelist/kae-ota-exodwn-2014/7513","http://data.openbudgets.eu/resource/codelist/kae-ota-administration-2014/70","http://reference.data.gov.uk/id/year/2014","http://data.openbudgets.eu/resource/codelist/budget-phase/revised"	
11	"http://data.openbudgets.eu/resource/dataset/budget-kilkis-expenditure-2014/observation/30.7134.0004/executed",8856,"http://data.openbudgets.eu/resource/codelist/kae-ota-exodwn-2014/7134","http://data.openbudgets.eu/resource/codelist/kae-ota-administration-2014/30","http://reference.data.gov.uk/id/year/2014","http://data.openbudgets.eu/resource/codelist/budget-phase/executed"	
12	"http://data.openbudgets.eu/resource/dataset/budget-kilkis-expenditure-2012/observation/10.6641/approved",14900.27,"http://data.openbudgets.eu/resource/codelist/kae-ota-exodwn-2014/6641","http://data.openbudgets.eu/resource/codelist/kae-ota-administration-2014/10","http://reference.data.gov.uk/id/year/2012","http://data.openbudgets.eu/resource/codelist/budget-phase/approved"	
13	"http://data.openbudgets.eu/resource/dataset/budget-kilkis-expenditure-2012/observation/45.6245/revised",0,"http://data.openbudgets.eu/resource/codelist/kae-ota-exodwn-2014/6245","http://data.openbudgets.eu/resource/codelist/kae-ota-administration-2014/45","http://reference.data.gov.uk/id/year/2012","http://data.openbudgets.eu/resource/codelist/budget-phase/revised"	
14	"http://data.openbudgets.eu/resource/dataset/budget-kilkis-expenditure-2013/observation/45.6312/executed",0,"http://data.openbudgets.eu/resource/codelist/kae-ota-exodwn-2014/6312","http://data.openbudgets.eu/resource/codelist/kae-ota-administration-2014/45","http://reference.data.gov.uk/id/year/2013","http://data.openbudgets.eu/resource/codelist/budget-phase/executed"	
15	"http://data.openbudgets.eu/resource/dataset/budget-kilkis-expenditure-2012/observation/50.6322/reserved",0,"http://data.openbudgets.eu/resource/codelist/kae-ota-exodwn-2014/6322","http://data.openbudgets.eu/resource/codelist/kae-ota-administration-2014/50","http://reference.data.gov.uk/id/year/2012","http://data.openbudgets.eu/resource/dataset/greek-municipalities/codelist/budget-phase/reserved"	
16	"http://data.openbudgets.eu/resource/dataset/budget-kilkis-expenditure-2013/observation/10.6264.0001/reserved",601.09,"http://data.openbudgets.eu/resource/codelist/kae-ota-exodwn-2014/6264","http://data.openbudgets.eu/resource/codelist/kae-ota-administration-2014/10","http://reference.data.gov.uk/id/year/2013","http://data.openbudgets.eu/resource/dataset/greek-municipalities/codelist/budget-phase/reserved"	
17	"http://data.openbudgets.eu/resource/dataset/budget-kilkis-expenditure-2012/observation/10.6232.0001/executed",12672,"http://data.openbudgets.eu/resource/codelist/kae-ota-exodwn-2014/6232","http://data.openbudgets.eu/resource/codelist/kae-ota-administration-2014/10","http://reference.data.gov.uk/id/year/2012","http://data.openbudgets.eu/resource/codelist/budget-phase/executed"	
18	"http://data.openbudgets.eu/resource/dataset/budget-kilkis-expenditure-2012/observation/20.7422.0000/reserved",0,"http://data.openbudgets.eu/resource/codelist/kae-ota-exodwn-2014/7422","http://data.openbudgets.eu/resource/codelist/kae-ota-administration-2014/20","http://reference.data.gov.uk/id/year/2012","http://data.openbudgets.eu/resource/codelist/budget-phase/revised"	

Figure 38.: A CSV file is automatically generated for the input of the algorithm

Output of the core algorithm

The LOF outlier-detection algorithm will compute the average outlier score for each data items in subpopulations, and select top 25 data items based on outlier scores, as below.

Outlier score,	item id
(2666688.9775540209,	item 5606)
(1733647.1235530956,	item 9104)
(1656253.1099827976,	item 745)
(1039410.8461285697,	item 9546)
(848755.18378324737,	item 4068)
(702023.87435262429,	item 1302)
(657773.0874665142,	item 835)
(642069.34324842691,	item 8762)
(583268.55018808821,	item 186)
(582848.07107327459,	item 1104)
(549368.7929624652,	item 6300)
(549177.89014284546,	item 5184)
(525719.58216976351,	item 8415)
(504376.91300859861,	item 9637)
(424449.9335763898,	item 3616)
(424107.16816790245,	item 5604)
(422212.10605806066,	item 3886)

(349694.91562271549, item 3048)
 (333275.3656219258, item 697)
 (322196.16402656323, item 4016)
 (320012.51162248827, item 6397)
 (309400.83711114962, item 1511)
 (308184.70558750362, item 7767)
 (295287.57137482765, item 7466)
 (295046.08235537738, item 2778)

The selected 25 outliers will be saved in a csv file, as illustrated in Figure 39.

	Result_top25.csv	2017-01-16_09-49-55-454449.csv
1	Item,Features,,,Target,Score	
2	,,""economicClass","",""adminClass","",""year","",""budgetPhase","","	
3	5606,""http://data.openbudgets.eu/resource/codelist/kae-ota-exodvn-2014/6213""",""http://data.openbudgets.eu/resource/codelist/kae-ota-administration-2014/15""",,""http://reference.data.gov.uk/id/year/2012""",,""http://data.openbudgets.eu/resource/codelist/budget-phase/draft""",,120000.0,2666688.97755	
4	9104,""http://data.openbudgets.eu/resource/codelist/kae-ota-exodvn-2014/7323""",""http://data.openbudgets.eu/resource/codelist/kae-ota-administration-2014/30""",,""http://reference.data.gov.uk/id/year/2013""",,""http://data.openbudgets.eu/resource/codelist/budget-phase/draft""",,1034425.16,1733647.12355	
5	745,""http://data.openbudgets.eu/resource/codelist/kae-ota-exodvn-2014/7323""",""http://data.openbudgets.eu/resource/codelist/kae-ota-administration-2014/30""",,""http://reference.data.gov.uk/id/year/2014""",,""http://data.openbudgets.eu/resource/codelist/budget-phase/draft""",,983207.79,1656253.10999	
6	9545,""http://data.openbudgets.eu/resource/codelist/kae-ota-exodvn-2014/7323""",""http://data.openbudgets.eu/resource/codelist/kae-ota-administration-2014/30""",,""http://reference.data.gov.uk/id/year/2012""",,""http://data.openbudgets.eu/resource/codelist/budget-phase/draft""",,5574642.28,1839410.84613	
7	4068,""http://data.openbudgets.eu/resource/codelist/kae-ota-exodvn-2014/7323""",""http://data.openbudgets.eu/resource/codelist/kae-ota-administration-2014/30""",,""http://reference.data.gov.uk/id/year/2014""",,""http://data.openbudgets.eu/resource/dataset/greek-municipalities/codelist/budget-phase/reserved""",,427838.65,848755.183783	
8	1302,""http://data.openbudgets.eu/resource/codelist/kae-ota-exodvn-2014/6471""",""http://data.openbudgets.eu/resource/codelist/kae-ota-administration-2014/15""",,""http://reference.data.gov.uk/id/year/2014""",,""http://data.openbudgets.eu/resource/codelist/budget-phase/revised""",,107790.0,642069.343248	
9	835,""http://data.openbudgets.eu/resource/codelist/kae-ota-exodvn-2014/6471""",""http://data.openbudgets.eu/resource/codelist/kae-ota-administration-2014/15""",,""http://reference.data.gov.uk/id/year/2014""",,""http://data.openbudgets.eu/resource/codelist/budget-phase/draft""",,186250.22,657773.087467	
10	8762,""http://data.openbudgets.eu/resource/codelist/kae-ota-exodvn-2014/7326""",""http://data.openbudgets.eu/resource/codelist/kae-ota-administration-2014/45""",,""http://reference.data.gov.uk/id/year/2013""",,""http://data.openbudgets.eu/resource/codelist/budget-phase/revised""",,107790.0,642069.343248	
11	186,""http://data.openbudgets.eu/resource/codelist/kae-ota-exodvn-2014/7413""",""http://data.openbudgets.eu/resource/codelist/kae-ota-administration-2014/30""",,""http://reference.data.gov.uk/id/year/2013""",,""http://data.openbudgets.eu/resource/codelist/budget-phase/revised""",,314395.75,583268.59188	
12	1104,""http://data.openbudgets.eu/resource/codelist/kae-ota-exodvn-2014/7326""",""http://data.openbudgets.eu/resource/codelist/kae-ota-administration-2014/45""",,""http://reference.data.gov.uk/id/year/2012""",,""http://data.openbudgets.eu/resource/codelist/budget-phase/draft""",,95800.0,582848.071073	
13	6300,""http://data.openbudgets.eu/resource/codelist/kae-ota-exodvn-2014/7413""",""http://data.openbudgets.eu/resource/codelist/kae-ota-administration-2014/30""",,""http://reference.data.gov.uk/id/year/2013""",,""http://data.openbudgets.eu/resource/codelist/budget-phase/draft""",,29454.74,549368.792962	
14	5184,""http://data.openbudgets.eu/resource/codelist/kae-ota-exodvn-2014/7336""",""http://data.openbudgets.eu/resource/codelist/kae-ota-administration-2014/45""",,""http://reference.data.gov.uk/id/year/2014""",,""http://data.openbudgets.eu/resource/codelist/budget-phase/revised""",,152680.51,549177.890143	
15	8415,""http://data.openbudgets.eu/resource/codelist/kae-ota-exodvn-2014/6011""",""http://data.openbudgets.eu/resource/codelist/kae-ota-administration-2014/40""",,""http://reference.data.gov.uk/id/year/2012""",,""http://data.openbudgets.eu/resource/codelist/budget-phase/revised""",,280800.0,525719.58217	
16	9637,""http://data.openbudgets.eu/resource/codelist/kae-ota-exodvn-2014/7323""",""http://data.openbudgets.eu/resource/codelist/kae-ota-administration-2014/30""",,""http://reference.data.gov.uk/id/year/2013""",,""http://data.openbudgets.eu/resource/codelist/budget-phase/approved""",,228971.26,504376.913089	
17	3616,""http://data.openbudgets.eu/resource/codelist/kae-ota-exodvn-2014/7326""",""http://data.openbudgets.eu/resource/codelist/kae-ota-administration-2014/45""",,""http://reference.data.gov.uk/id/year/2012""",,""http://data.openbudgets.eu/resource/codelist/budget-phase/executed""",,69700.05,424400.933576	
18	5604,""http://data.openbudgets.eu/resource/codelist/kae-ota-exodvn-2014/7326""",""http://data.openbudgets.eu/resource/codelist/kae-ota-administration-2014/45""",,""http://reference.data.gov.uk/id/year/2012""",,""http://data.openbudgets.eu/resource/codelist/budget-phase/approved""",,69700.05,424107.168168	
19	3886,""http://data.openbudgets.eu/resource/codelist/kae-ota-exodvn-2014/7331""",""http://data.openbudgets.eu/resource/codelist/kae-ota-administration-2014/30""",,""http://reference.data.gov.uk/id/year/2012""",,""http://data.openbudgets.eu/resource/codelist/budget-phase/draft""",,245531.92,422212.106058	
20	3848,""http://data.openbudgets.eu/resource/codelist/kae-ota-exodvn-2014/6471""",""http://data.openbudgets.eu/resource/codelist/kae-ota-administration-2014/15""",,""http://reference.data.gov.uk/id/year/2013""",,""http://data.openbudgets.eu/resource/dataset/greek-municipalities/codelist/budget-phase/reserved""",,190092.21,349694.915623	
21	697,""http://data.openbudgets.eu/resource/codelist/kae-ota-exodvn-2014/7326""",""http://data.openbudgets.eu/resource/codelist/kae-ota-administration-2014/45""",,""http://reference.data.gov.uk/id/year/2013""",,""http://data.openbudgets.eu/resource/codelist/budget-phase/executed""",,53734.22,333275.365622	
22	4016,""http://data.openbudgets.eu/resource/codelist/kae-ota-exodvn-2014/6471""",""http://data.openbudgets.eu/resource/codelist/budget-phase/executed""",,179699.94,322196.164027	
23	6397,""http://data.openbudgets.eu/resource/codelist/kae-ota-exodvn-2014/6011""",""http://data.openbudgets.eu/resource/codelist/kae-ota-administration-2014/70""",,""http://reference.data.gov.uk/id/year/2013""",,""http://data.openbudgets.eu/resource/codelist/budget-phase/draft""",,144000.0,320012.511622	
24	1511,""http://data.openbudgets.eu/resource/codelist/kae-ota-exodvn-2014/6011""",""http://data.openbudgets.eu/resource/codelist/kae-ota-administration-2014/70""",,""http://reference.data.gov.uk/id/year/2014""",,""http://data.openbudgets.eu/resource/codelist/budget-phase/approved""",,165309.01,309400.837111	
25	7767,""http://data.openbudgets.eu/resource/codelist/kae-ota-exodvn-2014/6021""",""http://data.openbudgets.eu/resource/codelist/kae-ota-administration-2014/15""",,""http://reference.data.gov.uk/id/year/2014""",,""http://data.openbudgets.eu/resource/codelist/budget-phase/draft""",,174900.0,308184.705588	
26	7461,""http://data.openbudgets.eu/resource/codelist/kae-ota-exodvn-2014/6011""",""http://data.openbudgets.eu/resource/codelist/kae-ota-administration-2014/35""",,""http://reference.data.gov.uk/id/year/2013""",,""http://data.openbudgets.eu/resource/codelist/budget-phase/approved""",,159235.62,295287.571375	
27	2778,""http://data.openbudgets.eu/resource/codelist/kae-ota-exodvn-2014/6011""",""http://data.openbudgets.eu/resource/codelist/kae-ota-administration-2014/35""",,""http://reference.data.gov.uk/id/year/2014""",,""http://data.openbudgets.eu/resource/codelist/budget-phase/executed""",,148712.31,295046.002355	

Figure 39. The top 25 outlier data-items are saved in a csv file

3.6.3.2 Frequent patterns

Experiment on OBEU dataset with examples of results and preliminary explanations/visualizations. We selected data about Czech European Social Funds (ESF-CZ-2007-2013). It contains information about projects with additional attributes (e.g., partner, partner type or operational programme) and amounts of assigned money (The dataset contains 107311 instances). The goal is to reveal instances in data that deviates from others. We experimentally selected the following subset of three attributes: partnerTypeBroader, operationalProgrammeBroader, and certifiedEu. We use FPI algorithm with the following

setting: minimumSupport =0.0001 and maximumLength of patterns is not limited. Amount values are discretized to 1000 equal lengths intervals. FPI produced anomaly scores from 3.2 to 71553 (with 1245 frequent patterns).

Example of outputs:

- Regular Instance (amount=[0, 7.45 millions]):
- Matching: 7 patterns (3 of 3 attributes = 100%)
- Patterns

	(support) :
•	{ amount = [0 . 0 0 e + 0 0 , 7 . 4 5 e + 0 6) } (0 . 8 5)
•	{ partnerTypeBroader=Other } (0 . 2 1)
•	{ partnerTypeBroader=Other, amount = [0 . 0 0 e + 0 0 , 7 . 4 5 e + 0 6) } (0 . 2)
•	{ operationalProgrammeBroader=7-1 } (0 . 1 8)
•	{ operationalProgrammeBroader=7-1, amount = [0 . 0 0 e + 0 0 , 7 . 4 5 e + 0 6) } (0 . 1 7)
•	{ partnerTypeBroader=Other, operationalProgrammeBroader=7-1 } (0 . 1 5)
•	{ partnerTypeBroader=Other, operationalProgrammeBroader=7-1, amount = [0 . 0 0 e + 0 0 , 7 . 4 5 e + 0 6) } (0 . 1 5)
- Anomaly score: 3.2
- Computed as: mean({1/(0.85 * 1),1/(0.21 * 1), 1/(0.2 * 2), 1/(0.18 * 1), 1/(0.17 * 2), 1/(0.15 * 2, 1/(0.15 * 3))})
- Anomaly Instance (amount=3.34 billions):
- Matching: 1 pattern (1 of 3 attributes = 33.3%)
- Patterns

	(support) :
•	{partnerTypeBroader= Educational and research Institution} (0.028)
- Anomaly score: 71552.57
- Computed as: mean({1/(0.028 *1), 107311, 107311})

The instance with the lowest score is matched with 7 frequent patterns that cover all attributes of the instance and there is no need for any penalization. The instance with the highest anomaly score is matched with only one frequent pattern that covers only one attribute with significantly lower support. The two remaining attributes (infrequent operational programme together with infrequent amount about 3.34 billions) are penalised. FPI algorithm is able to detect instances of the data that are composed from the frequent values and associates them with the low anomaly scores (Figure 41). For instances that contain infrequent values the significantly higher anomaly is assigned (Figure 40). The drawback of the method is that numeric values are treated as categorical values with the same effect as other values and in this specific use case, amount plays a special role.

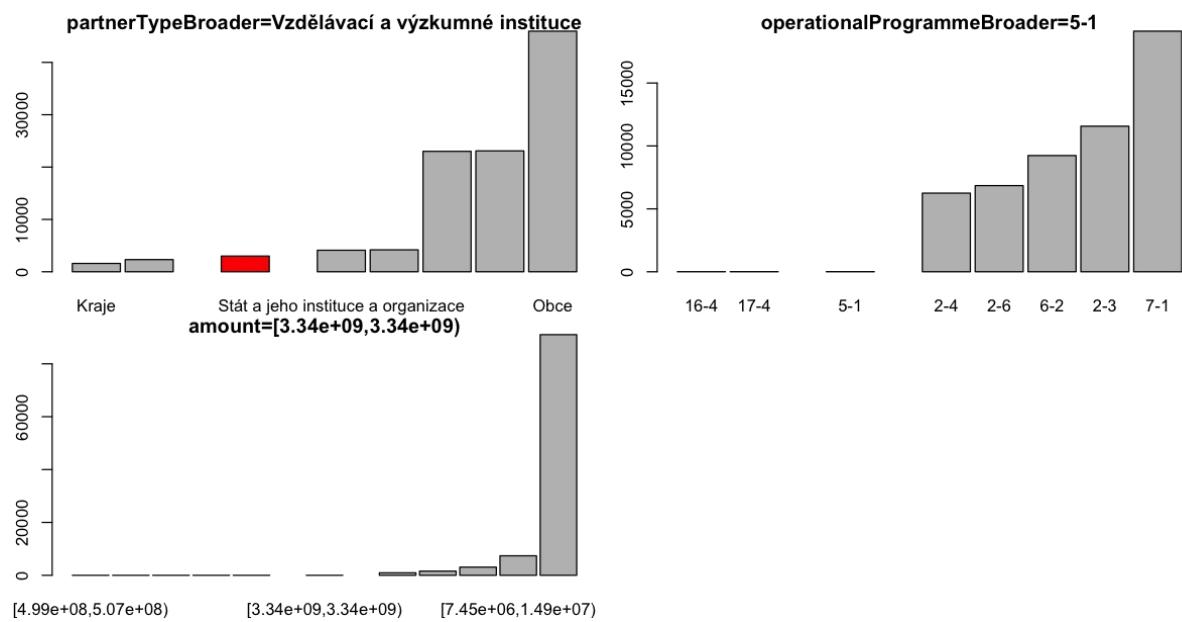


Figure 40: Visualization of the anomaly instance - it is composed from less frequent items (red bars in the middle of each individual bar plot).

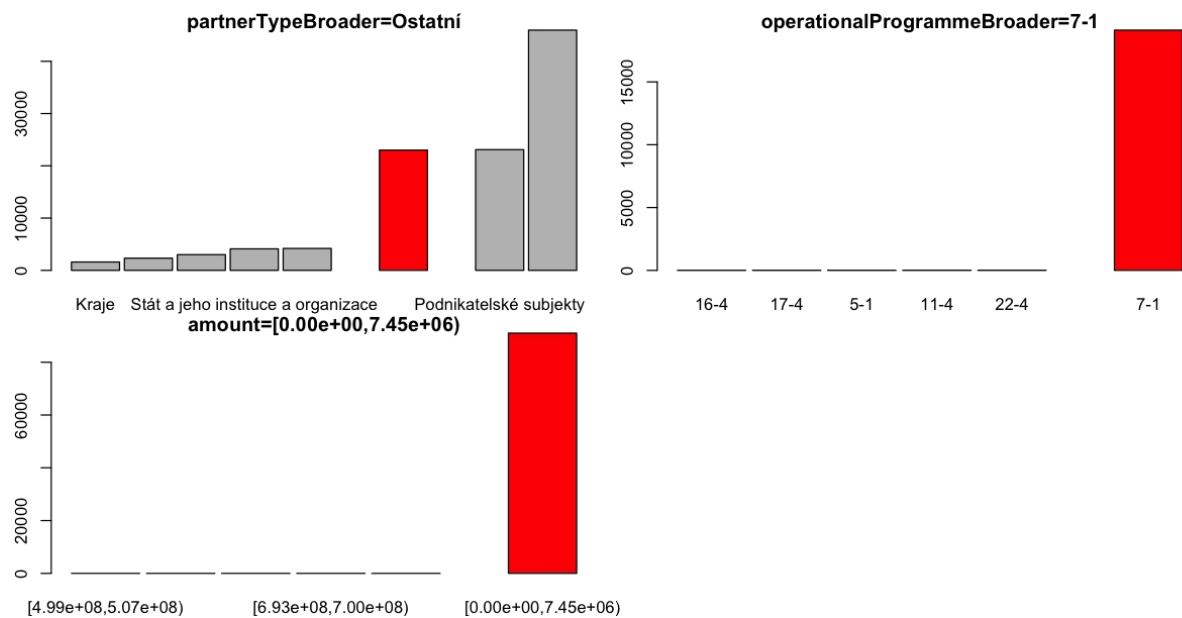


Figure 41: Visualization of the regular instance - it is composed from more more frequent items (red bars in each individual bar plot).

3.6.3.3 Financial ratios

For demonstration purposes we selected a specific dataset that contains financial data for several countries (namely [ESIF 2014](#)). The dataset contains observations described by several attributes. This demonstration considers only attributes that define a country name (MS Name), fund (Fund), category (To short) and amounts (Total Amount).

We grouped all observations by the country (e.g. Austria, Belgium, ...). For each group (country) we pre-computed all possible combinations of values based on fund and category attributes. Such combinations can be used to compute ratios of amounts.

Example for Austria:

The sum of all amounts for Austria is approx. 10,655,136,237.8

For Fund=EAFRD is the sum of amounts: 7,699,887,667.78

The ratio of Fund=EAFRD on the total sum is: 0.722646 (the ratio is further labelled as /Fund=EAFRD)

For Fund=EAFRD we can generate more sub-ratios:

The sum of amount for Fund=EAFRD: 7,699,887,667.78

For category "Climate Change Adaptation & Risk Prevention" within the Fund=EAFRD is the sum of amounts: 2,516,805,874.14

The ratio of To Short=Climate Change Adaptation & Risk Prevention on Fund=EAFRD for Austria is: 0.326863 (Further labelled as /Fund=EAFRD/To Short=Climate Change Adaptation & Risk Prevention)

Computed ratios are directly not very useful for detection of uncommon observations. Therefore, we computed the deviation of the ratio from the mean value of the same ratios calculated for other groups (countries).

Example for Austria:

The mean value of ratios for Fund=EAFRD on total sum of amounts is 0.29 (ratio for Austria is excluded).

The deviation from the mean value is thus approx. 0.44. When compared to other values of the same ratios, this deviation is unusually higher and such observation can be considered as "outlier".

The deviations of ratios are visualized on the figure below. Red colour represents “positive outliers”, where ratios are higher than the mean value. Blue colour stands for “negative outliers”, where the ratios are significantly lower than the mean value. Groups (countries) are represented as rows and ratios as columns. The interactive visualization is available as [HTML page](#) (Figure 42).

Examples of results:

- Netherlands has significantly higher ratio of EAFRD fund on Educational & Vocational training than the others. The ratio of ESF is significantly lower (Figure 43).
- The ratio of EAFRD fund on sum of all is higher for Luxembourg than for the other countries (Figure 44).

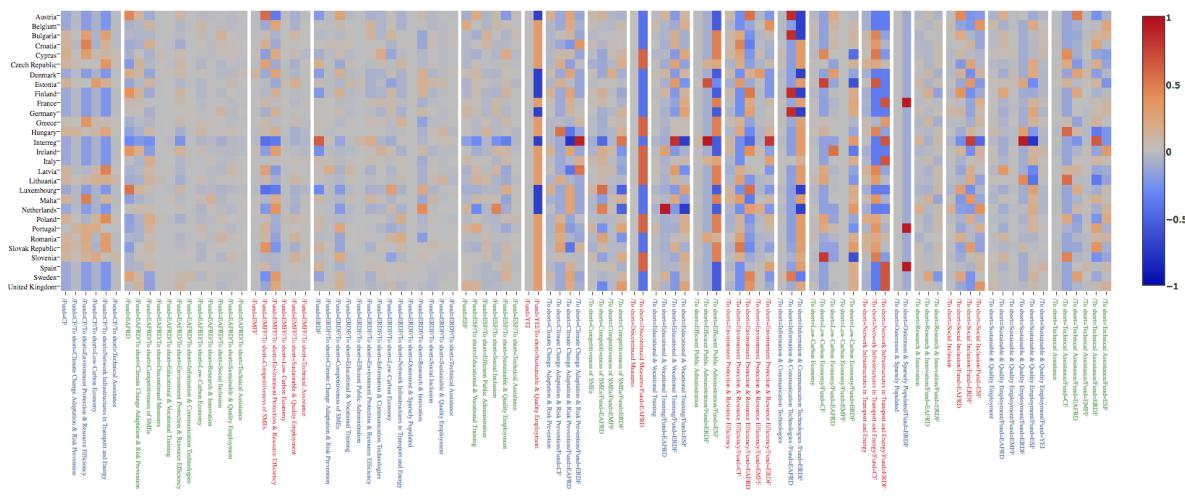


Figure 42: Financial ratios visualization.

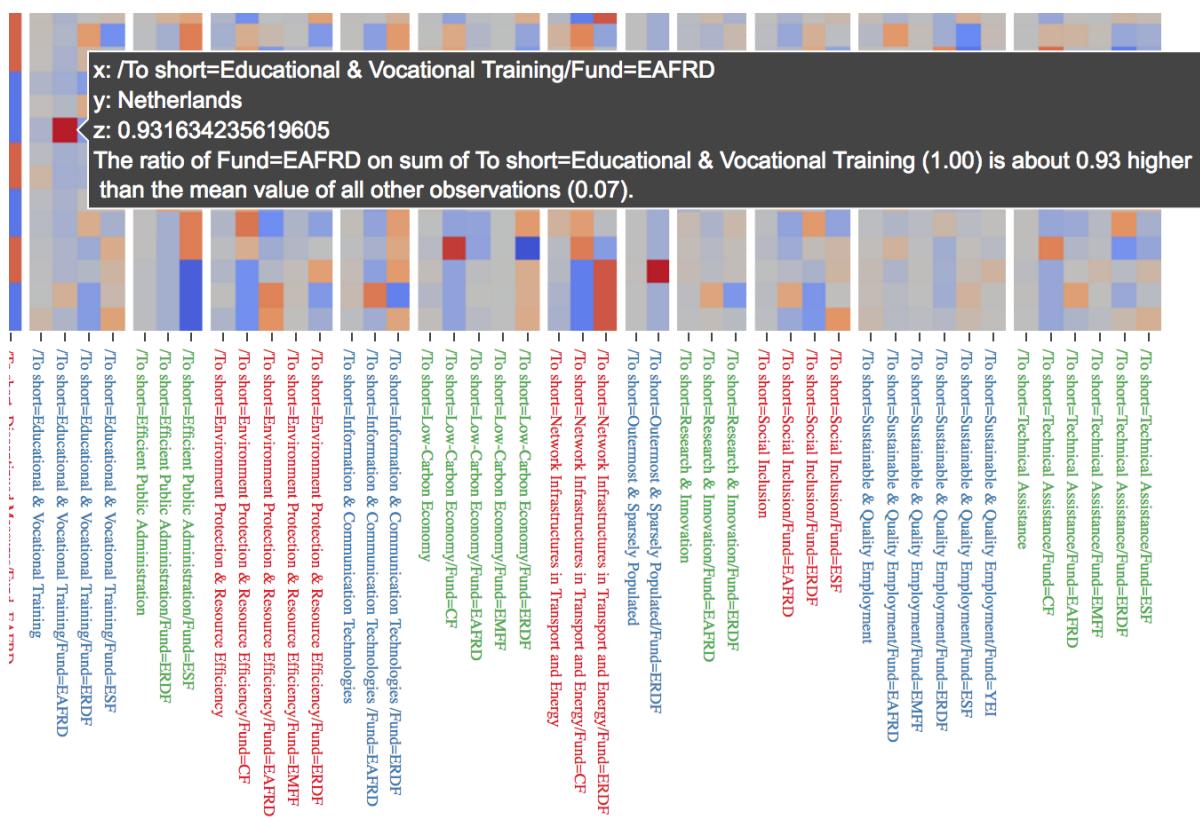


Figure 43: Detail of the EAFRD fund for Netherlands.

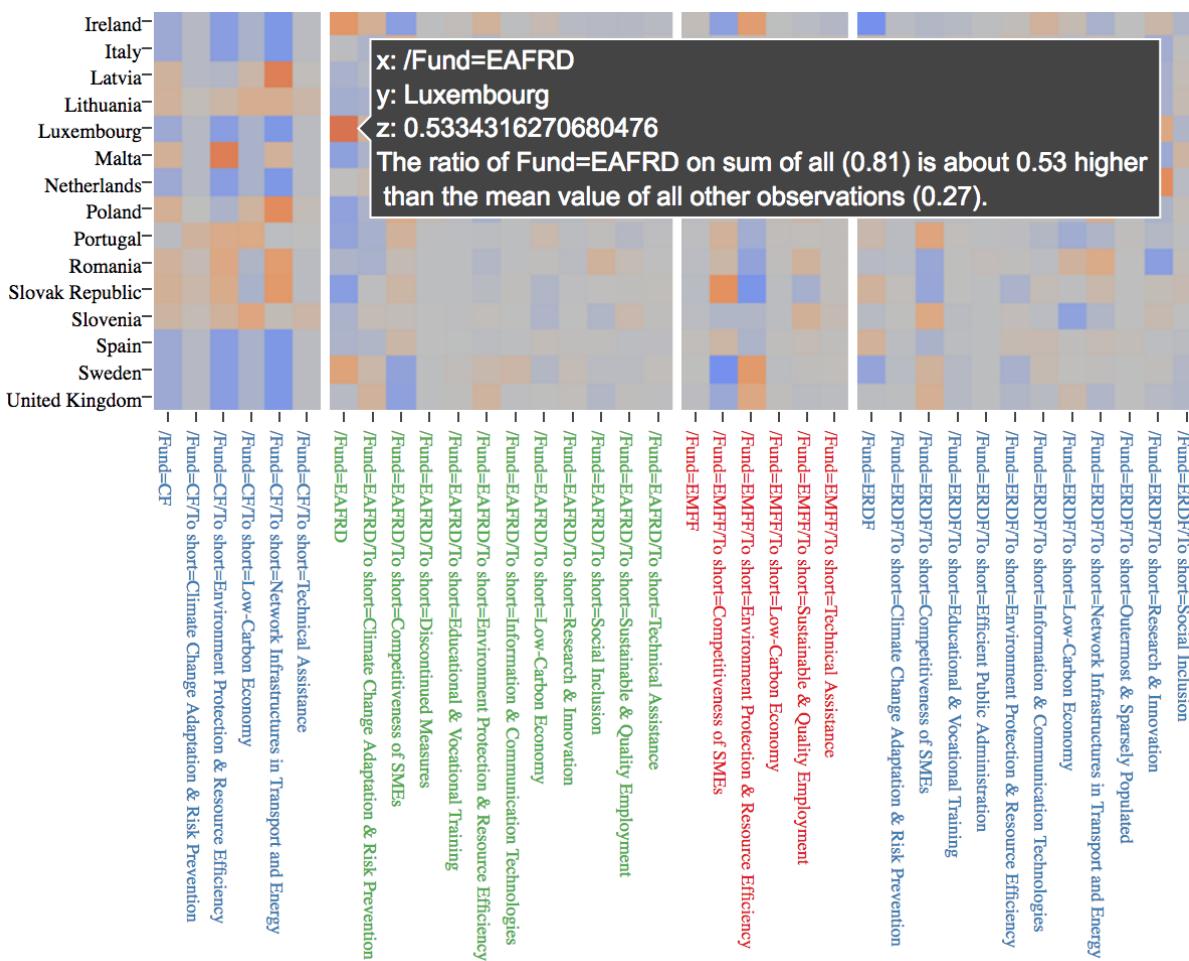


Figure 44: Detail of the EAFFRD fund for Luxembourg.

4 Guidance on methods and comparison to requirements

This section attempts to summarize the inventory of developed and implemented methods with respect to possible end-user requirements. The first section, 4.1, traces back from the methods to situations in which they could be useful. The second section, 4.2, maps the achieved coverage of end-usage requirements as they have been collected in D2.3.

4.1 Relevant situations for applying the mining methods

Abstracting from the large number of detailed user needs, three broad types of information-seeking situations can be identified: seeking anomalies, commonalities, and making (mostly pairwise) comparisons.

A large part of the user requirements may concern the detection of *unusual* or *anomalous* situations. The methods explicitly implemented as ‘Outlier/anomaly detection’ (Section 3.7), but also some other methods, can serve for this purpose.

The suitability of the methods depends on the structure of data in the analyzed table:

- For data with granularity of individual *projects* funded from the budget, the method of anomaly detection based on *frequent patterns* (3.7.1.2) can be useful. It is however not specifically focused on the *monetary amount*; thus the outlier status can easily be based on a combination of low-frequency values in other features of the project. Anomalous projects are worth further investigation, which possibly, if confirmed by analyzing non-budget data as well, might indicate corruption cases. The simplest cases of anomaly, consisting of extraordinarily high/low monetary amount values, can be visually identified via descriptive statistics, especially using boxplots (3.1.2).
 - The subpopulation-based LOF method can be extended to a family of subpopulation-based method, if we apply other outlier-detection method for each subpopulation. One advantage of subpopulation-based method, as described in Fleischhacker, et al. (2014), is to increase the accuracy of outlier-detection by considering several subpopulations of a suspected outlier-data. The creation of subpopulation is largely flexible. New methods can be created based on real use cases. If we create a subpopulation where “data with granularity of individual *projects* funded from the budget” and examine their *frequent patterns*, we just have our frequent pattern based methods.
 - For data aggregated from the level of individual projects to the level of, e.g., regions or countries, the presumed task is that of finding bulk deviations from expected use of budgets. The method of calculating and visualizing the *financial ratios* (3.7.1.3) may be applied, in order to indicate long-term issues such as loopholes in some national legislation or improper mapping of classifications between the higher and lower levels of the fiscal hierarchy. Another possible task is that of tracking *frequent patterns* possibly indicating mass behavior of project partners, such as optimizing the co-funding rate to a specific value or (almost) exclusively using a funding source for a particular purpose (or vice versa).
 - The methods based on (association) *rule mining* (3.4) can be used to detect frequent patterns of this kind. Unexpected strong relationships, e.g., between a spending area and a funding source, are then worth investigating using extra-budget sources, in particular, the regulatory code and the repeated features in calls for tender.
 - Quantitative strength of structurally simple, quantitative patterns, such as that of projects proceeding from one approval phase of another in the workflow, can be assessed using simple visualizations on the top of *descriptive statistics* such as correlation coefficients (3.1.2).
- If the goal is to position a given municipality (or other fiscal subject) *with respect to others*, in order to find out about possible overall deviations, two ways are possible:

- Selected subjects can be compared, typically pairwise, using *comparative analysis* (3.4). This presumes the selection has been already carried out with respect to similar “size” of the subjects (as the current one), be it according to the municipality population, GDP level, or the like. Simpler, using this kind of methods, or possible time series analysis (3.2), is the comparison of the budget of the same entity across years.
- The subjects can be first *clustered* (3.3). Then the co-clustered subjects can be examined more carefully, especially if the co-clustering looks “surprising”.

4.2 Coverage of end-user requirements

At the end of D2.3, there is a notification that the requirements might be updated according to the upcoming deliverables of our use case partners, e.g. D5.3 by Kayser-Bril (2016).

Therefore, all the requirements of D2.3 are being revisited here, in a tabular form, referring to the original numbers in this previous deliverable. For each, an assessment is formulated of whether the need is covered by some of the implemented methods, and classify the degree of coverages in five qualitative level: Covered, Partially covered, Not covered/Solved elsewhere, Not covered/Under research, and Not covered/ignored.

A need being “Covered” is understood as follows: (1) there is a data structure defined, which is based on budget datasets, and (2) there is a data-mining algorithm, which accepts the data structure in (1), and produces an output which meets the need.

A need being “Partially Covered” is understood as follows: (1) it is possible to define a data structure for the need, and (2) there is a data-mining algorithm, which accepts the data structure in (1), or the output of the algorithm contains information which meets the need. But, selecting needs further domain knowledge.

A need being “Not covered/Solved elsewhere” is understood as follows: the need should not be addressed in the core part of the data-mining techniques, rather elsewhere, e.g. in data-mining pre-processing stage.

A need being “Not covered/Under research” is understood as follows: (1) there are budget datasets, to understand their contents, we need specific domain knowledge, and are currently working on manually, and (2) data mining algorithm relies on external AI techniques, which are not reliable yet.

A need being “Not covered/Ignored” is understood as follows: (1) there is no budget dataset, which can be used to abstract some data structures for data mining algorithms, or (2) specific domain knowledge beyond budget datasets are missing, which however is mandatory for the analysis.

Need	Description <i>Details from the initial</i>	Coverage
-------------	---	-----------------

	<i>requirement collection are in italics</i>	
N01	Filtering commensurable objects	<p>Covered in 3.3 Clustering and Similarity learning: clustering (applied using a restricted set of features) can help identify objects that are likely commensurable for the purpose of further analysis.</p> <p>Partially covered In 3.6 Outlier/anomaly detection. Incommensurable objects normally have lower density compared with commensurable objects. However, Identifying commensurability among a set of objects requires much stricter similarity than just the fact that none of them is an outlier.</p>
N02	Version tracking of budgets	<p>Covered</p> <p>in 3.2 Time series analysis and predictions</p> <p>in 3.3 Comparative analysis</p> <p>Version tracking of budgets means “analysis of evolution of budgets throughout its preparation phase”. Given a series of budget dataset in its different phases, both of the algorithms in 3.2 and 3.3 can be applied.</p>
N03	Indexing data with respect to tabular versus graph structures	<p>Not covered/Solved elsewhere</p> <p>This need is collected in D4.2, and will be solved within WP4 and the pre-processing stage for data-mining.</p>
N04	<p>Outlier detection</p> <p><i>Reveal categories that are used disproportionately.</i></p> <p><i>Outlier detection can find misclassifications, where lot of spending is non-transparently classified.</i></p>	<p>Covered</p> <p>with automated analysis and visualization only serving as support for manual expert assessment</p> <p>Method: Anomaly detection via financial ratios (3.7.1.3)</p>
N05	Extrapolations on data	<p>C o v e r e d</p> <p>in 3.2 Time series analysis and predictions</p> <p>Here, Extrapolations on data refers to “the ability to outline trends for future budget allocations”, which is a typical case of 3.2.</p>
N06	Aggregation by time interval	<p>C o v e r e d</p> <p>in 3.1 Descriptive statistics.</p> <p>The description of N06 is “Ability to aggregate (e.g., sum, average) amounts over a user-defined period of time (e.g., quarter)”. After user defined a period, the pre-processing module will group item in the</p>

		same period, and generate the input to the data-mining package in 3.1.
N07	Temporal trend of the difference between planned and actual spending	C o v e r e d in 3.2 Time series analysis and predictions. in 3.3 Comparative analysis This data mining and analytics need is related to (N02) and extends it with a temporal dimension involving budget data from several years and incorporating corresponding spending data. Another aspect is to investigate and analyze the reasons for the detected trends.
N08	Perform aggregations and simple statistics	Covered in 3.1 Descriptive statistics. The meaning of this need is: Perform aggregations and simple statistics for a better understanding of the data and to support journalist unexperienced in budgeting to find the demanded values. This is a typical case of 3.1 Descriptive statistics
N09	Features for experienced users/journalists	Not covered/Ignored A series of case studies conducted in D.5.3 trying to identify some useful (hopefully, also computable) features resulted in ‘non-availability’ and ‘obscurity’.
N10	Detect in-kind spending and gifts	Not covered/Ignored A series of case studies conducted in D.5.3 showed that it is extremely complex and sometimes impossible to find out who are the actual beneficiaries. As described in D.5.3: “In all of these cases, any misuse of fund that follows the bribe does not show in public budget data. Journalists pursuing these stories mostly rely on investigations from anti-corruption police or on leaks and testimonies.” A similar conclusion follows from a research at UEP: http://openbudgets.eu/post/2016/06/14/tracing-eu-funds/
N11	Incorporate accounting legislation into the analysis	Not covered/Ignored As reported in D5.3, “most of the categories used in regular accounting cannot be linked to a specific mission. Instead, they are linked to paying units” “Doing analytical accounting can be, in itself, extremely complex. ” “In some other cases,

		analytical accounting is impossible to carry out because the documents that would allow it (mostly contracts between parties to the realization of a public service) are not released.”
N12	Perform comparisons measuring how the data has changed when a data set has been updated	Covered in 3.4 Comparative Analysis The meaning of this need is: how the data has been changed when a data set has been updated? A case addressed in 3.4 Comparative Analysis
N13	Analyze larger trends over time and in different funding areas	Covered in 3.2 Time series analysis and predictions This need matches with (N02) and (N07) and extends it to a general trend analysis on the temporal dimension in budget and spending data.
N14	Identify both good and bad examples of financial management	Uncovered/Ignored in 3.5 Rule and pattern mining Predictive rules can be learned from labeled training data. However, it has been found out that such data is hard to obtain. Eg., audit report results are usually not available in machine-readable format. in 3.6 Outlier/anomaly detection Outliers may under some circumstances be viewed as candidates for bad financial management. However, in some contexts (e.g., in countries with dysfunctional government bodies) the inliers may be “bad” and some outliers may be “good”, in turn.
N15	Pay special focus on analyzing the spending and management of EU budget funds	Covered in 3.2 Time series analysis and predictions 3.4 Comparative analysis 3.5 Rule and pattern mining
N16	Identify systematic problems of project implementation in different funds and programmes, rather than in-depth engagement with individual projects	Partially Covered in 3.5 Rule and pattern mining 3.6 Outlier/anomaly detection, both being applied on aggregated data on funds and programmes.
N17	Consider fiscal indicators like error, performance and absorption rates	Covered in 3.1 Descriptive statistics 3.2 Time series analysis and predictions 3.6 Outlier/anomaly detection.
N18	Perform comparative	Covered in 3.4 Comparative Analysis

	analysis of certain budget and expenditure areas through the use of timelines; geographically; and by sector	
N19	Complement the raw budget data with other sources such as annual audit or activity reports	Not covered/Ignored
N20	Comparisons of previous years' budgets with the current one	Covered in 3.4 Comparative Analysis In 3.2 Time series analysis and predictions
N21	Provide context information	Not covered/Ignored A case study reported in D5.3 shows that it is not possible even for journalists to find the context of budget data. https://github.com/openbudgets/openbudgets.github.io/blob/7b159376a055a2d9e103e199ba84be49b9f2b719/_posts/2016-06-27-cost-refugees.md
N22	Comparative analysis	Covered in 3.4 Comparative Analysis
N23	Aggregations	Covered in 3.3 Clustering and Similarity learning
N24	Identifying fishy relations and red flags using network analysis	Not covered/Under researching As networking analysis needs interlinking of two codelists. However, interlinking is not trivial, codelists can be explained in different languages, budgets can be calculated in different accounting systems. A research on inter-linking is being carried out at UBon, and UEP.
N25	Red Flags for tenders and contracts indicating corruption, mistakes, ...	Not covered/Ignored
N26	Detection of politicians involved in receiving subsidies	Not covered/ignored. A case study reported in D5.3 shows that it is not possible even for journalists to find the context of budget data. https://github.com/openbudgets/openbudgets.github.io/blob/7b159376a055a2d9e103e199ba84be49b9f2b719/_posts/2016-06-27-cost-refugees.md
N27	Incorporating information of the budget process, information on politicians, public procurement, and	Not covered/Under researching An on-going research at UEP, trying to process and link public procurement data

	private companies receiving money from the state	
N28	Detection of corruption	<p>Not covered/Ignored.</p> <p>A case study reported in D5.3 shows that it is not possible even for journalists to find the context of budget data.</p> <p>https://github.com/openbudgets/openbudgets.github.io/blob/7b159376a055a2d9e103e199ba84be49b9f2b719/_posts/2016-06-27-cost-refugees.md</p>
N29	Follow the state's money flows all the way down to transaction data and then questioning who was receiving the money and if this happened in a proper manner	<p>No covered/Ignored.</p> <p>A case study reported in D5.3 shows that it is not possible even for journalists to find the context of budget data.</p> <p>https://github.com/openbudgets/openbudgets.github.io/blob/7b159376a055a2d9e103e199ba84be49b9f2b719/_posts/2016-06-27-cost-refugees.md</p> <p>http://openbudgets.eu/post/2016/06/14/tracing-eu-funds/</p>
N30	Include actual statistics	<p>Covered</p> <p>in 3.1 Descriptive statistics, input, algorithms, output are explained in detail in 3.1</p>
N31	Provide context to budget and spending data	<p>Not covered/Ignored.</p> <p>A case study reported in D5.3 shows that it is not possible even for journalists to find the context of budget data.</p> <p>https://github.com/openbudgets/openbudgets.github.io/blob/7b159376a055a2d9e103e199ba84be49b9f2b719/_posts/2016-06-27-cost-refugees.md</p>
N32	Compare the same budget line across countries and cities	<p>Partially Covered</p> <p>in 3.3 Comparative Analysis</p>
N33	Detect council members tied to companies winning tenders	<p>Not covered/Ignored.</p> <p>A series of case studies conducted in D.5.3 show that it is extremely complex and sometimes possible to find out who are the actual beneficiaries. As described in D.5.3: "In all of these cases, any misuse of fund that follows the bribe does not show in public budget data. Journalists pursuing these stories mostly rely on investigations from anti-corruption police or on leaks and testimonies."</p>

		The same conclusion is made by a research at UEP: http://openbudgets.eu/post/2016/06/14/tracing-eu-funds/
N34	Implement notifications on specific changes in certain data sets, monitoring	Not covered/Ignored. This need does not belong to the part of the data-mining technique, rather belong to the simulation of digital user, who monitors the data-set on web.
N35	Address questions like “How is the money really used?” and “How do I profit from my salary taxes?”	Not covered/Ignored. A series of case studies conducted in D.5.3 shows that it is extremely complex and sometimes possible to find out who are the actual beneficiaries. The same conclusion is made by a research at UEP: http://openbudgets.eu/post/2016/06/14/tracing-eu-funds/
N36	Tracking the EU money through the different levels down to the actual beneficiaries	Not covered/Ignored. A series of case studies conducted in D.5.3 show that it is extremely complex and sometimes possible to find out who are the actual beneficiaries. The same conclusion is made by a research at UEP: http://openbudgets.eu/post/2016/06/14/tracing-eu-funds/
N37	Incorporate key performance indicators in the analysis	Not covered/Ignored Datasets with KPI are not available.

Table 44. List of 37 needs in D2.3, with an evaluation to the coverage by data-mining tools

5 Conclusions and Ongoing work

In this deliverable, we summarized (1) software tools developed for the data-mining and analysis, (2) detailed descriptions about the data-mining methods and techniques used for the software development, (3) an evaluation of how the requirements reported in D2.3 are met, with an update based on D5.3.

Six data-mining packages are developed: (1) descriptive statistics, (2) time series analysis and prediction, (3) comparative analysis, (4) rule/pattern mining, (5) clustering and similarity learning, and (6) outlier/anomaly detection. These 6 packages cover 17 needs appears in D2.3. Open-source libraries, such as R, Java, and Python data-mining libs, are used. In case no existing lib available, we searched up-to-date scientific publications, developed data-mining packages, and make them public accessible.

Among totally 37 needs in D2.3, 15 needs do not be accompanied with datasets, therefore ignored, as reported in D5.3. For example, the Need 14 (Identify both good and bad examples of financial management) can be covered in the task Rule and pattern mining or Outlier/anomaly detection, However, it has been found out that labeled training data is hard to obtain. Two needs are addressed elsewhere, one (N16) is partially covered. Details are listed in Table 45.

Status of Coverage	Number of Needs	Need	Needs from Whom
Covered	17	N1-N2, N4-N7	D4.2
		N8, N12	D5.1
		N13, N15-N19, N32	D6.2
		N20	D7.1
		N22	D8.3
Partially covered	1	N16	D6.2
Not covered/Solved elsewhere	2	N3	D4.2
		N9	D5.1
Not covered/Under researching	2	N24, N27	D8.3
Not covered/Ignored	15	N10-N11	D5.1
		N14, N19	D6.2
		N21	D7.1
		N25, N26, N28, N29, N31, N33-N35	D8.3
		N36, N37	raised during discussions in project meetings

Table 45. A statistics of the status of coverage, and needs

For the 8 needs from D4.2, 7 among them (87.5%) are covered, the left one is addressed in the pre-processing stage of the data-mining task. For the 5 needs from D5.1, 2 among them (40%) is covered, one (N9) is addressed at the data-mining interface, two of them are ignored. For the 10 needs from D6.2, 7 among them (70%) are covered, one are partially covered, two are ignored. For the 2 needs from D7.1, 1 among them (50%) is covered, the other is ignored. For the 11 needs from D8.3, 2 are under research, the rest (81.82%) are ignored.

Some Needs demand hard and time-consuming manual work to search and investigate datasets and related datasets. For example, Need 24 (Identifying fishy relations and red flags using network analysis) requires networking analysis, which needs interlinking of two codelists. However, interlinking is not trivial, codelists can be explained in different languages, budgets can be calculated in different accounting systems. A research on interlinking is being carried out at Ubonn, and UEP. Similar to the Need 27 (Incorporating information of the budget process, information on politicians, public procurement, and private companies receiving money from the state): A research work is being carried out at UEP, trying to process and link public procurement data. These two needs are marked as “Uncovered/Under research”.

When items in two datasets can be associated using any kind of certain classifications, analysis can be conducted in more interesting way. We can consider datasets from different EU region, which use different languages, such as datasets from Bonn (in German) and from Aragon (in Spanish). Both datasets uses functional classification. For example, we can compare budgets allocated for social health care from both public administrations. Currently as a research task, we are currently researching on how to make such interlinking of similar concepts from dataset with different languages. Our initial experiment on doing interlinking from out-of-the-box translation using SILK framework resulted in around 50% accurate created links. To improve this interlinking task, we are going to use two approaches: Natural Language Processing and Semantic Web approach.

References

- Agrawal, et al. (1993). Buneman, Peter and Sushill Jajodia: *Proceedings of the 1993 (ACM SIGMOD) International Conference on Management of Data*. New York, USA: ACM Press, pp. 1-10. ISBN 0-89791-592-5, 1993.
- Alberts, et al. (2016a). Alberts, A., Wurning, D., Kayer-Bril, Bouyer, A.L: Deliverable 5.1-Test Beds and Evaluation - Journalism. https://drive.google.com/file/d/0B9Qc3_LouTLSnZTNkNVdHNBeTg/view 2016
- Alberts, et al. (2016b). Alberts, A., Wagner, E., Guen, C.L., Kayer-Bril, N., Campos, A.D., Lämmerhirt, D., Graz, J., Sedmihradská, L. Deliverable 8.3-Stakeholder Identification Strategy and Outreach Plan. https://drive.google.com/file/d/0B9Qc3_LouTLWm5NVUIRdUFScGs/view 2016.
- Breuning, et al. (2000). Breunig, M. M., Kriegel, H.-P., Ng, R. T., Sander, J. *LOF: Identifying Density-based Local Outliers* (PDF). *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*. SIGMOD. pp. 93–104, 2000
- Calderón, D.C. et al. (2016). Calderón, D.C., Belda, E.B., Poblete, R.D., Reviriego, D.A.C., de Vega de la Sierra, J.: Deliverable 7.1 - Test Beds and Evaluation - Participatory Budget . <https://drive.google.com/file/d/0Bx9zPWPFoVRaNDY3UFhsUkNUMm8/view> 2016.

- Chandola, et al. (2009). Chandola, Varun; Banerjee, Arindam; Kumar, Vipin. Anomaly detection: A survey. In *ACM computing surveys (CSUR)*. pp. 1 - 72. 2009.
- Feng, et al. (2010). Feng Lin, Wang Le, Jin Bo - Research on Maximal Frequent Pattern Outlier Factor for Online High Dimensional Time-Series Outlier Detection. *Journal of Convergence Information Technology* 5(10):66-71 · December 2010.
- Fleischhacker, et al. (2014). Fleischhacker, D., Paulheim, H. Bryl, Völker, J., Bizer, Ch. *Detecting Errors in Numerical Linked Data using Cross-Checked Outlier Detection*. In: 13th International Semantic Web Conference, pp 357-372, Riva del Garda, Italy, October 19-23, 2014. Proceedings, Part I.
- Gökgöz, F. et al. (2015). Gökgöz, F., Auer, S., Takis, J.: Deliverable 4.2 - OpenBudgets.eu Requirements, Platform Architecture Integration and Development. https://drive.google.com/file/d/0B9Qc3_LouTLUFhmT1BkaDFTWTQ/view 2015
- He, et al. (2005). He, Z., Xu, X., Huang, J. Z., Deng, S.: FP-Outlier: Frequent Pattern Based Outlier Detection. *Computer Science and Information Systems*, Vol. 2, No. 1, 103-118. 2005.
- Jiadong, et al. (2009). Jiadong Ren, Qunhui Wu, Changzhen Hu, and Kunsheng Wang: An Approach for Analyzing Infrequent Software Faults Based on Outlier Detection. In *Proceedings of the 2009 International Conference on Artificial Intelligence and Computational Intelligence - Volume 04 (AICI '09)*, Vol. 4. IEEE Computer Society, Washington, DC, USA, p. 302-306, 2009.
- Kayser-Bril (2016). Kayser-Bril, N.: OBEU - Deliverable 5.3 Test Beds and Evaluation - Journalism, <https://drive.google.com/file/d/0Bx9zPWPFoVRadzRfWXUwY2Y5MzA/view> 2016.
- Klímek (2015a). Klímek J., Kučera J., Mynář J., Sedmihradská L., Zbranek J.: OBEU - Deliverable D1.2 - Design of data structure definition for public budget data, <http://openbudgets.eu/assets/deliverables/D1.2.pdf>, 2015a
- Klímek (2015b). Klímek J., Kučera J., Mynář J., Sedmihradská L., Zbranek J.: OBEU - Deliverable D1.3 - Design of data structure definition for public spending data, <http://openbudgets.eu/assets/deliverables/D1.3.pdf>, 2015b.
- Koupidis, et al. (2016). Koupidis, K., Bratsas, C., Karampatakis, S., Martzopoulou, A., Antoniou, I., *Fiscal Knowledge discovery in Municipalities of Athens and Thessaloniki via Linked Open Data*. In: *Proceedings - 11th International Workshop on Semantic and Social Media Adaptation and Personalization*, SMAP 2016, art. no. 7753405, pp. 171-176.
- Rauch, Jan (2013). Observational Calculi and Association Rules [online]. 1. vyd. Berlin : Springer-Verlag. 296 p. ISBN 978-3-642-11736-7. ISSN 1860-949X.
- Tang, et al.(2009). X. Tang, G. Li and G. Chen, "Fast Detecting Outliers over Online Data Streams," 2009 International Conference on Information Engineering and Computer Science, Wuhan, 2009, pp. 1-4.
- Zhang, et al.(2010). W. Zhang, J. Wu and J. Yu, "An Improved Method of Outlier Detection Based on Frequent Pattern," *Information Engineering (ICIE), 2010 WASE International Conference on*, Beidaihe, Hebei, 2010, pp. 3-6.
- Zhou, et al. (2007). ZHOU Xiao-Yun+, SUN Zhi-Hui, ZHANG Bai-Li, YANG Yi-Dong - A Fast Outlier Detection Algorithm for High Dimensional Categorical Data Streams. *Journal of Software* 18(4) · April 2007.

Referenced Internet Links:

1. https://en.wikipedia.org/wiki/Probability_distribution
2. <https://statistics.laerd.com/statistical-guides/measures-central-tendency-mean-mode-median.php>
3. <https://statistics.laerd.com/statistical-guides/measures-central-tendency-mean-mode-median.php>
4. <http://www.abs.gov.au/websitedbs/a3121120.nsf/home/statistical+language+-+measures+of+spread>
5. <https://en.wikipedia.org/wiki/Percentile>
6. https://en.wikipedia.org/wiki/Interquartile_range
7. https://en.wikipedia.org/wiki/Variance#Sample_variance
8. https://en.wikipedia.org/wiki/Central_moment
9. https://en.wikipedia.org/wiki/Standard_deviation
10. <https://en.wikipedia.org/wiki/Skewness>
11. <https://en.wikipedia.org/wiki/Kurtosis>
12. <http://www.investopedia.com/terms/k/kurtosis.asp>
13. <https://en.wikipedia.org/wiki/Histogram>
14. https://en.wikipedia.org/wiki/Normal_distribution
15. <http://www.investopedia.com/terms/p/platykurtic.asp>
16. https://en.wikipedia.org/wiki/Box_plothttps://upload.wikimedia.org/wikipedia/commons/thumb/1/1a/Boxplot_vs_PDF.svg/550px-Boxplot_vs_PDF.svg.png
17. <https://en.wikipedia.org/wiki/Histogram>
18. https://en.wikipedia.org/wiki/Pearson_correlation_coefficient
19. <https://en.wikipedia.org/wiki/Ranking>
20. https://en.wikipedia.org/wiki/Monotonic_function
21. https://en.wikipedia.org/wiki/Spearman%27s_rank_correlation_coefficient
22. https://en.wikipedia.org/wiki/Kendall_rank_correlation_coefficient#Accounting_for_tie
23. https://en.wikipedia.org/wiki/Rank_correlation
24. <http://www.ats.ucla.edu/stat/sas/library/loesssugi.pdf>
25. https://en.wikipedia.org/wiki/Local_regression
26. https://en.wikipedia.org/wiki/Partial_autocorrelation_function
27. https://en.wikipedia.org/wiki/KPSS_test
28. <https://www.rtmath.net/help/html/695835bf-570e-411f-9d76-05ee2570d0d7.htm>
29. https://en.wikipedia.org/wiki/Augmented_Dickey%20%93Fuller_test
30. https://en.wikipedia.org/wiki/Phillips%20%93Perron_test
31. <http://staff.bath.ac.uk/hssjrh/Phillips%20Perron.pdf>
32. http://vsp.pnnl.gov/help/Vsample/Design_Trend_Mann_Kendall.htm
33. https://en.wikipedia.org/wiki/Autoregressive_integrated_moving_average#Definition
34. <https://people.duke.edu/~rnau/411arim.htm>
35. https://en.wikipedia.org/wiki/Vector_quantization
36. https://en.wikipedia.org/wiki/Voronoi_diagram
37. https://en.wikipedia.org/wiki/K-means_clustering
38. <https://en.wikipedia.org/wiki/K-medoids>
39. <http://www.cs.umb.edu/cs738/pam1.pdf>

40. <https://en.wikipedia.org/wiki/Centroid>
41. https://en.wikibooks.org/wiki/Data_Mining_Algorithms_In_R/Clustering/CLARA
42. <http://www.cs.ecu.edu/dingq/CSCI6905/readings/CLARANS.pdf>
43. https://en.wikipedia.org/wiki/Fuzzy_clustering
44. https://en.wikipedia.org/wiki/Normal_distribution
45. <http://www.public.iastate.edu/~maitra/stat501/lectures/ModelBasedClustering.pdf>
46. <http://www.sthda.com/english/wiki/model-based-clustering-unsupervised-machine-learning>
47. <http://faculty.iiit.ac.in/~mkrishna/PrincipalComponents.pdf>
48. <https://georgemdallas.wordpress.com/2013/10/30/principal-component-analysis-4-dummies-eigenvalues-eigenvectors-and-dimension-reduction/>
49. https://en.wikipedia.org/wiki/Principal_component_analysis
50. https://en.wikipedia.org/wiki/Likelihood_function
51. https://en.wikipedia.org/wiki/Akaike_information_criterion
52. https://en.wikipedia.org/wiki/Bayesian_information_criterion
53. [https://en.wikipedia.org/wiki/Silhouette_\(clustering\)](https://en.wikipedia.org/wiki/Silhouette_(clustering))
54. https://ncss-wpengine.netdna-ssl.com/wp-content/themes/ncss/pdf/Procedures/NCSS/Fuzzy_Clustering.pdf
55. https://en.wikipedia.org/wiki/Local_outlier_factor

7. Appendix

7.1 lists of open-source data-mining tools developed till now

1. <https://github.com/okgreece/DescriptiveStats.OBeu>
2. <https://github.com/okgreece/TimeSeries.OBeu>
3. <https://github.com/okgreece/Cluster.OBeu>
4. <https://github.com/kizi/easyminer>
5. <https://github.com/jaroslav-kuchar/fpmoutliers>
6. https://github.com/openbudgets/outlier_dm
7. <http://okfnrg.math.auth.gr/ocpu/test/>
8. https://github.com/openbudgets/okfqr_dm
9. https://github.com/openbudgets/uep_dm
10. https://github.com/openbudgets/preprocessing_dm

7.2 Installation (Ubuntu) and start the server

The installation of the data-mining base module on a local Ubuntu platform is described in the README.md file at

https://github.com/openbudgets/DAM/tree/staging_indigo.

Step 1. First clone it, by typing

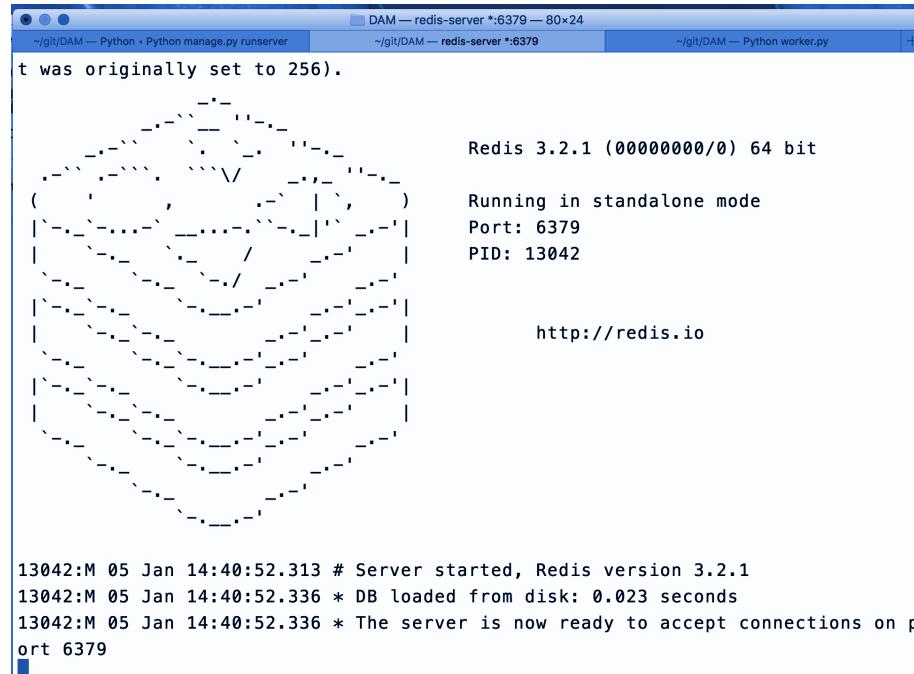
```
$ git clone https://github.com/openbudgets/DAM
Step 2. check out the staging_indigo branch, by typing
$ git checkout staging_indigo
Step 3. go to the DAM directory by typing
$ cd DAM
Step 4. install necessary libraries, such as git, redis-server, postgresql, python, and virtual environment. All these are realized by typing
$ make pre
Step 5. activate virtual environment by typing
$ source env/bin/activate
Step 6. install the basic DAM module, by typing
$ make dam
```

To start the data-mining server, we need to open three terminals, each in the `DAM` directory, and activate virtual environment.

Type `$ python3 manage.py runserver` in the first terminal. If it works correctly, following screenshot shall be seen.

```
(env) just-try:DAM tdong$ python3 manage.py runserver
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
 * Restarting with stat
 * Debugger is active!
 * Debugger pin code: 168-174-808
127.0.0.1 - - [05/Jan/2017 14:44:30] "GET / HTTP/1.1" 200 -
```

then type `$ redis-server` in the second terminal. If it works correctly, following screenshot shall be seen, shown in Figure A1.



The screenshot shows a terminal window with three tabs:

- `DAM — redis-server *:6379 — 80x24`
- `~/git/DAM — Python manage.py runserver`
- `~/git/DAM — redis-server *:6379`
- `~/git/DAM — Python worker.py`

The main pane displays the Redis server startup logs:

```
t was originally set to 256).
                                     Redis 3.2.1 (00000000/0) 64 bit
                                     Running in standalone mode
                                     Port: 6379
                                     PID: 13042
                                         http://redis.io

13042:M 05 Jan 14:40:52.313 # Server started, Redis version 3.2.1
13042:M 05 Jan 14:40:52.336 * DB loaded from disk: 0.023 seconds
13042:M 05 Jan 14:40:52.336 * The server is now ready to accept connections on port 6379
```

Figure A1.: Screenshot of a running redis-server

type \$ python3 worker.py in the third terminal. If it works correctly, following screenshot shall be seen.

```
(env) just-try:DAM tdong$ python3 worker.py
14:41:10 RQ worker 'rq:worker:just-try.13063' started, version 0.5.6
14:41:10
14:41:10 *** Listening on default...
█
```

To check whether the server functions well locally, open a web browser, and type <http://localhost:5000>. We shall see the response as illustrated in Figure A2.

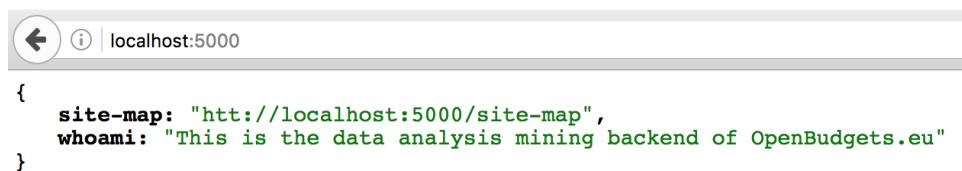


Figure A2.: Screenshot of a running DAM back-end