

## Temporal Text Mining: A Thematic Exploration of Don Quixote

Ray Wright, SAS Institute Inc.

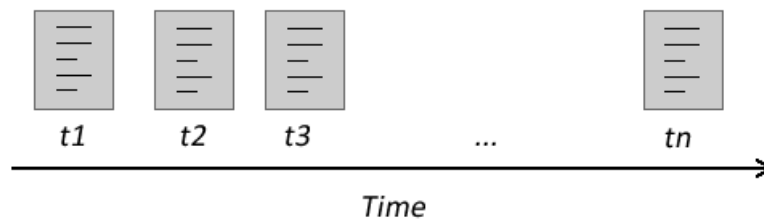
### ABSTRACT

Temporal text mining (TTM) is the discovery of temporal patterns in documents that are collected over time. It involves discovery of latent themes, construction of a thematic evolution graph, and analysis of thematic patterns. This paper uses text mining and time series analysis techniques to explore *Don Quixote de la Mancha*, a two-volume master work of Western literature. First, it uses singular value decomposition in SAS® Text Miner to discover 25 key themes that characterize the two volumes. Then it treats the chapters of the two books as time-ordered documents and creates a semiautomated visual summary of the two volumes. It also explores the trajectory of individual themes over the course of the chapters and identifies episodes, recurring themes, and climaxes. Finally, it uses time series clustering in SAS® Enterprise Miner™ to group chapters that have similar themes and to group themes that have similar trajectories. The TTM methods demonstrated in this paper lend themselves to business applications such as monitoring changes in customer sentiment and summarizing research and legislative trends.

### INTRODUCTION

Text mining is the processing of document collections to extract underlying themes or concepts. Often the goal of text mining is to transform unstructured text (such as survey responses) into structured data that can be used in predictive modeling.

Temporal text mining (TTM) adds a time dimension to text mining; it is the discovery and analysis of temporal patterns in documents that are collected over time, as illustrated in Figure 1.



**Figure 1. Time-Ordered Documents**

TTM uses a combination of text mining and time series analysis techniques. First, traditional text mining techniques are used to extract latent themes and their relevance weights for each time point. (If you are new to text mining, see the review of text topic extraction in the next section.) Once the themes are extracted, an analyst can explore temporal patterns and construct a thematic timeline by plotting the relevance weights over the time dimension. Formal analysis of temporal patterns can be carried out using time series methods such as time series clustering, dimension reduction, and forecasting. These techniques can be automated to a large extent, making it possible to process large document collections with little or no human intervention.

This paper illustrates the use of TTM methods for exploring the two volumes of *Don Quixote de la Mancha*. After a brief overview of the main characters and story, the paper describes the use of SAS Text Miner to import the text and derive key themes that characterize the text. Then it explores the trajectory of individual themes over the course of the chapters and describes the use of time series techniques available in SAS Enterprise Miner to automate the grouping and filtering of themes. Finally, some alternative ways to derive themes are considered.

## TEXT TOPIC EXTRACTION

The extraction of underlying concepts or themes from documents is at the heart of most text mining problems, whether the ultimate goal is to summarize the themes or to build a predictive model by using derived themes as inputs. The ability to summarize a document collection by a relatively small number of themes, as opposed to a large number of themes or individual terms, requires a powerful dimension-reduction technique. (If you are already familiar with text topic extraction, you might want to skip this section.)

Topic extraction is a two-step process: first the raw text is transformed to a sparse numeric representation, and then concepts are extracted by factoring the sparse matrix. Table 1 shows a simple example that uses four documents, each of which is a brief product review.

Doc. #	Document Text
1	"Good experience"
2	"It was well worth the money"
3	"Worth every penny"
4	"Meh"

**Table 1. Document Collection**

Each document in the collection contains a certain number of terms. The document collection can be transformed to a sparse matrix by counting the number of occurrences of each term for each document. These counts become the cell values in the document-by-term matrix, as shown in Table 2.

Doc. #	Term										
	every	experience	good	it	meh	money	penny	the	was	well	worth
1	0	1	1	0	0	0	0	0	0	0	0
2	0	0	0	1	0	1	0	1	1	1	1
3	1	0	0	0	0	0	1	0	0	0	1
4	0	0	0	0	1	0	0	0	0	0	0

**Table 2. Document-by-Term Matrix**

You probably notice that the document-by-term matrix is quite wide and sparse even though the number of documents in this collection is small. There are more terms than documents, and most of the terms appear in only one document. This pattern is quite common in text mining. Typically, "noise words" such as articles and prepositions are filtered out to reduce the number of columns of the sparse matrix. But even after filtering out noise words, you can imagine that with hundreds or thousands of free-form reviews, the matrix could still be extremely wide and sparse. Moreover, the sparse representation is not a very useful summary for a human analyst.

Fortunately, it is possible to factor the sparse document-by-term matrix into a much more manageable and informative representation of the document collection by using a method called singular value decomposition (SVD). The factored representation consists of three matrices: a term-by-topic matrix, a matrix of topic importance values, and a document-by-topics matrix. Figure 2 through Figure 5 show the factored representation of the example document collection. These results were obtained using the SVD subroutine in SAS/IML.

Figure 2 shows the term-by-topic matrix, in which each row corresponds to a term and columns represent the latent themes.

	v										
	COL1	COL2	COL3	COL4	COL5	COL6	COL7	COL8	COL9	COL10	COL11
ROW1	0.1154273	0.5827673	0	0	0.8043997	0	0	0	0	0	0
ROW2	0	0	0.7071068	0	0	-0.16381	0.5173635	-0.16381	-0.16381	-0.16381	-0.353553
ROW3	0	0	0.7071068	0	0	0.1638101	-0.517364	0.1638101	0.1638101	0.1638101	0.3535534
ROW4	0.3812306	-0.176448	0	0	0.0731272	-0.37136	-0.12864	-0.37136	-0.37136	-0.37136	0.5
ROW5	0	0	0	1	0	0	0	0	0	0	0
ROW6	0.3812306	-0.176448	0	0	0.0731272	0.8603023	0.1396977	-0.139698	-0.139698	-0.139698	0
ROW7	0.1154273	0.5827673	0	0	-0.438763	0.0698489	0.4301511	0.0698489	0.0698489	0.0698489	0.5
ROW8	0.3812306	-0.176448	0	0	0.0731272	-0.139698	0.1396977	0.8603023	-0.139698	-0.139698	0
ROW9	0.3812306	-0.176448	0	0	0.0731272	-0.139698	0.1396977	-0.139698	0.8603023	-0.139698	0
ROW10	0.3812306	-0.176448	0	0	0.0731272	-0.139698	0.1396977	-0.139698	-0.139698	0.8603023	0
ROW11	0.496658	0.4063195	0	0	-0.365636	-0.069849	-0.430151	-0.069849	-0.069849	-0.069849	-0.5

**Figure 2. Term-by-Topic Matrix**

The elements of this matrix can be interpreted as relevance weights—they describe the relationship of each term to each topic—and these relationships help you interpret the derived themes. Each theme is a linear combination of terms, so it is customary to label the topics by using the terms that have the highest weights. For example, for topic 2 (COL 2) the terms that have the highest weights are “every,” “penny,” and “worth” (the 1st, 7th, and 11th terms, respectively, in the sparse matrix), so the topic would be labeled “every, penny, worth.” Often a human analyst provides more descriptive names for derived topics by manually summarizing their key terms (for example, the analyst might summarize “apple,” “orange,” “banana” as “fruit”).

The diagonal entries of the singular values matrix constitute a vector of importance values, as shown in Figure 3.

s
2.5105329
1.6423229
1.4142136
1
7.609E-17
0
0
0
0
0
0

**Figure 3. Singular Values**

Each singular value is associated with a latent theme: the first singular value corresponds to theme 1, the second to theme 2, and so on. The singular values can be used to rank the latent themes by their importance.

Figure 4 shows the document-by-topic matrix, in which each row represents a document and the columns are latent themes.

	u										
	COL1	COL2	COL3	COL4	COL5	COL6	COL7	COL8	COL9	COL10	COL11
ROW1	0	0	1	0	0	0	0	0	0	0	0
ROW2	0.957092	-0.289784	0	0	0	0	0	0	0	0	0
ROW3	0.2897841	0.957092	0	0	0	0	0	0	0	0	0
ROW4	0	0	0	1	0	0	0	0	0	0	0

**Figure 4. Document-by-Topic Matrix**

Elements of the document-by-topic matrix can be interpreted as relevance weights—they describe how relevant a particular topic is to a particular document. For example, the first latent theme is most strongly associated with document 2, whereas the second theme is associated with document 3.

Notice that the document-by-topic matrix is just as wide as the sparse representation of the document collection (that is, there is one column per term). So no “savings” has been achieved by factorizing the sparse matrix. However, the document-by-topic matrix can be compressed by simply dropping the columns that have small importance values. The result is a truncated matrix that approximates the full document-by-topic matrix but has a relatively small number of topics. For this example, four latent topics suffice to describe the example document collection, as shown in Figure 5.

uPrime				
0	0	1	0	
0.957092	-0.289784	0	0	
0.2897841	0.957092	0	0	
0	0	0	1	

**Figure 5. Truncated Document-by-Topic matrix**

## THE TEXT

*Don Quixote de la Mancha* was written by Miguel de Cervantes Saavedra and published in two separate volumes in 1605 and 1615, respectively. The two volumes include a total of 126 chapters.

The following tables summarizes the main characters and provide a brief storyline.

<b>Protagonists</b>	Alonso Quixano (also known as Don Quixote and hereafter referred to as Don Q.)	Self-appointed knight errant who is inspired to action by his obsessive reading of fantasy adventure books.
	Sancho Panza	Loyal and gullible squire of Don Q.
<b>Supporting Characters</b>	Dulcinea del Toboso	In the eyes of Don Q., a fair maiden worthy of his knightly service. In reality, a common laborer (Aldonza Lorenzo) from a nearby town.
	Niece and Housekeeper	Don Q.'s closest relative and caretaker, respectively.
	Priest, Barber, and Sansón Carrasco	Loyal friends who conspire to bring Don Q. home and out of trouble.

**Table 3. Main Characters**

<b>Book 1</b>	Don Q. sets out on horseback to right wrongs and defeat terrible monsters in the name of Dulcinea del Toboso. Along the way Don Q. is knighted in a mock ceremony. Needing a squire, he enlists the help of Sancho Panza, to whom Don Q. promises tremendous rewards for his service. Many adventures ensue, the most famous being Don Q.'s iconic battle with a terrible "monster," which is actually a windmill. Throughout the course of his adventures, Don Q. suffers many losses and grows increasingly mentally unstable. Ultimately, the priest and barber fetch Don Q. from the Sierra Morena and bring him home to convalesce.
<b>Book 2</b>	Years later, Don Q. and Sancho are restless for adventure. They set out again, but word of their exploits has spread via publication of the first volume. Now famous throughout the land, they are received at castle of a Duke and Duchess who mockingly name Sancho governor of a small territory. Although Sancho is initially overjoyed, he ultimately grows tired of the responsibility (and mockery) and renounces his position in favor of the simple life he once had. He and Don Q. travel on to Barcelona, where Don Q. meets his match in a battle with the Knight of the White Moon (in reality, Sansón Carrasco).

**Table 4. Storyline**

## DERIVED THEMES

The first step in the analysis of the text was to extract the most important themes from the two books. SAS Text Miner enables you to extract themes by using a simple drag-and-drop process flow, as shown in Figure 6.



**Figure 6. Topic Extraction Flow**

A part of the SAS Enterprise Miner Suite, SAS Text Miner seamlessly integrates text analysis with traditional mining tools for structured data. Notice that no linear algebra or coding is needed to transform unstructured documents to themes.

In this process flow, the Text Import node reads the source text, treating each chapter as a document. The node can read online documents directly, but because the online source for this paper included many dedications and prologues, it was downloaded locally and front matter was excluded before the resulting text was imported into the Text Import node.

The Text Parsing node parsed the raw text into terms. Because the source text is in Spanish, the Spanish language options were used for parsing.

Finally, the Text Topic node was used to derive 25 multiterm topics. The node uses SVD to extract the topics, but there are some differences between how the Text Topic node and SAS/IML perform SVD. In particular, Text Miner rotates the dimensions of the term-by-topic matrix in order to optimize the discrepancy between absolute weights.

The derived topics are shown in Table 5. Each topic is a linear combination of terms. The terms were translated to English, and the topics were labeled subjectively.

Topic	Number of Chapters
Pedro the puppeteer and his magical monkey	3
Camacho's wedding	4
Don Q.'s battle with the Knight of the White Moon	5
The story of Anselmo	6
Zoraida and the prisoner	6
Gristósomo and his love for Marcela	6
Governor Sancho	7
Separated lovers reunited: Luscinda and Cardenio, Dorotea and Fernando	8
Battle with the Knight of the Mirrors	8
Princesses and castles	8
Flight on magical horse Clavileño	9
Don Q.'s encounter with the king's lions	9
Dispute over the helmet of Mambrino	10
Altisidora	11
Don Q.'s letter to Dulcinea	11
Don Q.'s descent into the cave of Montesinos	11
Sancho's loved ones	11
Minor skirmishes	12
Don Q.'s knighting ceremony	12
Roque the bandit	12
Ceremonies at the Duke's castle	12
Sansón Carrasco	13
The canon and goatkeepers	13
Doña Rodríguez and the ghost	13
Don Q.'s loved ones	14

**Table 5. Derived Topics**

As the Number of Chapters column indicates, the topics vary widely in their span across chapters of the book, from as few as 3 chapters to as many as 14. In other words, some topics are rather transient whereas others tend to have a longer lifespan or recur throughout the text.

Figure 7 shows smoothed topic weights that the Text Topic node produces, where each panel represents a derived theme. The time dimension is chapter order, and the Y-axis values are relevance weights. Thus, one or more peaks of each series indicate when the topic is most central to the story. Each theme was color-coded according to its temporal pattern: brief episode, recurring episode, declining theme, or extended episode. Although the grouping is subjective, it suggests that the key themes of the book are mostly single episodes (encounters, confrontations, and battles), followed by recurring themes, extended episodes, and declining themes.

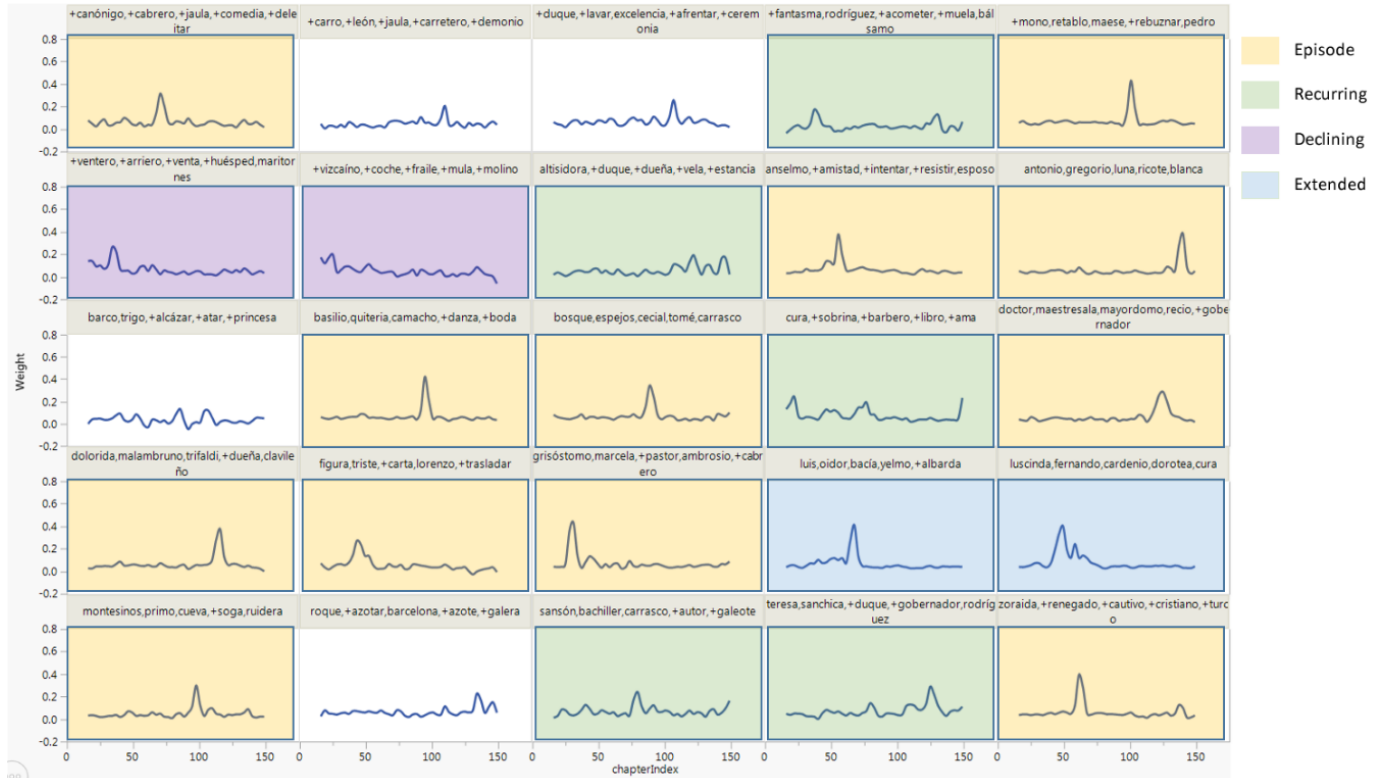


Figure 7. Spline-Smoothed Relevance Weights by Theme

Table 6 shows examples of each type of theme.

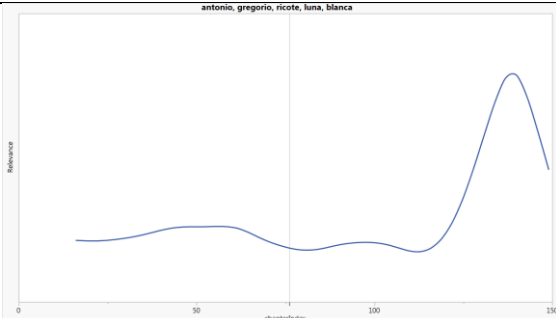
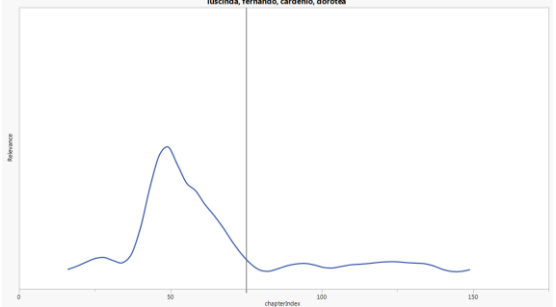
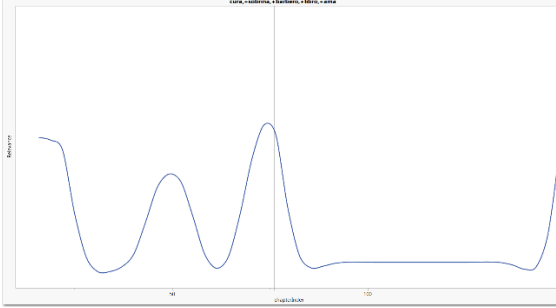
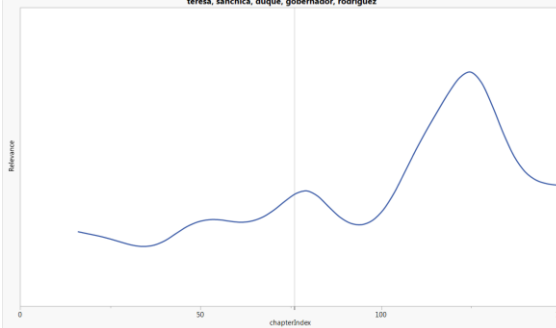
Type	Trajectory	Theme	Peaks
Episode		Don Q. battles with the Knight of the White Moon.	End of book 2
Extended Episode		Separated lovers are reunited: Luscinda and Cardenio and Dorotea and Fernando (a story within the story).	Middle of book 1
Recurring Theme		Don Q.'s support system (Niece, Housekeeper, Priest, Barber) intervene to discourage Don Q.'s irrational adventures, fetch him from the field, and assist in his convalescence.	Beginning and end of both books and middle of book 1
Building Theme		Governor Sancho: Don Q. promises Sancho Panza that he will be richly rewarded for helping to fight monsters and save fair damsels. Not seeing the rewards materialize, Sancho grows impatient, but ultimately he is named governor of a small territory.	Middle of book 2

Table 6. Theme Trajectory Examples



Finally, some important and well-spaced series were selected and overlaid to create a thematic timeline, a visual thematic summary of the two books.

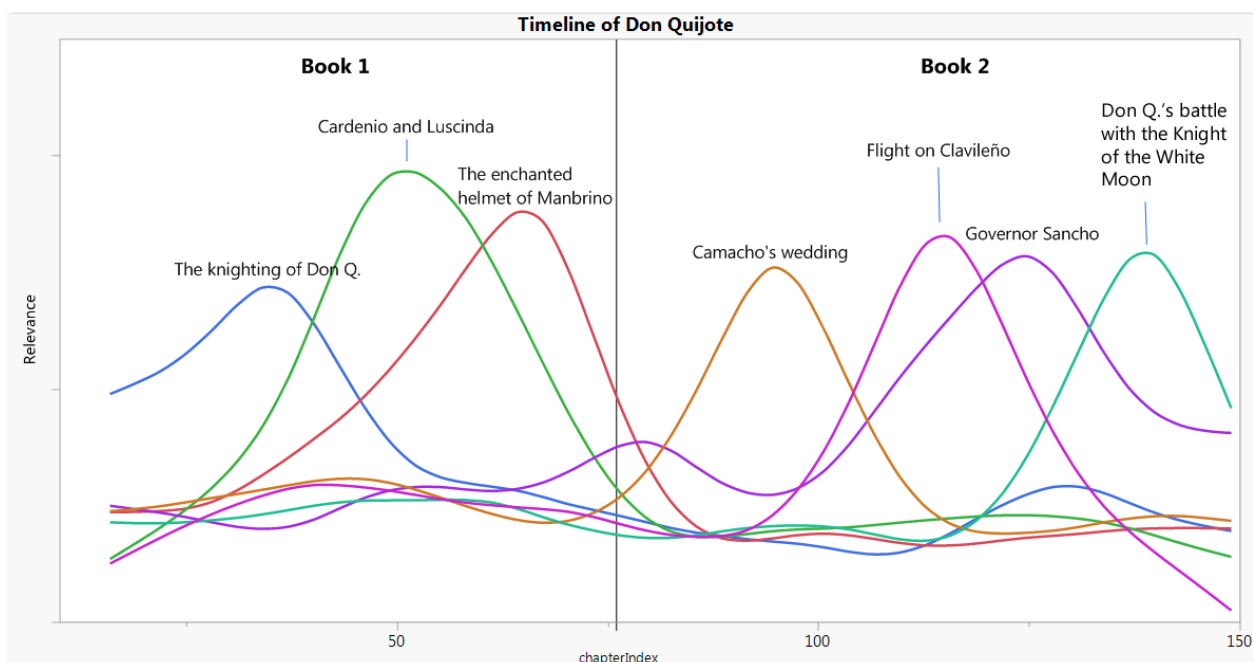


Figure 8. Thematic Timeline

## AUTOMATION

The preceding analysis illustrates how to identify interesting temporal patterns via manual exploration of derived themes. But what if you wanted to summarize an entire library of books or monitor customer reviews for thousands of products? If you have a very large document collection, manual grouping of themes would become time-consuming or even infeasible.

Fortunately, you can use time series methods to automatically group themes that have similar temporal trajectories and select well-spaced themes for a thematic timeline. To apply these methods to the text, you can simply extend the basic Enterprise Miner flow, as shown in Figure 9.

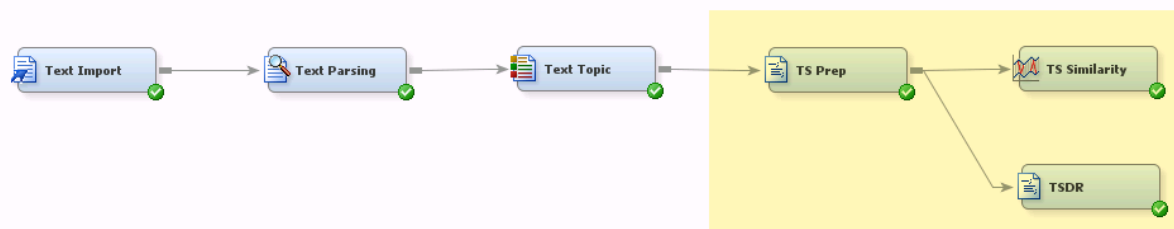


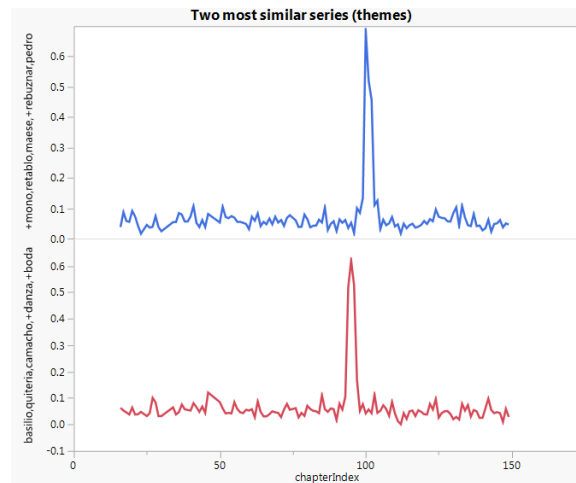
Figure 9. Extended Text Mining Flow

Here the TS Prep node is a custom SAS code node that transforms the topics data set that is exported by the Text Topic node. The code creates an integer time ID variable (**chapterIndex**) and sorts the topics by that time ID variable. The TS Similarity and TSDR nodes are described in subsequent sections.

## TIME SERIES CLUSTERING

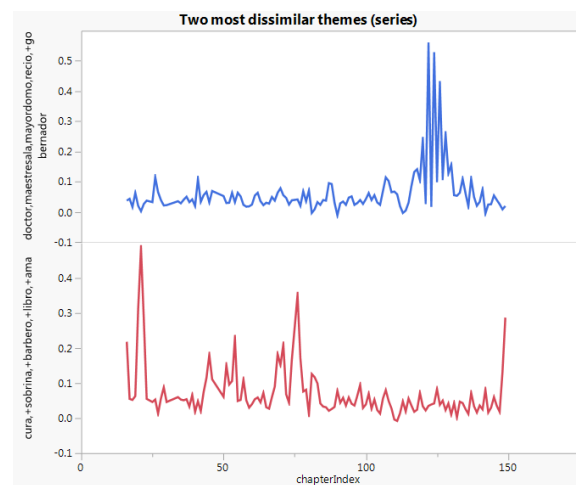
The TS Similarity node uses hierarchical clustering to group the series. Hierarchical clustering groups pairs of series iteratively: each iteration first groups the two most similar series into a cluster and then groups the next most similar pair (where one of the elements of the pair could be an existing cluster). This process continues until all series are joined.

By default, the TS Similarity node uses the sum of squared deviations between series values as the similarity criterion. For example, Figure 10 shows two pairs of topic series that are identified using the distance map that the node produces.



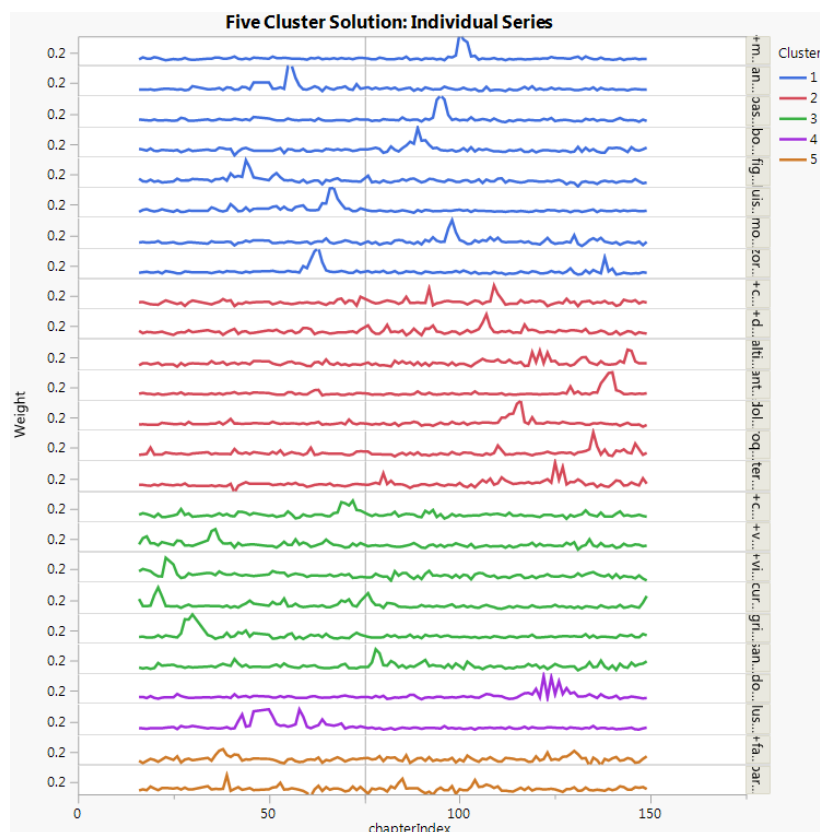
**Figure 10. Two Most Similar Series**

The first pair of series are the two topics that have the smallest pairwise distance (0.072) among all pairs of series. The series plots confirm that they are quite similar, both being essentially flat except for a pronounced spike around the same chapter. The two most dissimilar series (shown in Figure 11) have the highest pairwise distance (0.669), and the series plots confirm that they have very different temporal patterns.



**Figure 11. Two Most Dissimilar Series**

Various solutions were explored and a five-cluster solution, which is shown in Figure 12, provided the most intuitive groupings. Each series is color-coded by cluster number, and the clusters are defined in Table 7.



**Figure 12. Five-Cluster Solution**

Cluster	Description
1	Episodes occurring in the core of first and second books
2	Later episodes (book 2)
3	Early episodes (book 1)
4	Recurring or cycling episodes
5	Flat or noisy series

**Table 7. Cluster Descriptions**

These derived groups differ mostly by when spikes occur (if at all). For a business example of time series clustering, imagine that you have customer reviews collected over time for a large number of products. Clustering could be used to identify groups of products (that is, product families) where customer sentiment is trending favorably or unfavorably. Alternatively, if you score each product on themes that relate to particular product features or use cases, you could identify features and use cases of increasing or decreasing importance to your customers.

## TIME SERIES DIMENSION REDUCTION

As an alternative to manually selecting topics to include in a thematic timeline, you can use time series dimension reduction (TSDR) techniques to select well-spaced series. One example of TSDR is the piecewise aggregation approximation (PAA) method. PAA bins series along the time dimension and identifies the most representative series (that is, the series that has the highest mean value) in each bin.

PAA was performed using a DATA step and SAS/IML code in the TSDR node (a custom code node). The following macro call selects seven representative topic series, the same number of series as in the manually generated timeline shown in Figure 8.

```
%macro PAA(nBins=5);

    *get number of rows (chapters);
    data _null_;
        set &em_import_data end=eof;
        if eof then call symput('nChapters',_n_);
    run;

    %let nbinsPerChapter = %eval(&nChapters/&nBins.);

    *bin the rows (chapters);
    data binRows;
        set &em_import_data;
        if (_n_ <= &nbinsPerChapter.) then bin = 1;
        %do i = 2 %to &nbins.;
            else if (_n_ <= &nbinsPerChapter. * &i.) then bin = &i.;
        %end;
    run;

    *mean topic weight by bin;
    proc summary data = binRows;
        class bin;
        output out = binnedTopicWeights (keep= TextTopic2_raw:);
        mean(TextTopic2_raw1-TextTopic2_raw25) =
            TextTopic2_raw1 -TextTopic2_raw25;
    run;

    *create a list of the topics that have the highest mean in each bin;
    proc iml;
        use binnedTopicWeights;
        read all var _num_ into temp[c=varNames];
        close binnedTopicWeights;

        *drop _type_=0 row;
        X=temp[1:&nBins.,];

        idxMax = X[, <:>];
        varMax = varNames[idxMax];
        print varMax;
    quit;

%mend PAA;

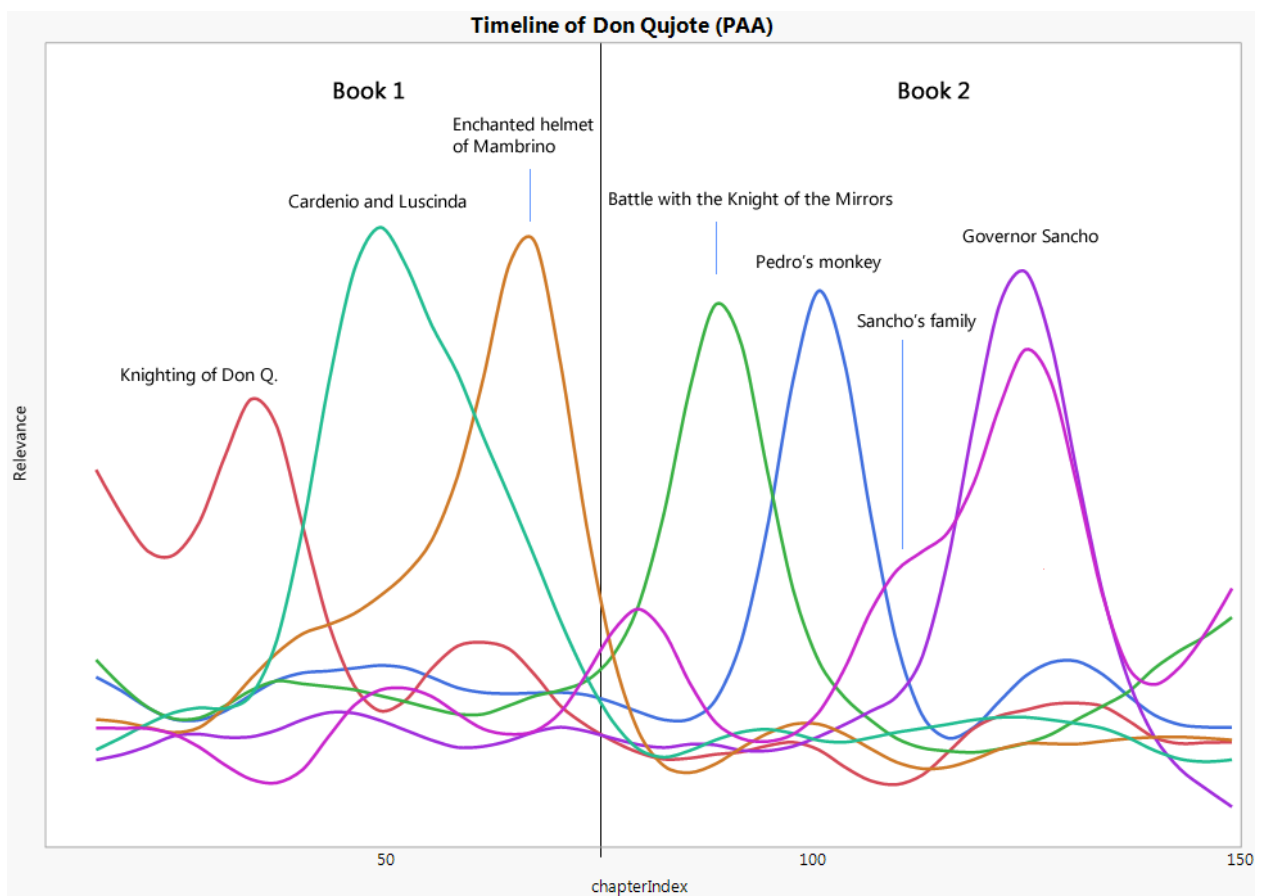
%PAA(nBins=7);
```

The preceding code generates the list of selected topics shown in Figure 13.

varMax
TextTopic2_raw18
TextTopic2_raw11
TextTopic2_raw19
TextTopic2_raw5
TextTopic2_raw21
TextTopic2_raw8
TextTopic2_raw2

**Figure 13. Selected Topics**

Figure 14 shows the revised timeline of the seven series that are selected using PAA.



**Figure 14. Revised Timeline**

PAA worked reasonably well: the selected series cover the full timeline, and most of the peaks are separated well (the exception being the last two peaks, which overlap).

PAA can also be useful in business contexts. You could use PAA to bin customer sentiment into 12 (monthly) or 4 (quarterly) bins and identify the products that have the highest or lowest acceptance in each time interval. This binning might help uncover seasonal patterns in customers' attitudes.

## ALTERNATIVE THEME EXTRACTIONS

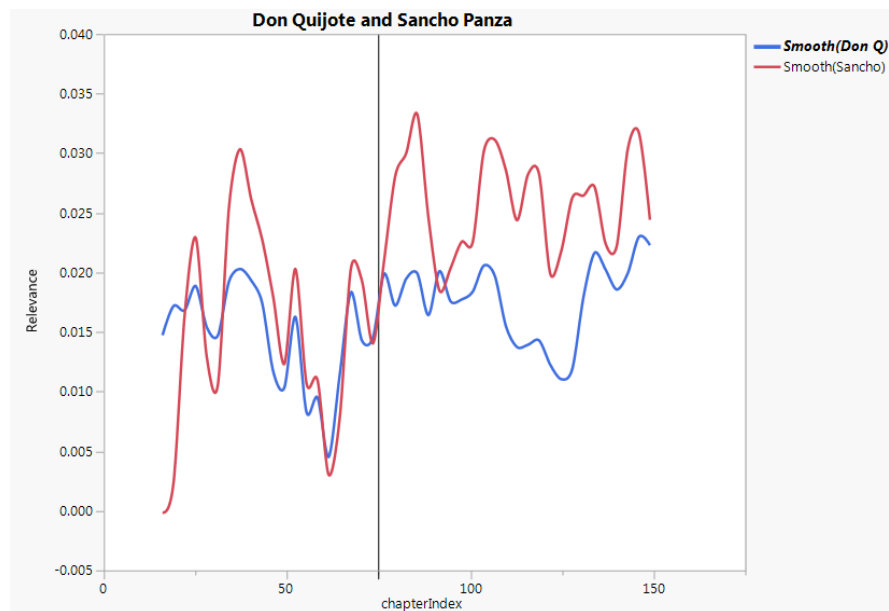
This section discusses some alternative theme extractions that provide a different perspective compared to the themes derived in the section "Derived Themes."

### CUSTOM TOPICS

The tense bond between Don Q. and Sancho is considered a key theme of *Don Quixote*, but it is not evident among the 25 derived themes. That is not too surprising considering that the topics were derived based on whether terms co-occurred, not on the scholarly significance or reader impact of the terms. Fortunately, Text Miner enables you to explore custom themes. You simply specify the terms that define each custom topic in the Text Topic node and use the node to score each document.

To explore the relationship between the two protagonists, two custom topics were created ("Don Quixote" and "Sancho Panza") by using the terms "Quixote" and "Sancho," respectively. These terms were chosen carefully. Although "quixote" has another meaning in Spanish, that usage is not common. Thus, you would expect very few, if any, instances of "quixote" in the book that do not refer to the title character. Similarly, there should be few, if any, instances in which "Sancho" refers to anyone other than Sancho Panza. Although the definition of a custom topic can include multiple terms, neither "don" nor "panza" were included as separate terms in the custom topic definition because both are common terms with Spanish meanings that have nothing to do with the main characters.

Figure 15 shows the time profiles for the custom topics.



**Figure 15. Custom Themes**

Interestingly, the graph shows that the two main characters are not always the main focus of the book: both Don Q. and Sancho take on a secondary role near the end of first book (chapters 50–70 or so).

These chapters correspond to the major “story within a story” of the reunited lovers (Luscinda and Cardenio and Dorotea and Fernando).

In addition, the two characters are definitely not joined at the hip. In particular, their respective importance to the story diverges in the second book. This divergence coincides with Sancho’s governorship, where Sancho becomes the central character and Don Q. fades into the background.

The support for custom topics provided in Text Miner gives you great flexibility in all kinds of applications, because you don’t need to be limited by the set of topics that are derived via SVD. Rather, you can score documents on any topic you might be interested in.

## MICROANALYSIS

In the preceding analysis, chapters are treated as the unit of analysis. This treatment yields themes about episodes and arcs within the larger story. Breaking up the text into smaller units such as paragraphs or even sentences can provide an alternative, more granular representation of the text that is driven by local information (Albright, Cox, and Jin 2016).

To identify potentially interesting themes that are not captured by the chapter-by-chapter analysis, the text of the two volumes was split into paragraphs (one file per paragraph) and the Text Topic node was used to derive 25 new themes. This analysis reveals some interesting themes—some of which seem like fundamental themes of the text—for example, knighthood and knights errant who travel the world (“caballero, andante, mundo, caballeria”) and reading and telling of stories (“historia, leer, contar, libro, mucho”).

## CONCLUSION

This paper illustrates the use of temporal text mining techniques to explore a semantically rich two-volume text. Using a combination of text mining and time series methods makes it possible to extract key themes, group themes that have similar temporal profiles, and create a visual thematic timeline. This paper also shows that significant automation is possible, which could prove critical when the task is to summarize large document collections such as a library of classic books.

The methods used in this paper are not just for analysis of Great Books; indeed, they easily lend themselves to business applications. If your task is to monitor customer opinions about a product, you could use text topic extraction to identify the most important themes among the reviews—such as themes that involve particular features of the product or customer sentiments. Furthermore, if the reviews are collected at regular intervals (or are grouped into regular intervals after the fact), then it should be possible to use time series forecasting techniques to characterize the time course of each theme, identify emerging themes (such as new customer needs, preferences, or use cases), and even predict future customer sentiment. A similar approach could be used in domains such as academia (where the documents consist of articles in a particular field of research) or legislation (where the documents consist of legislative actions or proposals).

## REFERENCES

Albright, R. 2004. “Taming Text with the SVD.” Cary, NC: SAS Institute Inc.  
<ftp://ftp.sas.com/techsup/download/EMiner/TamingTextwiththeSVD.pdf>.

Albright, R., Cox, J., and Jin, N. 2016. “Getting More from the Singular Value Decomposition (SVD): Enhance your Models with Document, Sentence, and Term Representations.” *Proceedings of the SAS Global Forum 2017 Conference*. Cary, NC: SAS Institute Inc. Available  
<https://support.sas.com/resources/papers/proceedings16/SAS6241-2016.pdf>.

Cervantes de Saavedra, M. *Don Quijote de la Mancha*. Available  
<http://cervantes.uah.es/quijote/htoc.htm>. Accessed December, 2016.

SAX-VSM. “Piecewise Aggregate Approximation of Time Series.” [https://jmotif.github.io/sax-vsm\\_site/morea/algorithm/PAA.html](https://jmotif.github.io/sax-vsm_site/morea/algorithm/PAA.html). Accessed November, 2016.

## ACKNOWLEDGMENTS

The author thanks Russ Albright and Ilknur Kaynar Kabul for their valuable feedback and suggestions.

## CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Ray Wright  
SAS  
Ray.wright@sas.com

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.