Heatmap Capstone Project Report
INFT 5021

Team:
Dona Joy Arimboor
Mansoureh Foruzandehshahraky
Aniket Varbude
Stephen Maher

Sponsor:
Sponsor: Dr. Jan Stanek
Sponsor: Prof. Georg Grossmann

Mentor:
Eric Lam

Executive Summary

This project delivers tools and workflows which provide for the analysis of event sequences and a report on the application of these to a supplied test dataset. The tools are designed as proof-of-concept for desktop PC analysis. This report outlines the theoretical justification behind the analytical techniques applied and demonstrates the application of the tools developed using a test dataset.

Exploratory Data Analysis (EDA) tools developed for visualisation and understanding encompass barplots, boxplots, directed graphs, bubble plots, n-gram tables, state sequence plots, state trace heatmaps among other analyses.

Modelling tools developed cover clustering, association rule and sequential association rule mining, LDA topic analysis and hazard and survival analysis modelling.

All tools have been produced within the R markdown framework with documentation provided between code blocks.

Results from both the EDA and modelling modules, as applied to the supplied test dataset are detailed in this report. There was no knowledge of the providence of the dataset and what the dataset represented.

The EDA module allowed ready identification and quantification of key data features. This included information such as trace-token distributions and token persistence.

The modelling modules identified trace clusters (with heatmap visualisations), motifs in the form of association rules embedded in the data, topic separation under LDA, and fitted survival curves.

No performance measures have been developed within the analysis as the project objective was to deliver analytical tools and models for a general data case.

Table of Contents

1.0 Statement of Purpose

The goal for the Heatmap Analysis Project (project) is to develop tools and workflows which support analysing event sequences.

The deliverables are workflow tools for EDA, modelling, and visualisation of event sequence data. The tools are to provide desktop proof-of-concept analysis for the client and will enable existing research into event sequence data and provide a platform for further development. The clients are familiar with R and with the algorithms used within the tools.

The project value proposition is that for a marginal change in their existing resourcing plan, the clients receive reporting on, and prototyped tools and workflows, which contribute to their future research activities.

2.0 Project Background

The clients are university researchers seeking to identify techniques available to facilitate meaningful analysis of a dataset. This project uses a dataset provided by the clients, comprising event sequence data. This type of data contains records (traces), comprising an ordered sequence of events (tokens or states) which are represented symbolically.

Traces are observed across a wide range of domains and there exists research which informs analysis, modelling, and visualisation methodologies. The volume of data can be material and analysis complicated by data complexity and heterogeneity. Using traditional analytical approaches, data volume and variety may result in time consuming and ineffectual analysis of the data, hence the need for tools which support analysis, visualisation, and modelling.

3.0 Literature Review

This literature review establishes the theoretical basis for the tools developed.

The basic terminology associated with event sequence analysis (ESA) varies with the domain analysed. Abbott [1] and Studer and Ritschard [2] describe event sequences occurring in psychology, economics, archaeology, linguistics, political science, sociology and biology, among others.

The dataset supplied comprises a set of sequentially ordered states which represent event sequences. These event sequences are called *traces*, and each trace is a row of data where each successive column in the row represents an incremental progression in sequence. For example, the first column in the data represents sequence location "1", the second column is sequence location "2", through to the last column at sequence location "n". Traces are often defined as an "ordered list of elements" [1, p. 94], where the ordering is temporal, but it could be steps in a process. The state in each column for a row is represented by a *token*. Tokens are sourced from a set of potential tokens for the dataset under analysis. One final definition is needed and that is for *motifs*: motifs are recurrent sub-sequences of tokens within traces.

For this project, *trace* and *event sequence,* and *token, event,* and *state,* are used interchangeably.

Abbot [1] discussed two approaches for evaluating such traces: casual and narrative. The casual approach focusses on treating event occurrences as a function of some underlying stochastic process. Casual techniques are not limited to stochastic processes and include Markov chains and hazard models. These approaches focus on the change in state between each period. In contrast, the narrative approach, also discussed by [2], [3], [4], and [5], focusses on whole-of-life, or the trajectory through events, of a trace. Two approaches described by [1] for trajectory analysis are algebraic and metric. The algebraic approach reduces traces to a simpler form and then groups traces using a similarity function. The metric method focuses on estimating the distance between full traces using some distance measure.

3.1 Optimal matching

Optimal (edit) Matching (OM[1]) is a common technique for estimating the distance between traces. Distance between traces is measured by the insertion, deletion, or substitution of tokens between traces until they are equivalent. A cost is assigned to this alignment of traces, defining the distance between the traces. Studer et.al [5] observed that OM is a useful tool for exploring and characterising traces, allowing analysis in the absence of an initial hypothesis. Abbot [1] noted that OM is readily adapted to specific problems, where the clustering of traces into similar groups using OM measures allows these groups to be subsequently used as dependent or independent variables in further analysis.

Limitations to OM were identified by [4], though [4] focussed on minimum requirements for trace lengths (more than 25 tokens), missing tokens (less than 30% in a trace) and operation costs. Another limitation [4], is that large datasets can dramatically increase computation time. Dlouhy et. al [4, pp. 166-167] tabled details from 27 analyses, with the number of unique tokens averaging around 10 and trace lengths averaged well below 100. Hence a combination of high frequency measurement (such as daily versus monthly or annually) and exceptionally long measurement histories, may present problems when applying OM to large datasets.

[1] Optimal Matching is also referred to as Optimal Alignment (OA)

3.2 Cluster analysis

Cluster analysis has been used extensively for the analysis of event sequences. Lu et. al [6] applied trace clustering to the clinical pathways experienced by patients. Le et. al [7] apply trace clustering to the analysis of business process logfiles.

Cluster analysis of traces requires data features between which distance can be measured. Lu et. al [6] identify features such as event frequency and the frequency of "*directly-followed* relations" [6, p. 200] and are token and token transition counts (also used by [8]). Other features used include OM measures ([6] and [8]) and model based methods using Markov models and Petri nets [6]. Le et. al [7] also identify a range of probabilistic modelling apporaches such as linear autoregression, graphs and hybrid Markov models. While noting the use of models to support trace analysis, [7] develop a hybrid approach to trace modelling in their research using localised OM as a feature for subsequnce clustering and then using those as input for a hybrid Markov model. The use of subsequence clustering in [7] generated clusters which preserved event ordering. Imran et. al [8] also employed trace clustering as a precursor to modelling, with the objective of modelling business processes at a cluster level.

References [6] and [7] identify event diversity within traces, along with data volumes as key issues to be addressed. Le et. al [7] also observed that preserving sequential integrity is a challenge for trace cluster analysis. In their analysis, [8] (also [9]) specifically observed that when processes are unstructured, analytical complexity and interpretation becomes increasingly challenging. Notwithstanding this reservation, [8] claimed that trace clustering provided more effective information retention compared to other techniques such as filtration, abstraction and pattern mining.

Analysis of traces using clustering in and of itself, while useful, is usually not the end goal for the analysis: identified clusters are often used as inputs for further analysis.

3.3 Association rule mining

Pattern mining methods are applicable across a wide range of domains as identified [10]. Two methodologies applicable to traces are association rule and sequential association rule mining. Association rule mining focusses on transaction baskets and basket pattern identification. Applied to trace analysis, a trace containing some specific tokens may have a higher probability of containing other tokens over traces which contain just some of the specific tokens. And while not strictly predictive, rule mining gives measures for the uplift in probability of a token(s) occurring.

The relative increase in probability is described by the *lift* and is central [11] in developing sequential association rule mining. There is no temporal component in association rule mining: antecedent tokens can occur prior, between or after antecedent tokens in the trace and the lift measure simply signals the relative increase for the probability of the antecedent occurring.

Mooney and Roddick [11, p. 4] note that while the Apriori algorithm [12] is typically used for "intra-transaction associations", the addition of a sequential component requires the Apriori algorithm to be extended. Extensions such as Generalised Sequential Patterns (GSP) algorithm [13] allows the identification of rules with ordering.

The inclusion of order is often a relative feature and not an absolute feature. Hence token order, while maintained, does not include a duration component, and may not imply immediately

following. Rather, time ordering implies at some time subsequent. The inclusion of ordering requires that each token has a time or order identifier.

The addition of order increases the volume of data associated with sequential association rule mining and this leads to more complex algorithms. It was observed by [11] and [14] that dependent on the sequential mining algorithm ([11] identified 13 distinct algorithms developed between 1995 and 2005), computing time and/or memory constraints can be an issue for processing. Singer and Lemmerich [15] expressed the same view and noted that SPaDE (Sequential Pattern Discovery using Equivalence Classes [16]) equivalent algorithms are more suited to dense data while PrefixSpan [17] equivalent algorithms are better suited to sparse data.

3.4 Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA) is a well-established approach within the field of natural language processing. Its primary use is to determine the latent thematic structure within a given corpus of text and LDA help users track the presence of relevant themes and topics. At its core, LDA deals with sets of symbols where each set is a document, and the symbols are words. Zhao et. al [18] apply LDA to analysing diversity in sequences of genetic data. Jelodar et. al [19] document a range of researched applications, including event segmentation in twitter feeds, diagnosis-medication analysis, and clinical process evaluation.

The primary purpose of LDA is to uncover latent thematic structures within a collection (corpus) of documents or text data [20]. LDA is a generative probabilistic model that treats each document in a corpus as a mixture of a small number of topics and each topic as a mixture of words. Traces are functionally equivalent to documents and tokens are equivalent to words. The thematic structures are then the topics, which are characterised by a list of words, with each document modelled as a probabilistically measured exposure to each topic. LDA serves to identify topics present in the corpus, the words associated with each topic, and assigns topic weightings to each document [21].

The output from LDA is a compact and interpretable representation of documents in terms of their topic distributions. Hence, LDA supports other analysis, such as clustering of documents and document classification. Jelodar et. al [19] describe LDA as a robust and malleable framework for revealing latent topics and structures in symbolic data, which makes it possible to perform analyses and develop applications to identify meaningful patterns in large collections of documents, and to organize and extract information from those documents.

3.5 Survival analysis

Survival and hazard analysis are forms of Time-to-Event (TTE) analysis and as noted by [22], is of interest insofar as this type of analysis focusses on both whether an event occurred (that is, a state change in the context of this analysis), but also the timing of the event. Reference [22] notes that unlike more standard methods of regression, TTE analysis can also accommodate data censoring.

As survival or hazard analysis is a well-developed subject, there are a range of approaches to defining appropriate probability representations for survival. At the highest level, there are nonparametric, semi-parametric and parametric methods. A common nonparametric survival analysis method uses the Kaplan-Meier estimator (useful for unbiased estimates of descriptive statistics [22]) which is an interval-based analysis. In the limit, as intervals approach zero width, the Kaplan-Meier estimator converges to the empirical cumulative distribution function. A common semi-parametric model described by [22] is the Cox Proportional Hazard model.

Parametric survival or hazard models are particularly useful as they offer predictive capabilities. As observed by [22], they can be used to predict survival times, hazard rates, and mean and median times for both. Common parametric survival distributions are the exponential, Weibull, Gompertz, log-logistic and generalised gamma. It is noted that lognormal and Weibull distributions are derivable from the generalised gamma distribution.

4.0 Exploratory Data Analysis

The dataset provided is semi-synthetic as the underlying data has been masked using symbols. Letters (symbols) have been used as tokens and "*" has been used for blank fields in the data. No meaning has been ascribed to tokens and no relationships between tokens have been provided[2] . The dataset contains 12,849 traces across 7001 columns (sequential from left to right, with a unique identifier in the first column). The dataset comprises 254 unique tokens. In construction of the dataset, many of these unique tokens were formed as a combination[3] of a core thirteen tokens (Table 1). Many of the combination tokens occur only a few times within the total 89.943 million records. This presented substantial issues for EDA, modelling and visualisation development, namely:

1. It is unknown if combination tokens are relevant to the analysis.
2. Processing for 254 tokens across ninety million records requires significant desktop computing power.
3. 254 tokens create contrast issues in visualisations.
4. Frequency scales for tokens range from one to 74.8 million, distorting frequency perception.

Table 1: Unique single character tokens

blank	D	X	H
C	F	E	K
B	J	G	I
A			

As the project is focussed on tool development and proof-of-concept, the total number of unique tokens was reduced to facilitate development. This was done through substituting either a new token for a frequent combination token or by using the first core unique token of a combination token[4]. The code developed provides for this substitution, and for the exclusion of specified traces. Such utility was explicitly required by the client. In addition to substitutions, the number of traces used was also reduced. Hereafter, this reduced dataset will be referenced as the dataset.

The new token set in the dataset (twenty unique tokens) has been formulated as follows:

- The twelve single character tokens were retained.
- Blank fields are assumed to be a state and assigned the token "*".
- The top seven most frequently occurring combination tokens have been assigned a new unique token.
- The remaining combination tokens were assigned a token defined by the first character of the combination.
- The dataset set has been filtered for traces which are greater than or equal to 2000 tokens (excluding "*" counts).

As the primary objective of the project is to develop tools and workflows and to demonstrate their use, to the extent that the reduction impacts analysis, this is secondary to tool and workflow development.

[2] This is a salient point as the tools and workflows are intended to be generally applicable to event sequences.
[3] For example, the token "CD" is formed from the core unique token "C" and "D".
[4] If "CD" was to be modified in this manner, the token would be "C".

The changes reduce the *original dataset* to twenty tokens and 3,048 traces. Dataset token frequencies and the distribution of trace lengths by token count are shown in Figures 1 and 2. Figure 1 shows that the dataset is characterised by a set of four frequent tokens.

Figure 1: Token frequencies over the dataset.

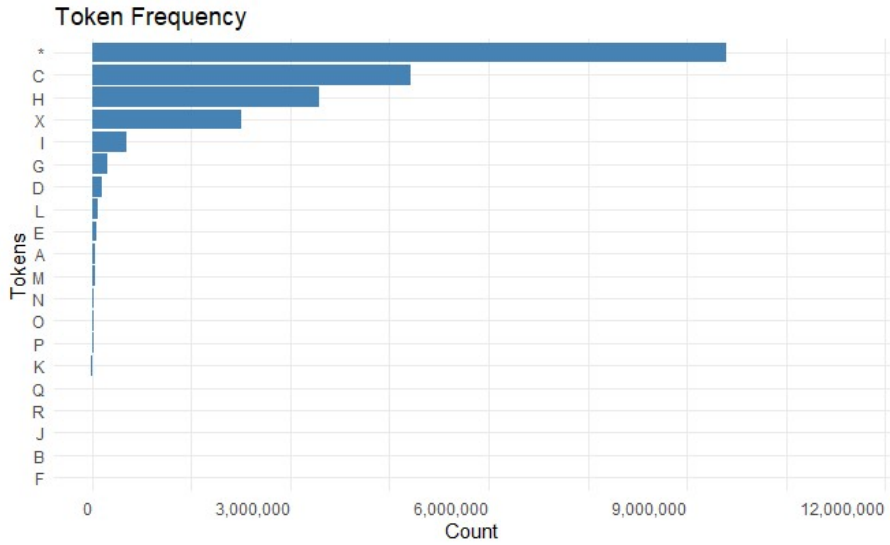
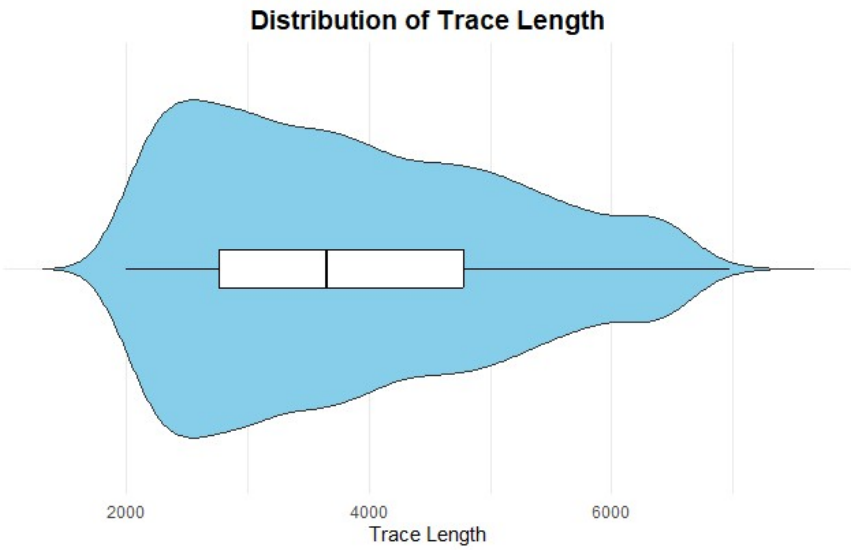
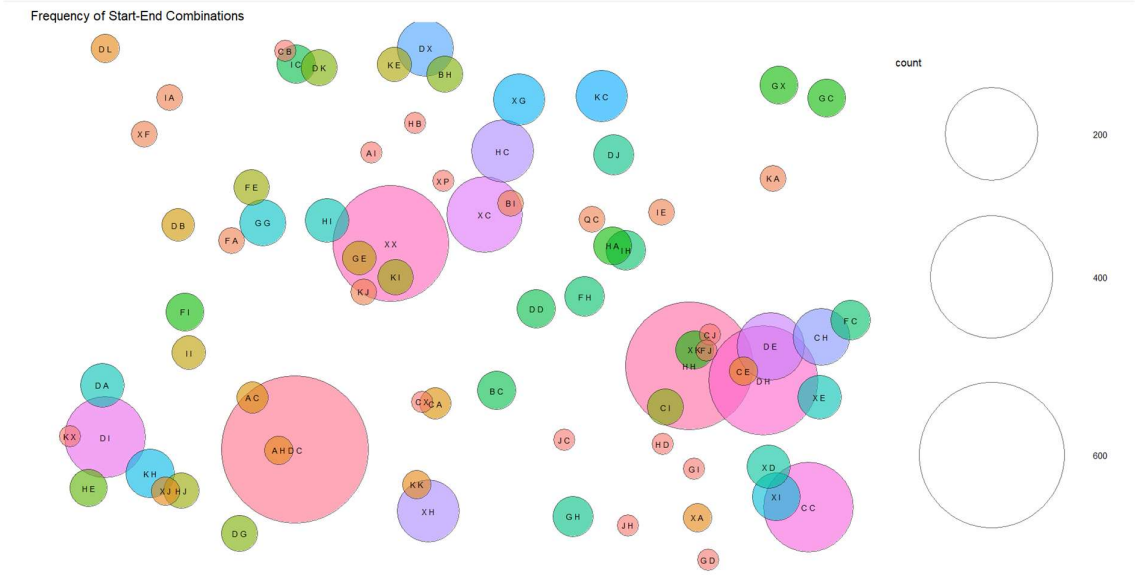


Figure 2: Violin plot of 3,046 trace lengths by token count (excluding "").*



With traces apparently comprising mostly a small set of frequent tokens, the question arises as to the distribution of tokens within traces. Figure 3 shows the relative counts for the first (start) and last (end) non "*" tokens within each trace. The most frequent tokens (Table 1) show the greatest number of trace start-end combinations with other combinations tailing off quickly.

Figure 3: Start-End token combinations (frequency)



These differences are further emphasised in Figures 4 and 5. Figure 4 shows the token distributions through time and Figure 5 is a heatmap of the tokens within traces through time. The preponderance of a few tokens evident in Figures 4 and 5 suggests that these frequent tokens appear in long contiguous sequences. Further, Figure 4 shows that "*" tokens occur more frequently towards the start and end of traces,. The majority of non-"*" tokens are concentrated between sequence positions 1500 and 5500.

Figure 4: Token distribution through time.

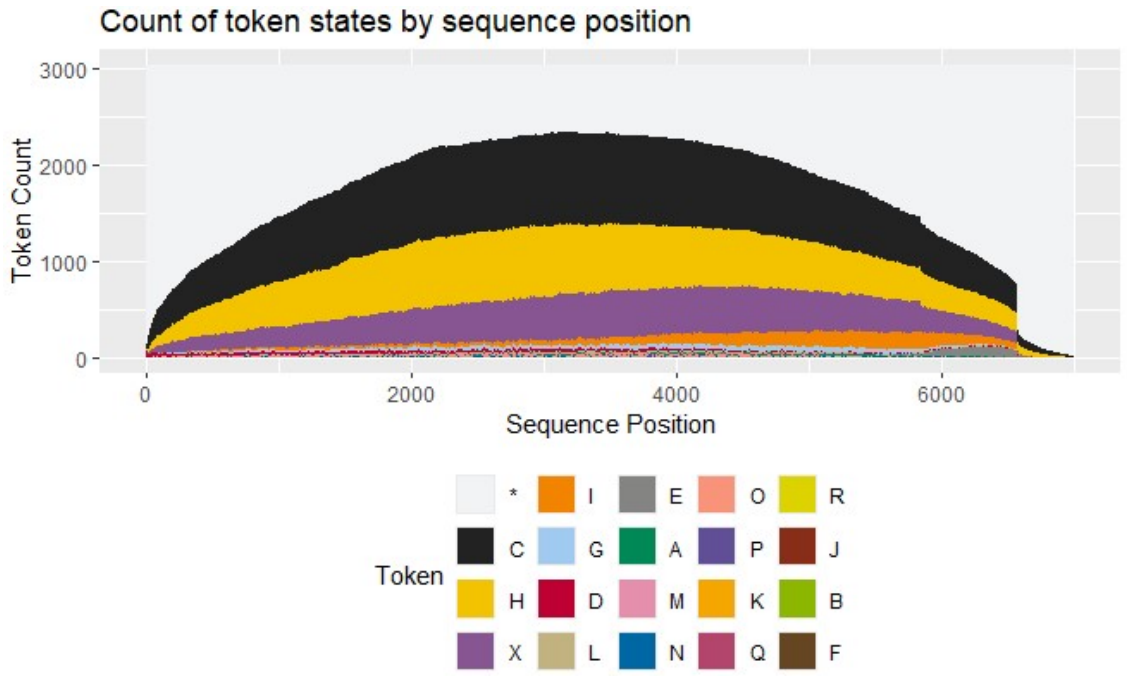
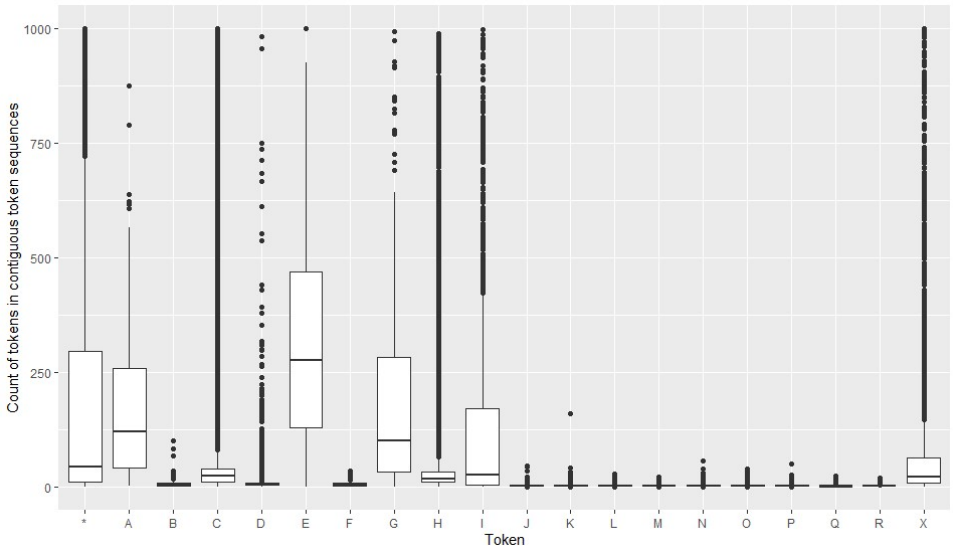


Figure 5: Trace – token heatmap (x-axis is time:1 – 7000, y-axis is individual traces).

Figure 6: Boxplots of tokens by contiguous sequence lengths (y-axis truncated at 1,000 – extends to c. 7,000).

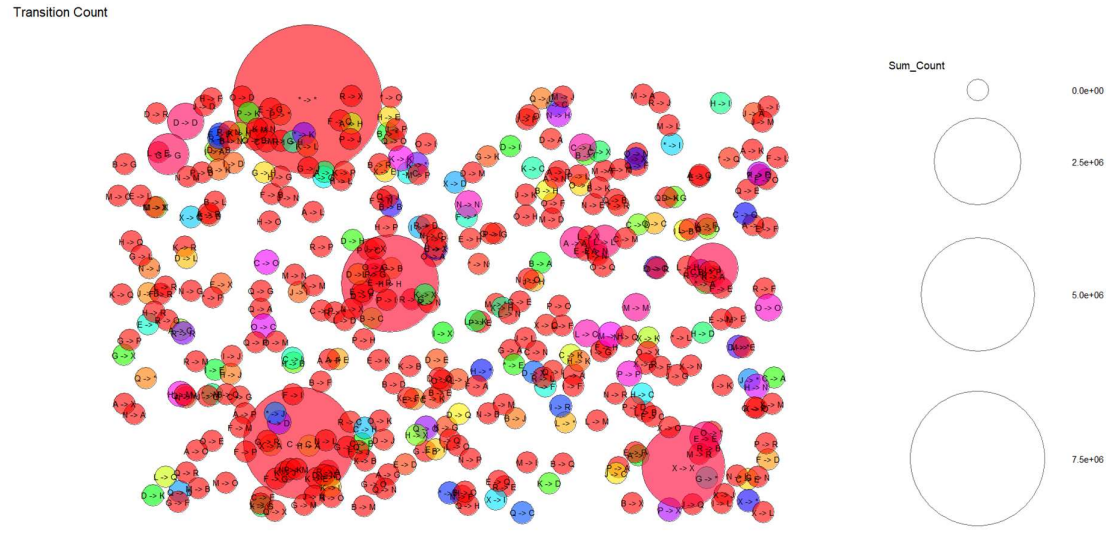


This presence of long sequences of contiguous tokens shows in Figure 6. The "*", C, G, H, I and X tokens have exceedingly long tails, signifying a broad range of contiguous sequence lengths for these tokens. It is straightforward to visualise long sequences of these tokens interspersed with shorter sequences of other, less frequent tokens. This property of the dataset has influenced further analysis, with the focus extending to transition behaviours.

Figure 7[5] shows relative transition counts between tokens. As expected from Figure 6, the tokens demonstrate a high level of self-transition, with lower levels of transitions between different tokens. Where self-transitions are lower in absolute terms, those tokens also appear to transition to a higher frequency token (*, C, H, X, I, and I). Examples of this behaviour from Figure 7 are M->H, N->H, L->C, O->C and P->X.

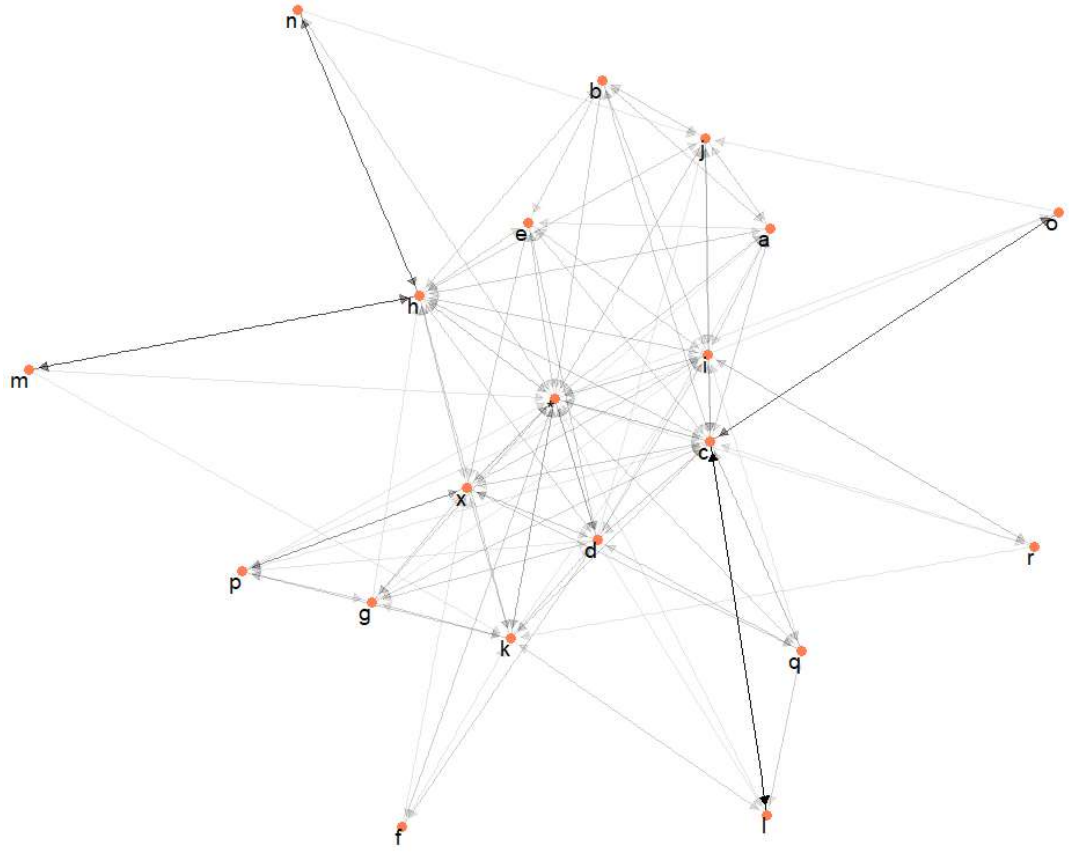
An alternative visualisation to the transition bubble plot or a transition matrix, is a directed graph with weighted edges (Figure 8, self-transitions are excluded as they dominate the visualisation). The relative darkness of each line signifies the frequency of a specific transition from a token measured against the other transitions from that token. Importantly, Figure 8 also provides a straightforward way to view the token transitions which do occur within the dataset against those that do not. For example, N->J occurs, but J->N does not. Hence the outer ring of tokens in Figure 8 are those that lie on paths less travelled while the inner ring of tokens realises a greater range of transitions.

Figure 7: Bubble plot of transition counts (including self-transitions).



[5] A bubble plot was used as a full transition matrix added c. 5% to the word count.

Figure 8: Directed graph of token transitions (excluding self-transition).



4.1 EDA Summary

This project is centred around building tools and workflows to analyse traces. The dataset provided appears to have some extreme characteristics for a general form of this type of dataset. These can be summarised as:

1. Extreme disparities between the frequencies of each token within the dataset.
2. Traces are characterised by very long contiguous sequences (having long right-hand tailed distributions) of a few highly frequent tokens.
3. Transitions between two different tokens are infrequent events, with the most common transitions between tokens occurring between the more frequent tokens.

In developing the EDA, and for subsequent model development, the project team made some simplifying assumptions which have already been documented. These were not expected to affect the analysis or modelling as both the analysis and modelling are symbolic with no underlying domain reference. In effect, the purpose of the analysis and modelling is to demonstrate that it works. As to whether it is effective, that is the province of the clients. Noting this caveat, the subsequent modelling has employed the approaches outlined in the Literature Review as these represent some of the ways in which trace analysis and modelling can be conducted.

5.0 Modelling

This section details some of the results from the model modules developed for this project as applied to the dataset. Specifically:

- The clustering module allows multiple options (number of clusters and clustering algorithm).
- The LDA module requires the number of topics and leverages a browser-based interactive visualisation tool.
- The association rule mining module covers basket analysis and its sequential equivalent.
- The survival module requires the "from" and "to" tokens for transition modelling.

Hence, not all options and features of the modules are shown in this modelling section. Where material has been included, it has been included on the basis that it showcases results.

There were issues in analysing traces comprising long contiguous sequences of the same token for patterns contained within the traces. Whether this is normal for this type of data is unknown, but some reasons for these issues were:

- Very high levels of self-transitions.
- Very few contiguous sequences of tokens were of the same length (particularly relevant for the most frequent tokens) and lengths showed considerable variability.
- The trace lengths and number of records within the dataset materially affect compute times for the analysis.

These issues have been reflected in the types and methods of analyses chosen for the modelling modules. The sequential association rule mining is closest to pure pattern mining. In contrast, the cluster analysis and LDA are classification analyses focussed on identifying similarities between traces based on feature sets. Subsequent regression analysis using exogenous variables (socio-demographic data for example) would likely use the results from the classification analysis as an input.

A unique approach adopted in this project to manage long contiguous token lengths in some modules is the categorisation (bucketing) of token lengths into distinct groups: very short, short, medium, long, and very long. These lengths are based on the distribution of all contiguous token lengths within the dataset and represent interquartile length cut-offs of 5%, 25%, 50%, 75% and 95% respectively from this distribution. Very short contiguous lengths of the same token are hence represented as a very short token (a very short contiguous sequence of C would be *C_short*) and similarly for other categories of token lengths. This data transformation was used in the association rule mining, the sequential association rule mining and the LDA modules.

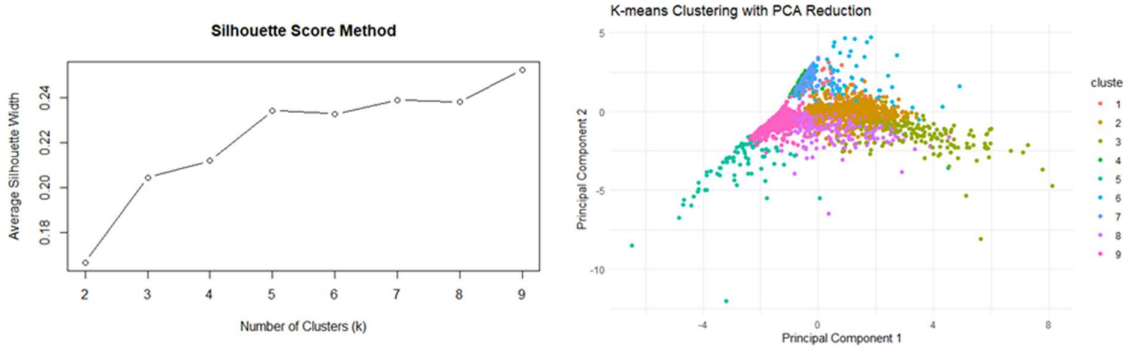
No performance measures have been developed within the analysis as the project objective was to deliver analytical tools and models for a general data case.

5.1 Clustering

The clustering module developed allows for the analysis of two distinct feature sets: token length counts per trace and token transition counts per trace. There are three clustering options provided within the module: K-means, hierarchical and distribution-based. When using the module, elbow and silhouette plots are provided for the user to support specifying the number of clusters and the user can specify the algorithm used. The subsequent analysis summarises the results from the use of the clustering module where the features are based on token counts per trace, under the K-means algorithm with nine clusters specified as per the silhouette plot (Figure 9).

Figure 9 shows the silhouette plot and the cluster plot under nine clusters. The cluster plot shows good differentiation between cluster groups within the dataset and there are some differences in the relative size of each cluster.

Figure 9: Silhouette plot (eft) and two-dimensional (9) cluster plot (right).



Using the cluster assignments shown in Figure 9 (right), Figures 10 and 11 show more detail on the trace structure for each of the clusters. The varying vertical granularity between each of the plots in Figures 10 and 11 is due to the number of traces within each cluster. Token colour assignments are consistent across all the charts in Figures 10 and 11.

The token counts per trace are the discriminating feature used in the cluster analysis and each of the clusters in both Figures 10 and 11 show predominant token types. For example, cluster 4 is distinguished by token “G”, while cluster 7 is characterised by token “X”. In contrast, clusters 1, 6 and 8 are distinguished by the presence of three primary tokens. Within each of these clusters, the dominant sequences are not all the same length and are distributed differently over the length of each trace. Moreover, the dominant tokens also have varying start and end points within each trace.

Whether these outcomes would support subsequent regression analysis is unknown. What is known, is that using the cluster analysis, the analysis was capable of classifying traces within the dataset into discrete sub-groups.

Figure 10: Trace token patterns by cluster (x-axis is time:1 – 7000, y-axis is individual traces).

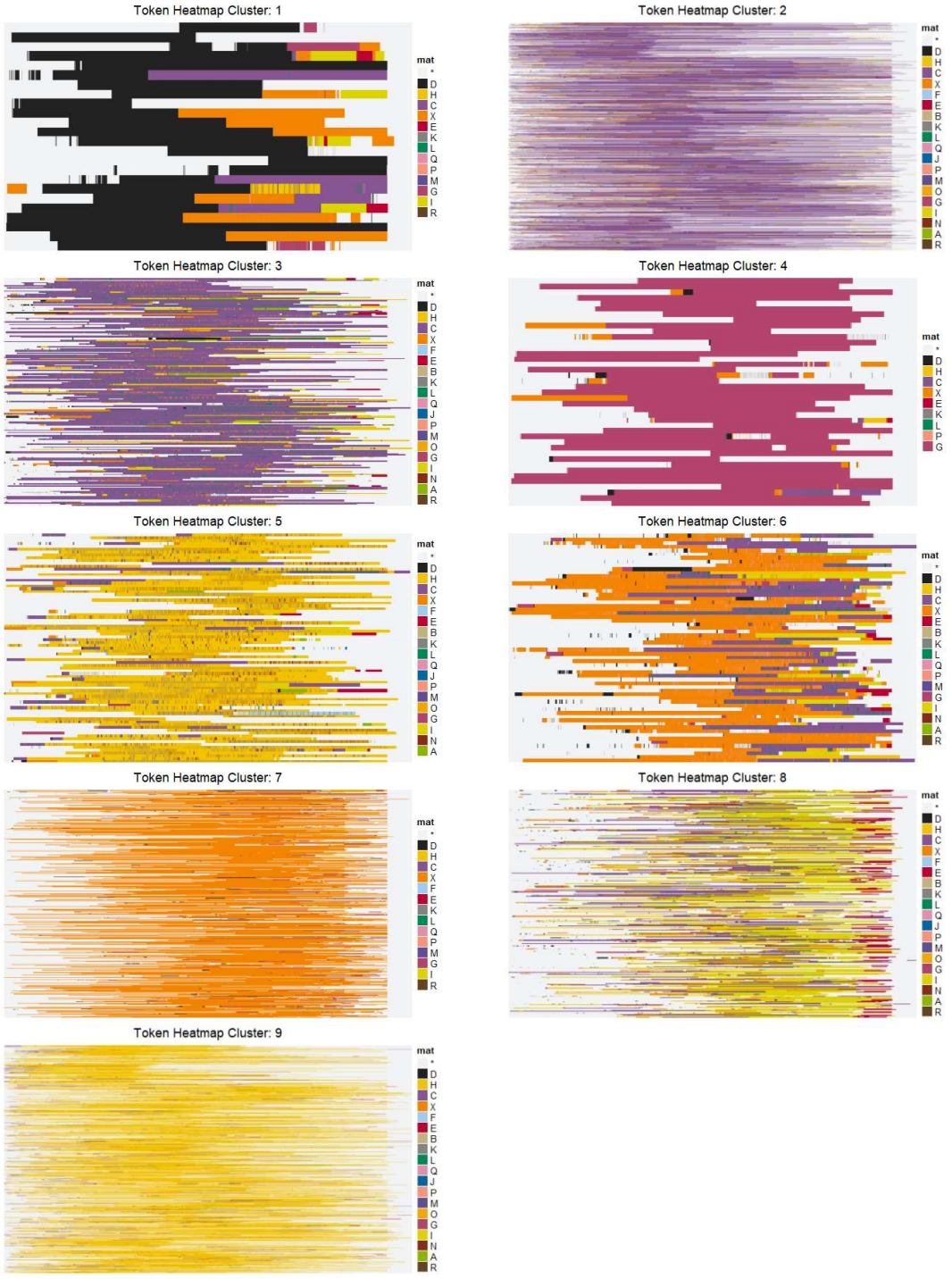
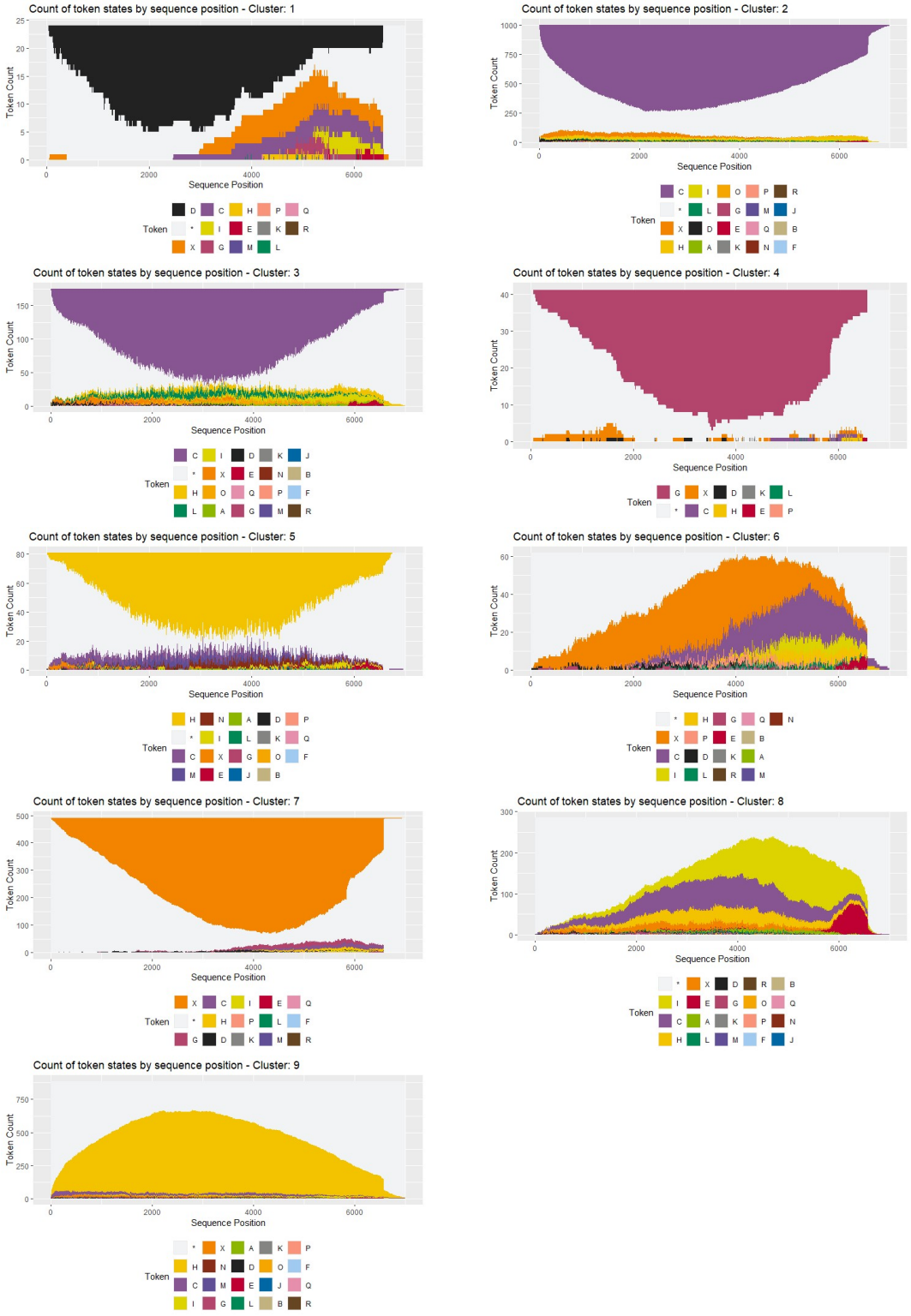


Figure 11: Trace token sequence distribution by cluster.



5.2 Association rule mining

The association rule mining module developed allows for two forms of association rule mining: basket analysis and sequence rule mining. For both forms of analysis, the user is prompted to input the support, confidence, and number of rules to be displayed. The subsequent analysis summarises the results from the association rule and sequential association rule mining in a table.

Association sequence rule mining extends the principles of association rule mining to consider the order of events, offering deeper insights into the sequential dynamics of patterns. This project employed sequence rule mining to identify ordered relationships between sequences of tokens where the tokens were categorised by contiguous token lengths such as "Very Long" or "Short" as described earlier. Each sequence rule shows how the presence of an initial set of events (LHS) influences the occurrence of subsequent events (RHS) within the dataset.

The top 5 rules from the association sequence rule mining analysis of the dataset using support of 10% and confidence of 80% are shown in Table 2. The third top rule, <{L_Short}, {L_Very Short}> -> <{L_Short}>, features in 14.9% of the traces and has a lift of 3.00 and confidence of 85.05%. This shows that if a trace has the token sequence <{L_Very Short, L_Short}> in it, then there is a considerable increase in the likelihood of observing another {<{L_Short>} token sequence following these. The results in Table 2 demonstrate a range of sequence rules highlighted by lift measures which show material increases over baseline probabilities. For instance, the lift of 6.07 for the rule <{C_Very Long},{O_Short}> -> <{O_Short}> indicates nearly a seven-fold increase in the probability of observing <{O_Short}> at some point following <{C_Very Long, O_Short}>.

Table 2: Top 10 dataset sequential association rules: Support 10%, Confidence 80%.

LHS	RHS	SUPPORT	CONFIDENCE	LIFT
<{C_VERY LONG},{O_SHORT}>	<{O_Short}>	0.1024631	0.8062016	6.076445
<{L_SHORT}>	<{L_Short}>	0.2354680	0.8308227	2.931466
<{L_VERY SHORT},{L_SHORT}>	<{L_Short}>	0.1494253	0.8504673	3.000780
<{L_SHORT},{L_VERY SHORT}>	<{L_Short}>	0.1336617	0.8011811	2.826879
<{L_VERY SHORT},{L_MEDIUM}>	<{L_Short}>	0.1313629	0.8179959	2.886208

Without specific knowledge about what each token sequence represents (the meaning of token sequences denoted by C_Long or L_Short for example), interpreting these rules requires domain expertise. Hence, this analysis provides the statistical significance of these rules but does not capture causal mechanisms or practical applications. Noting this reservation, this analysis does serve to identify a form of frequent patterns. It is important to note that association rule mining is not strictly a predictive technique, it just identifies strong relationships. In this context, association rule mining could also be viewed as pattern (motif) identification.

Finally, and separate to association rules mining, a feature embedded within the module is that the module generates an event log for the dataset. For each event, the event log comprises a unique trace ID, a bucketed token sequence, and the start and end points for that bucketed sequence. This event log structure could also be used in a process mining analysis.

5.3 Latent Dirichlet Allocation

The LDA module allows for the analysis of traces through considering each trace as a document. To facilitate this approach, bucketing of contiguous sequences as described previously was used as the dictionary in the analysis. In effect, each trace is considered a document comprised of words created from contiguous sequences of tokens. LDA is then applied to the identification of word-topic and topic-document associations. The subsequent analysis summarises the results from the use of the LDA module with five topics specified.

Using five topics, the top five bucketed tokens for each topic are shown in Table 3. The topic weights for each trace are shown in Figure 12 (darker lines represent a greater relative weight). The topics represent probabilistically distinct subjects with the dataset and from Table 3, the topics show clear differences in the top five terms.

While Figure 12 shows the relative weight of each topic within a trace, the topics do not represent clusters. Rather, the topics are abstract characterisations of a subject or domain. It is up to the client to determine whether there is a more concrete way to characterise a topic. This may be via domain knowledge, or through further analysis such as some form of regression. Whether the output would support subsequent regression analysis is unknown, but the LDA analysis was capable of classifying traces within the dataset into discrete, probabilistically weighted, sub-groups.

Table 3:Top 5 bucketed token sequences by topic.

Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
*_verylong	l_short	o_short	h_long	p_short
*_extremelylong	c_long	c_long	m_short	x_long
d_medium	c_verylong	c_verylong	h_verylong	*_long
h_extremelylong	l_medium	o_medium	n_short	x_verylong
c_extremelylong	l_veryshort	o_veryshort	n_medium	*_verylong

To support analytical interpretation, access to the LDAvis [23] interactive visualization is embedded in the LDA module. Using the output from the LDA analysis, LDAvis (as per Figures 13 and 14) shows the following:

- Global Topic Overview: A two-dimensional plot where each circle represents a topic with the distance between circles a measure of the similarity between topics.
- Term Distribution: A bar chart that displays the most relevant terms for the selected topic and ranked by their relevance to the selected topic.
- Topic Distribution: A bar chart that shows the distribution of topics across the entire corpus.
- Document Display (not shown): A list of documents in the corpus, ranked by their relevance to the selected topic.

LDAvis is browser-based and designed to provide an intuitive and interactive way to explore the topics generated by a LDA model.

Figure 12: Trace – topic heatmap: intra-trace relative weights for each topic.

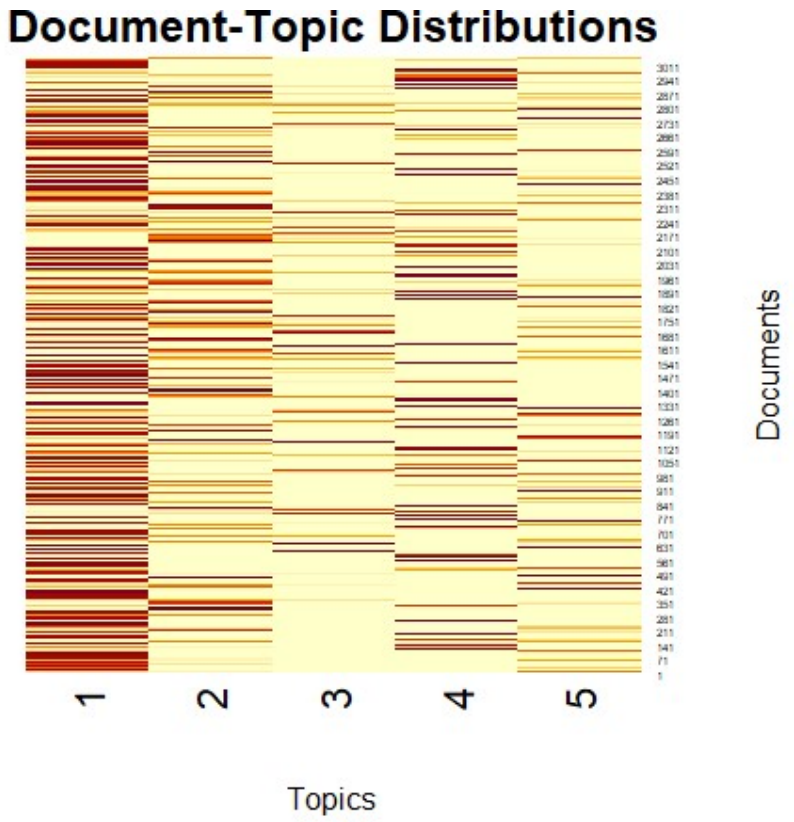


Figure 13: Interactive view of trace-topic distributions (topic 5 selected)

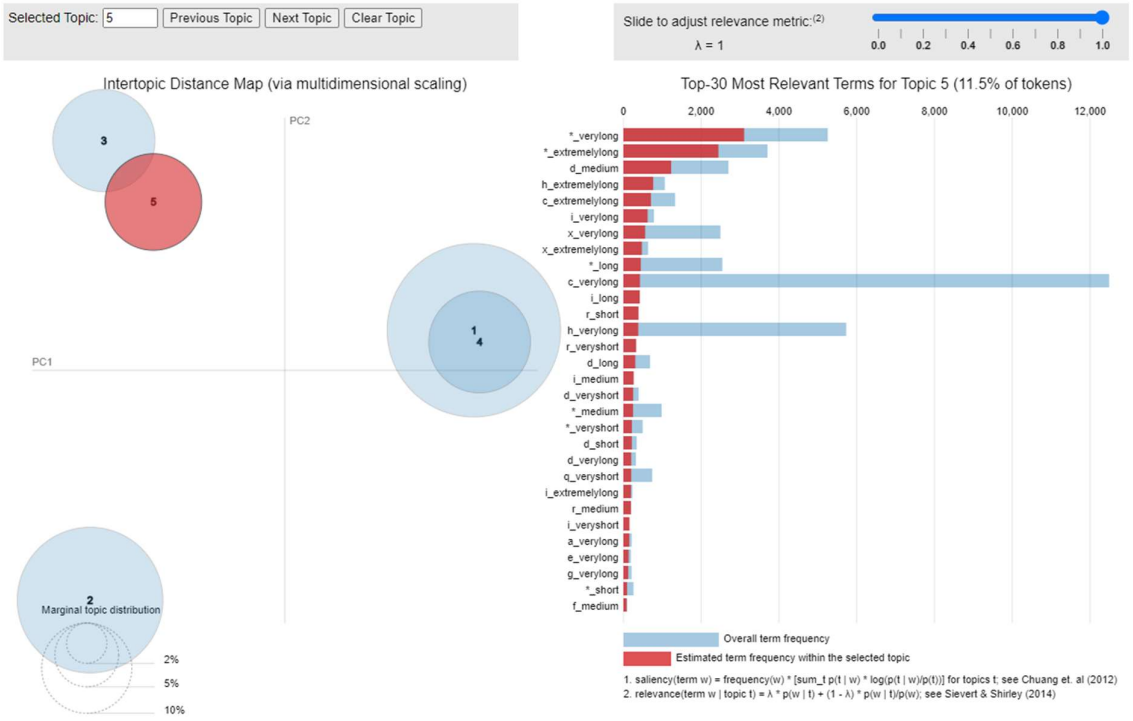
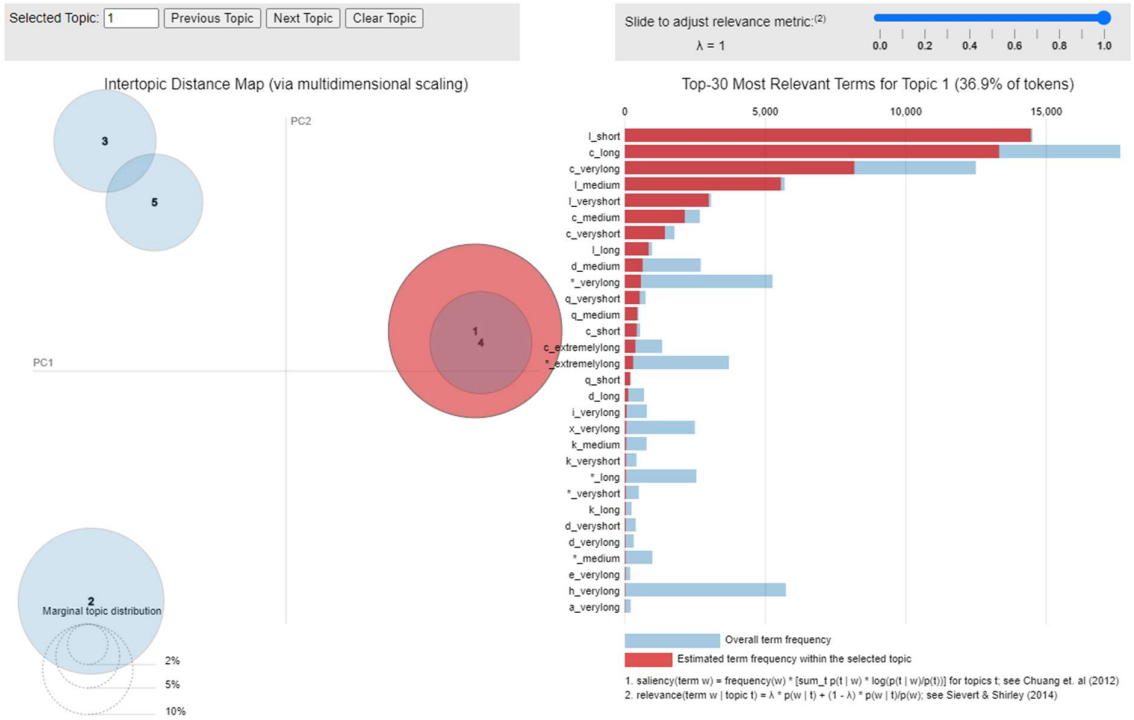


Figure 14: Interactive view of trace-topic distributions (topic 1 selected)



5.4 Survival modelling

Survival or hazard modelling for this project focuses on modelling longevity characteristics of a state prior to transition. Key for this analysis is establishing a start point from which the duration of the pre-transition event in a trace is measured. For this analysis, it starts from and includes the column in which a new state is first realised. Each subsequent right adjacent column with the same token reflects accrued time until the transition is realised (after *n* columns). When time-to-event data is analysed in this manner, it can be modelled as a survival function, probability density function, hazard function or as a cumulative hazard function.

Using a hazard curve, a sample of the module output is shown in Figure 15 which graphs the time taken for each contiguous sequence of "*" prior to "*" transitioning to a new token (and similarly for token X). The y-axis in Figure 15 is a cumulative count of the "*" and X tokens as they transition to a new token. Figure 15 reflects on a more detailed scale, the "*" and X tokens represented in Figure 6 if all the "to" tokens in Figure 15 were aggregated.

For those tokens with relatively higher frequencies, the plots are smooth curves which can be modelled using either parametric or semi-parametric models. For this project, three parametric survival models have been used: Weibull, lognormal and generalised gamma. Figure 16 shows an example of this for the survival of tokens * and X prior to transitioning to token C, with the Kaplan-Meier model included for comparison. In this example, the generalised gamma function appears to be the best fit for both cases. This conclusion is supported by the relative value of the AIC measures for each function (Tables 4 and 5).

While there is a clear benefit in assessing the probability of transitioning through time, the weakness in this analysis is that it does not predict which token an existing token will transition to.

*Figure 15: Cumulative count by token time to transition: *-> (lhs), X-> (rhs).*

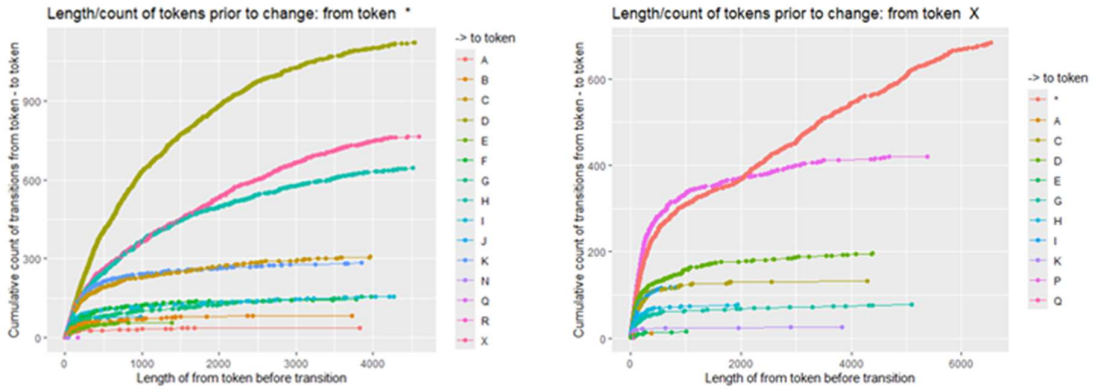


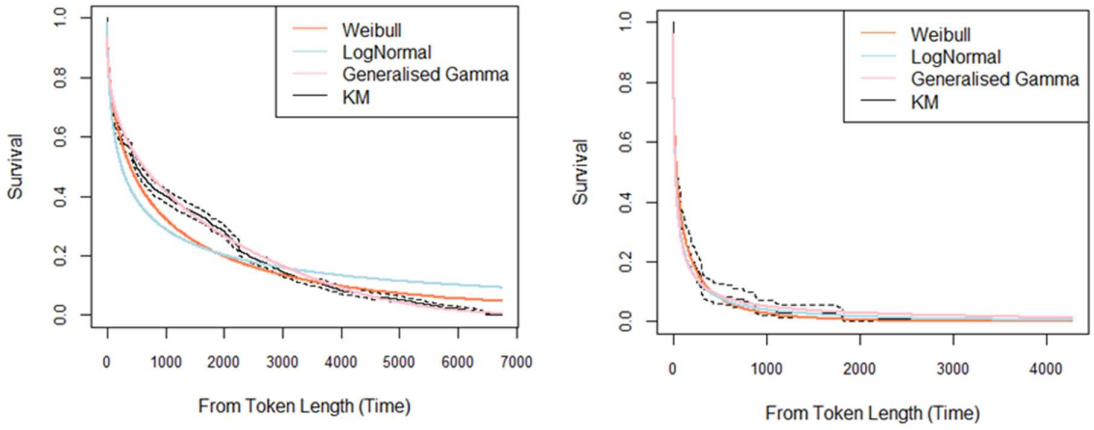
*Table 4: Fitted curve parameters and AIC measure for *->C transition survival.*

Function	*Parameters*	*AIC*
Generalised Gamma	mu: 8.0415476, sigma: -0.1443964, Q: 3.5012871	28386
Weibull	shape: -0.6436101, scale: 6.6782209	28675
Lognormal	mean log: 5.519223, std. dev. log: 0.912656	28142

Table 5: Fitted curve parameters and AIC measure for X->C transition survival.

Function	*Parameters*	*AIC*
Generalised Gamma	mu: 2.9234590, sigma: 0.6735809, Q: -0.4190547	3145
Weibull	shape: -0.6827088, scale: 4.3791012	3103
Lognormal	mean log: 3.3463846, std. dev. log: 0.7099869	3101

*Figure 16: Fitted survival curves (inc KM): *->C (lhs), X->C (rhs).*



6.0 Module Outline

This section outlines the purpose for each module and provides a diagram (Figure 17) of the overall workflow. Internal module workflow descriptions occur as R markdown code blocks.

R markdown module files and data should be stored in the same directory and this directory must be set as the R working directory. Output files are stored to this directory.

The project team developed five analyses and modelling modules for this project. Further details on required data formats and structure, key dependencies and R libraries, along with documentation are embedded within the modules. R markdown was used as the framework for the modules as this facilitated the inclusion of documentation within each module.

The developed modules are:

1. EDA modules
2. Clustering module
3. Association rule mining module
4. LDA module
5. Survival module

6.1 Input data requirements

In order for the modules to work, the initial input data must be in the format as shown in Table 6 as a csv file. The first column in the data is a unique identifier, the subsequent columns must be ordered sequentially from left to right and contain a single character representation of a token (or the field can be blank). The data must not have a header row and each row in the data should represent one unique trace. The modules can accommodate up to 22 unique tokens within the traces, including blank fields. This restriction is in place to support contrasting colours in the heatmaps and token sequence count plots.

Table 6: Example input data format (tokens are X, C, A, D, J and V).

UniqueID_1	X	C	A	D	J
UniqueID_2		V	X	D	A
…	…	…	…	…	…
UniqueID_n	C			J	D

6.2 EDA module

The EDA module generates a set of visuals and tables for the dataset.

These inform the user of the general characteristics of the data. On conclusion of the first run through the data using the data formatting module, the module generates a csv file (“replace_file.csv”) which the user can use to manually configure the data for subsequent use. This configuration includes token re-assignment and trace selection. Once the configuration file has been modified, a new file called “mod_r_data.csv” is generated using the configuration options and saved. The “mod_r_data.csv” file can then be run through the four EDA modules for a more detailed view of the data. These views include boxplots, state transition and heatmap plots, directed graphs and matrices for transition views and barplots, among a range of available visuals.

The user can re-run the data formatting module and make futher configuration changes to data if required based on the results from the four EDA modules. The configured data ("mod_r_data.csv") is used in the modelling modules.

6.3 Clustering module

The clustering module imports the modified dataset file and provides the user with three forms of feature clustering:

1. Clustering on token counts by trace.
2. Clustering on token transition counts by trace (including self- transitions).
3. Clustering on token transition counts by trace (excluding self- transitions).

One the feature choice for clustering is made by the user, the module generates elbow and silhouette plots for the user to determine optimal cluster counts. Once the user specifies the cluster count, the user is prompted for the type of cluster algorithm to be applied:

1. K-means.
2. Hierarchical.
3. Distribution based.

On selection of the algorithm, the module generates the clusters, prints a table of basic cluster statistics and a cluster plot, generates heatmaps and state-trace counts for each cluster and outputs a csv file containing trace-cluster associations.

6.4 Association rule mining module

The association rule mining module imports the modified dataset file and provides the user with two forms of rule mining:

1. Association rule mining where each trace is considered a basket and there is no ordering of token sequences.
2. Association rule mining where each trace is considered a basket and there is ordering of token sequences.

Once the rule mining approach is selected by the user, the user is prompted for inputs for support, confidence and the top number of rules to be displayed in a table. The sequential rule mining algorithm is computationally intensive and a low support parameter will materially extend run times.

6.5 LDA module

The LDA module imports the modified dataset file and provides the user with the option to select the number of topics for the topic modelling.

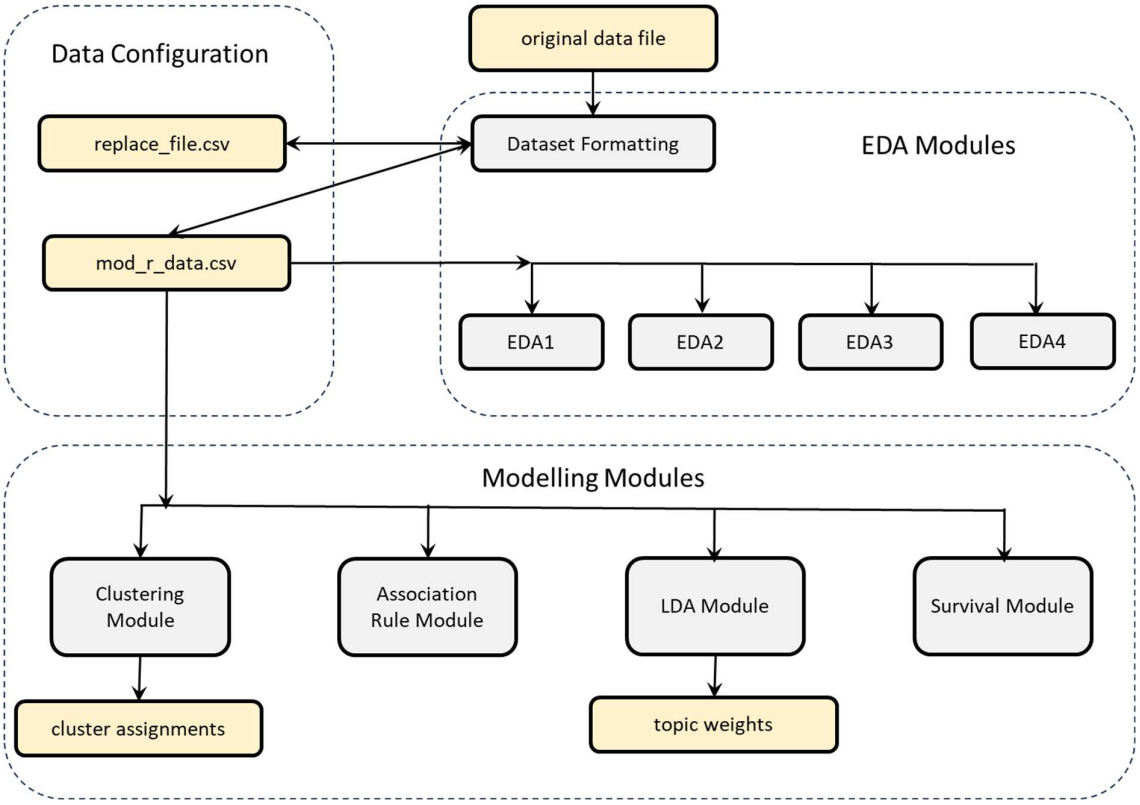
Once the analysis is complete, the LDA module prints summary tables of the top words relating to each topic. In addition, the LDA then generates a browser based interactive visualisation of the results. This allows the user to more clearly identify the differences between each topic (that is, relative word weightings between each topic group). Finally, the module outputs a csv file linking traces to a key topic and individual topic weights.

6.6 Survival module

The Survival module imports the modified dataset file and then generates the hazard curves for all token transitions (excluding self-transitions). Following this, the user is offered the option to select a specific token transition combination for survival curve fitting. There are three types of curves fitted:

lognormal, Weibull and generalised gamma. On fitting, the survival module generates the fitted curves and tables showing the AIC for each curve and the parameters for each curve.

Figure 17: Diagrammatic representation of the modules as a workflow.



7.0 Conclusion

The results from the EDA showed a dataset that was skewed to a small set of highly frequent tokens. These generally occurred as contiguous sequences which were separated by less frequent tokens. Blank fields wre most frequent fields within the dataset these occurred more frequently towards the start and end of traces. The majority of non-blank tokens were concentrated between sequence positions 1500 and 5500 (from Figure 4). The distribution of contiguous token lengths was long right tailed.

Due to lengthy contiguous token sequences, transitions between different tokens were relatively infrequent events. However, the more frequent and varied sets of token transitions (excluding self-transitions) occurred between the more frequent tokens and most of the other tokens. The less frequently occurring tokens transitioned between fewer tokens.

Once the analysis progressed to modelling, further insights were developed. In the cluster analysis, traces were separable into discrete clusters. Although the cluster analysis shown used token counts, the results were not dissimiliar to those seen under OM[6], suggesting that in at least this dataset, token counts were comparable to distance measures derived under OM.

The sequential association rule mining analysis showed ordered rules in the traces under high support and with high confidence. However, run-times for sequential association rule mining increase considerably for lower support values.

The LDA analysis showed reasonably distinct topics embedded within the dataset, formed from quite distinct words. For both the sequential association rule mining and the LDA analysis, token sequences were bucketed using interquatile length ranges to facilitate analysis. This approach simplified the analysis, but the information content from doing so needs be evaluated by the client.

The survival module leveraged the long right-tailed distributions of contiguous token lengths. In many cases, a parametric survival model fitted the data well. However, for the less frequent tokens, further development should include semi-parametric or non-parametric methods.

Further development work could progress in a number of directions:

1. Applying OM to the dataset to add a new feature dimension to the distance measures between traces, further developing the cluster analysis.
2. Analysing the dataset more formally using process mining techniques to find process patterns (noting that the association rule mining module generates an event log).
3. Applying process mining techniques at a cluster level.
4. Applying non- and semi-parametric models in the survival analysis.

Finally, in the absence of domain knowledge, no inferences can be made of the results.

[6] OM was not progressed as a distance measure for clustering due to excessive run-times for the analysis.

Bibliography

[1] A. Abbott, "Sequence Analysis: New Methods for Old Ideas," *Annual Review of Sociology,* vol. 21, pp. 93-113, 1996.

[2] M. Studer and G. Ritschard, "A comparative review of sequence dissimilarity measures," *LIVES Working papers,* vol. 033, pp. 1-47, 2014.

[3] T. Biemann, M. Mühlenbock and K. Dlouhy, "Going the distance in vocational behavior research: Introducing three extensions for optimal matching analysis based on distances between career sequences," *Journal of Vocational Behavior,* vol. 119, p. 103399, 2020.

[4] K. Dlouhy and T. Biemann, "Optimal matching analysis in career research: A review and some best-practice recommendations," *Journal of Vocational Behavior,* vol. 90, pp. 163-173, 2015.

[5] M. Studer, G. Ritschard, A. Gabadinho and N. Muller, "Discrepancy Analysis of State Sequences," *Sociological Methods & Research,* vol. 40, no. 3, pp. 471-510, 2011.

[6] X. Lu, S. A. Tabatabaei, M. Hoogendoorn and H. Reijers, "Trace Clustering on Very Large Event Data in Healthcare Using Frequent Sequence Patterns," in *Business Process Management*, Springer International Publishing, 2019, pp. 198-215.

[7] M. Le, D. Nauck, B. Gabrys and T. Martin, "Sequential Clustering for Event Sequences and Its Impact on Next Process Step Prediction," in *Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU 2014)*, Cham, 2014.

[8] M. Imran, M. Akmar Ismail, S. Hamid and M. Hairul Nizam Md Nasir, "A TRACE CLUSTERING FRAMEWORK FOR IMPROVING THE BEHAVIORAL AND STRUCTURAL QUALITY OF PROCESS MODELS IN PROCESS MINING," *Malaysian Journal of Computer Science,* vol. 36, no. 3, pp. 223-241, 2023.

[9] M. Imran, M. Akmar Ismail, S. Binti Hamid and M. Hairul Nizam Md Nasir, "Complex Process Modeling in Process Mining: A Systematic Review," *IEEE Access,* vol. 10, pp. 101515-101536, 2022.

[10] M. Gupta and J. Han, "Applications of Pattern Discovery Using Sequential Data Mining," in *Pattern Discovery Using Sequence Data Mining: Applications and Studies*, IGI Global, 2011, pp. 1-23.

[11] C. Mooney and J. Roddick, "Sequential pattern mining -- approaches and algorithms," *ACM Computing Surveys (CSUR),* vol. 45, pp. 19:1-19:39, 2013.

[12] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules," in *20th International Conference on Very Large Data Bases, (VLDB)*, Santiago, 1994.

[13] R. Srikant and R. Agrawal, "Mining sequential patterns: Generalizations and performance improvements," in *5th International Conference on Extending Database Technology (EDBT'96)*, Avignon, 1996.

[14] M. Skenduli, C. Loglisci, M. Ceci, M. Biba and D. Malerba, “An Empirical Evaluation of Sequential Pattern Mining Algorithms,” in *Advances in Internet, Data & Web Technologies*, Zurich, Springer International Publishing AG, 2018, pp. 615-626.

[15] P. Singer and P. Lemmerich, “Analyzing Sequential User Behavior on the Web,” 12 April 2016. [Online]. Available: https://sequenceanalysis.github.io/slides/analyzing_sequential_user_behavior_part2.pdf. [Accessed 6 April 2024].

[16] M. Zaki, “SPADE: An efficient algorithm for mining frequent sequences,” *Machine Learning,* vol. 42, no. 1, pp. 31-60, 2001.

[17] J. Pei, J. Han, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal and M. Hsu, “PrefixSpan,: mining sequential patterns efficiently by prefix-projected pattern growth,” in *Proceedings 17th International Conference on Data Engineering*, Heidelberg, 2001.

[18] W. Zhao, J. Chen, R. Perkins, Y. Wang, Z. Liu, H. Hong, W. Tong and W. Zou, “A novel procedure on next generation sequencing data analysis using text mining algorithm,” *BMC Bioinformatics,* vol. 17, pp. 1-15, 2016.

[19] H. Jelodar, Y. Wang, C. Yuan, X. Feng, X. Jiang, Y. Li and L. Zhao, “Latent Dirichlet allocation (LDA) and topic modeling: models, applications, a survey,” *Multimedia Tools and Applications,* vol. 78, no. 11, pp. 15169-15211, 2019.

[20] J. Kacprzyk, V. Balas and M. Ezziyyani, “Exploration, Sentiment Analysis, Topic Modeling, and Visualization of Moroccan Twitter Data,” in *Advanced Intelligent Systems for Sustainable Development (AI2SD’2020)*, Cham, 2022.

[21] D. Maier, D. Waldherr, P. Miltner, G. Wiedemann, A. Niekler, A. Keinert, B. Pfetsch, G. Heyer, U. Reber, T. Häussler, H. Schmid-Petri and S. Adam, “Applying LDA Topic Modeling in Communication Research: Toward a Valid and Reliable Methodology,” *Communication Methods and Measures,* vol. 12, no. 2-3, pp. 93-118, 2018.

[22] Columbia University Mailman School of Public Health, “Time-To-Event Data Analysis,” Columbia University Irving Medical Centre , [Online]. Available: https://www.publichealth.columbia.edu/research/population-health-methods/time-event-data-analysis#readings. [Accessed 15 4 2024].

[23] C. Sievert and K. Shirley, “LDAvis: A method for visualizing and interpreting topics,” in *Proceedings of the Workshop on Interactive Language Learning, Visualization, and Interfaces*, Stanford, 2014.

[24] A. Ganadinho, G. Ritschard, N. Muller and M. Studer, “Analyzing and Visualixing State Sequences in R with TraMineR,” *Journal of Statistical Software,* vol. 40, no. 4, pp. 1-37, 2011.

[25] A. Gabadinho, G. Ritschard, N. Müller and M. Studer, “ Analyzing and Visualizing State Sequences in R with TraMineR,” *Journal of Statistical Software,* vol. 40, no. 4, pp. 1-37, 2011.

[26] C. Brzinsky-Fay, U. Kohler and M. Luniak, “Sequence Analysis with Stata,” *The Stata Journal,* vol. 6, no. 4, pp. 435-460, 2006.

[27] B. Halpin, “SADI: Sequence Analysis Tools for Stata,” *The Stata Journal,* vol. 17, no. 3, pp. 546-572, 2017.

[28] P. Kumar, P. Radha Krishna and S. Bapi Raju, “Applications of Pattern Discovery Using Sequential Data Mining,” in *Pattern Discovery Using Sequence Data Mining: Applications and Studies*, IGI Global, 2011, pp. 1-23.