WHAT DID THEY MEAN BY THAT? THE ESSENTIALS OF TEXT EXPLORER IN JMP®

DON MCCORMACK AND BILL WORLEY

JMP TECHNICAL ENABLEMENT

PREPARING DATA FOR TEXT EXPLORER – INTRO

- Don – Preparing data for analysis
  - What to do before you put your data in JMP
  - Preparing your data with Text Explorer
- Bill – Text Analytics
  - Converting text to “numbers” or scores – several options
  - Saving a Document Term Matrix and/or Topic Scores
  - Improving predictive models with “words”
  - Warranty Data – Life Distribution
  - New in JMP 14 Discriminant Analysis

WHAT DID THEY MEAN BY THAT? THE ESSENTIALS OF TEXT EXPLORER IN JMP®

PREPARING YOUR DATA FOR TEXT EXPLORER

D. MCCORMACK – JMP TECHNICAL ENABLEMENT

PREPARING DATA FOR TEXT EXPLORER – INTRO

- Terms used for text exploration
  - term/token – the smallest independent piece of text. A string of consecutive alphanumeric characters typically containing no punctuation. Tokenizing is the process of breaking a document into tokens.
  - phrase – a collection of two or more terms.
  - document – a collection of phrases. Stored in JMP table cells
  - corpus – a collection of documents. Store in a JMP column.
  - stop word – common terms that add noise to the analysis. Articles and quantifiers are two examples. Somewhat language dependent.
  - stem/stemming – the process of grouping words that share a common base form (stem) but have different endings. Very language dependent.
- JMP uses a bag-of-words approach to text mining. Only terms and phrases that are shared by documents are considered. Outside of phrases, word order is ignored.

PREPARING DATA FOR TEXT EXPLORER – INTRO

1. Data access and pre Text Explorer cleaning
   - The data source and what you need to extract from it will determine how much work is needed before launching Text Explorer. Scanned hardcopy text, such as log files, will likely require more preprocessing than structured electronic data stored in a database.
2. Pre-analysis tuning
   - Default tokenization may be improved by recoding misspellings, abbreviations, hyphenations, or contractions, and adding domain specific stop words and multiple token phrases.
   - Custom regular expressions may further enhance tokenization for phrases that follow a systematic pattern.
   - Other parameters (words/phrase, chars/word, stemming, etc.) may also be used to optimize tokenization.

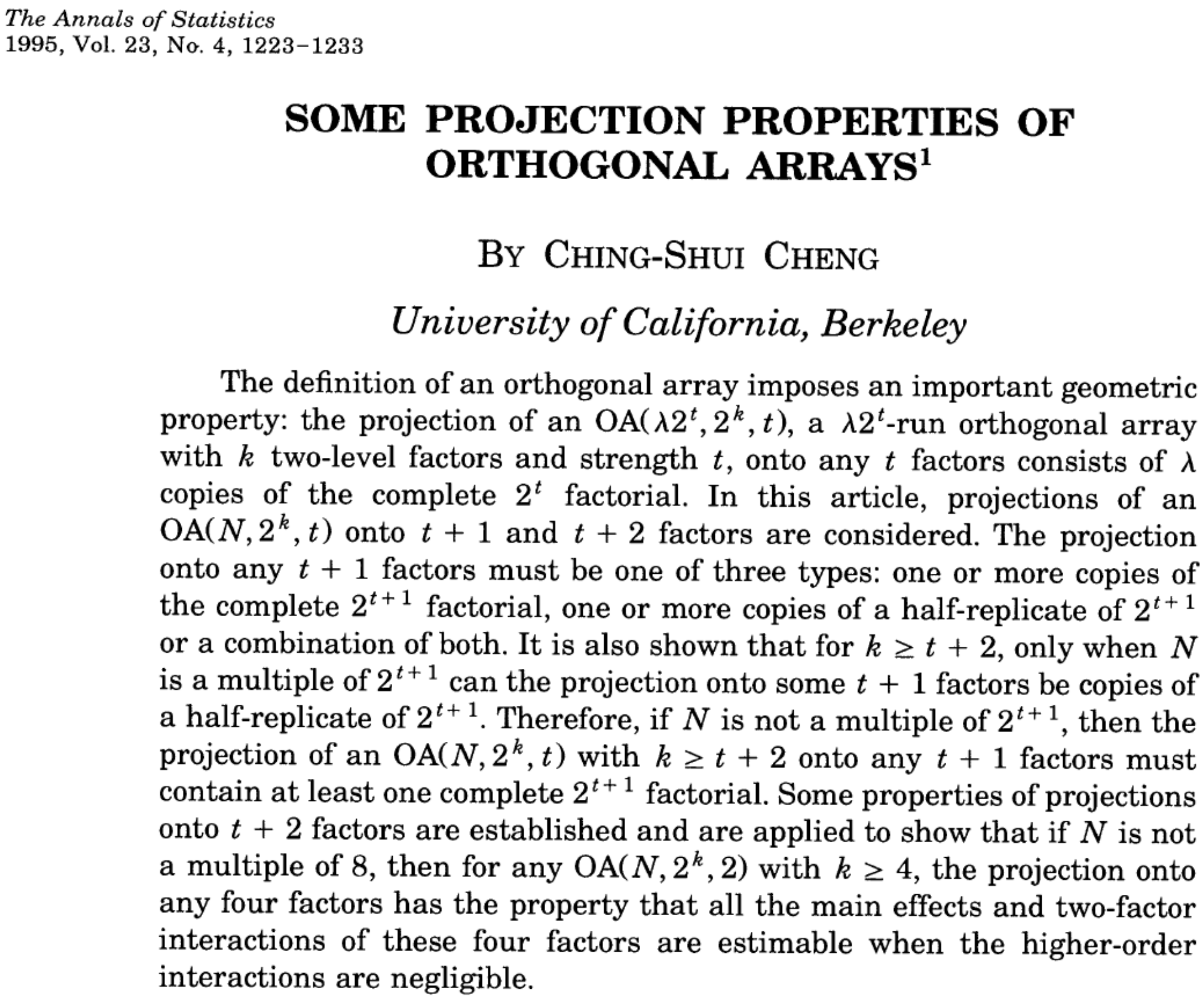
PREPARING DATA FOR TEXT EXPLORER – DATA ACCESS

- Tabular data: documents stored in a database, Web data retrieval programs, Excel files, HTML table, etc.
  - Easiest way to deal with data. Open in JMP and go to Text Explorer
- Web: scrape, crawl, and APIs
  - One table/multiple pages (crawl only) or single page, non tabular (scrape only) could be implemented with minimal coding.
  - Pre clean with Python (or Perl, or Ruby, or PHP, or …) or JSL?
  - Resources: Beautiful Soup, Scrapy
- PDF/Hardcopy: physical copy that may already be stored in PDF
  - OCR is a must for image based PDFs
  - There will be errors
  - Post processing work depends on quality of conversion, document regularity, and what you're trying to extract.
  - Resources: pdfminer, Other PDF conversion with OCR

PREPARING DATA FOR TEXT EXPLORER – DATA ACCESS

- Examples
  - Collecting journal abstracts using a combination of typing, reading (with aide of voice recognition software) and PDF converter with OCR.
    - Papers did not follow the same format
    - Mathematical symbols not recognized by OCR
    - The abstract was difficult to isolate from the rest of the article
  - Scraping the JMP Blogs at community.jmp.com
  - Using built in tools to gather all posts from community.jmp.com
    - Some post processing needed
  - Accessing German versions of Kafka works at the Gutenberg Project
    - PDF access did not separate the paragraphs
    - Web access required scraping code and crafty pre Text Explorer cleaning to remove HTML tags.
  - Traffic violations downloaded from the Montgomery County website

PREPARING DATA FOR TEXT EXPLORER – DATA ACCESS



The Annals of Statistics
1995, Vol. 23, No. 4, 1223–1233

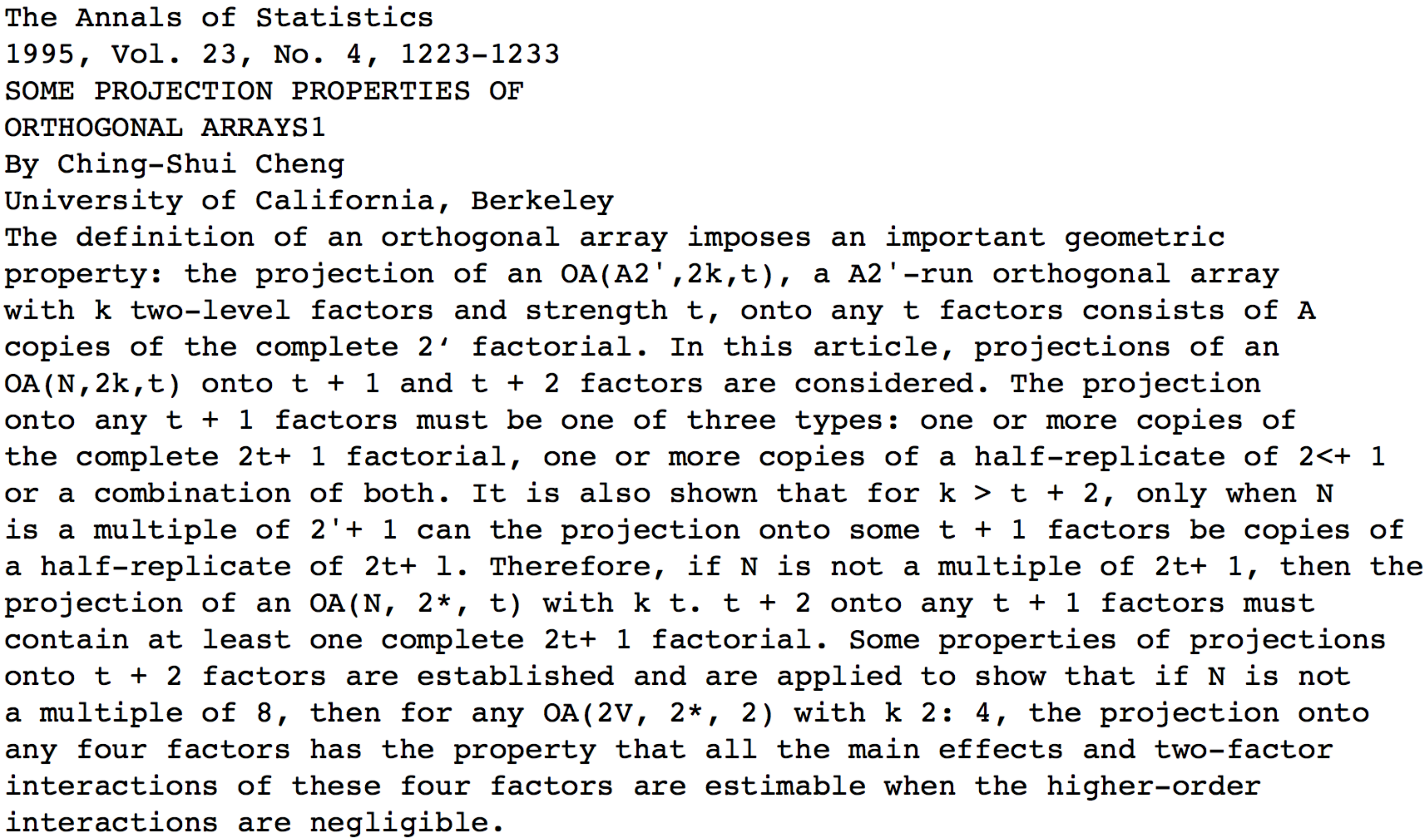
SOME PROJECTION PROPERTIES OF ORTHOGONAL ARRAYS[1]

BY CHING-SHUI CHENG

University of California, Berkeley

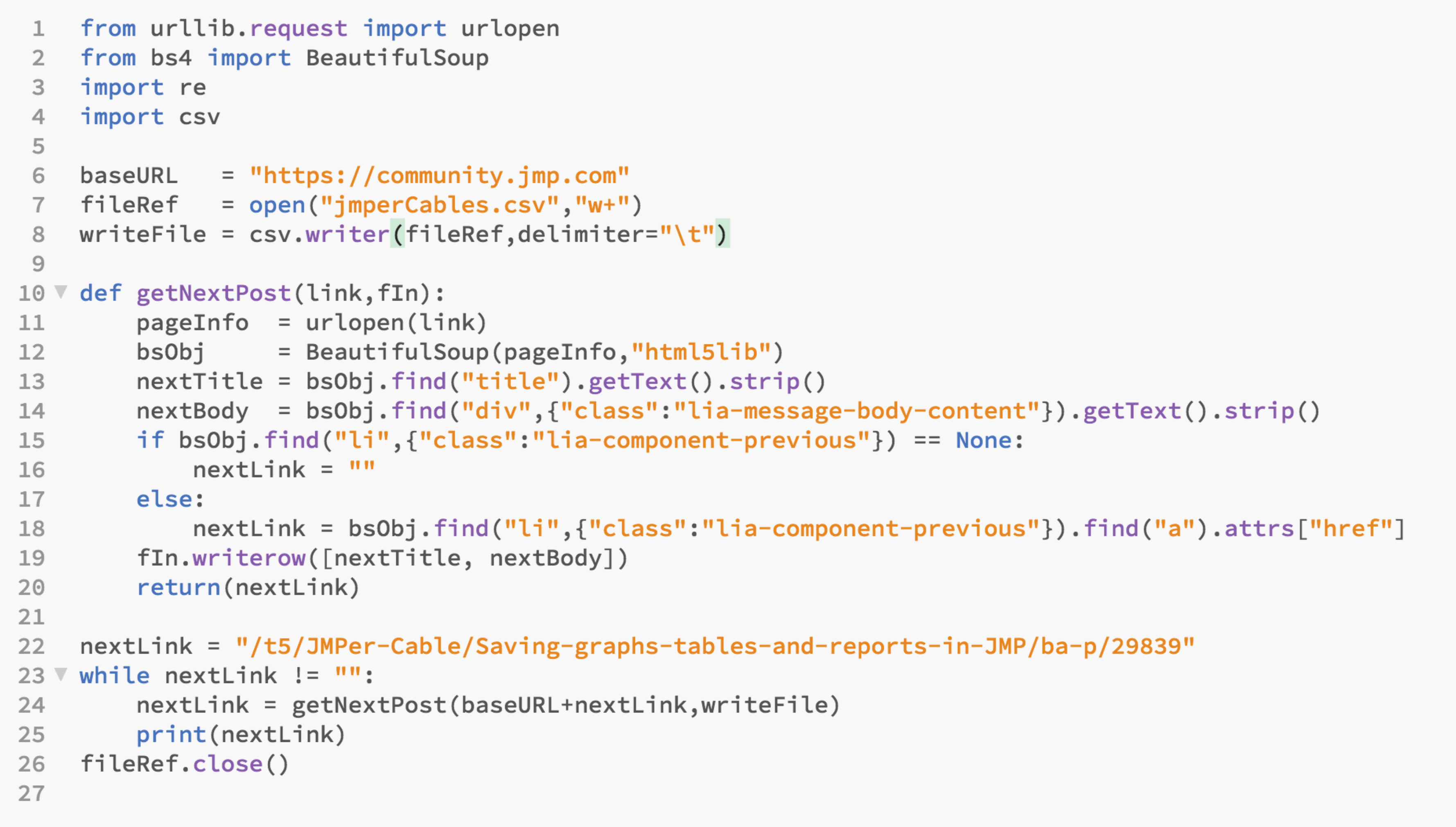
The definition of an orthogonal array imposes an important geometric property: the projection of an OA($\lambda 2^t, 2^k, t$), a $\lambda 2^t$-run orthogonal array with k two-level factors and strength t, onto any t factors consists of λ copies of the complete 2^t factorial. In this article, projections of an OA($N, 2^k, t$) onto $t + 1$ and $t + 2$ factors are considered. The projection onto any $t + 1$ factors must be one of three types: one or more copies of the complete 2^{t+1} factorial, one or more copies of a half-replicate of 2^{t+1} or a combination of both. It is also shown that for $k \geq t + 2$, only when N is a multiple of 2^{t+1} can the projection onto some $t + 1$ factors be copies of a half-replicate of 2^{t+1}. Therefore, if N is not a multiple of 2^{t+1}, then the projection of an OA($N, 2^k, t$) with $k \geq t + 2$ onto any $t + 1$ factors must contain at least one complete 2^{t+1} factorial. Some properties of projections onto $t + 2$ factors are established and are applied to show that if N is not a multiple of 8, then for any OA($N, 2^k, 2$) with $k \geq 4$, the projection onto any four factors has the property that all the main effects and two-factor interactions of these four factors are estimable when the higher-order interactions are negligible.

PREPARING DATA FOR TEXT EXPLORER – DATA ACCESS



```
The Annals of Statistics
1995, Vol. 23, No. 4, 1223-1233
SOME PROJECTION PROPERTIES OF
ORTHOGONAL ARRAYS1
By Ching-Shui Cheng
University of California, Berkeley
The definition of an orthogonal array imposes an important geometric
property: the projection of an OA(A2',2k,t), a A2'-run orthogonal array
with k two-level factors and strength t, onto any t factors consists of A
copies of the complete 2‘ factorial. In this article, projections of an
OA(N,2k,t) onto t + 1 and t + 2 factors are considered. The projection
onto any t + 1 factors must be one of three types: one or more copies of
the complete 2t+ 1 factorial, one or more copies of a half-replicate of 2<+ 1
or a combination of both. It is also shown that for k > t + 2, only when N
is a multiple of 2'+ 1 can the projection onto some t + 1 factors be copies of
a half-replicate of 2t+ 1. Therefore, if N is not a multiple of 2t+ 1, then the
projection of an OA(N, 2*, t) with k t. t + 2 onto any t + 1 factors must
contain at least one complete 2t+ 1 factorial. Some properties of projections
onto t + 2 factors are established and are applied to show that if N is not
a multiple of 8, then for any OA(2V, 2*, 2) with k 2: 4, the projection onto
any four factors has the property that all the main effects and two-factor
interactions of these four factors are estimable when the higher-order
interactions are negligible.
```

PREPARING DATA FOR TEXT EXPLORER – DATA ACCESS



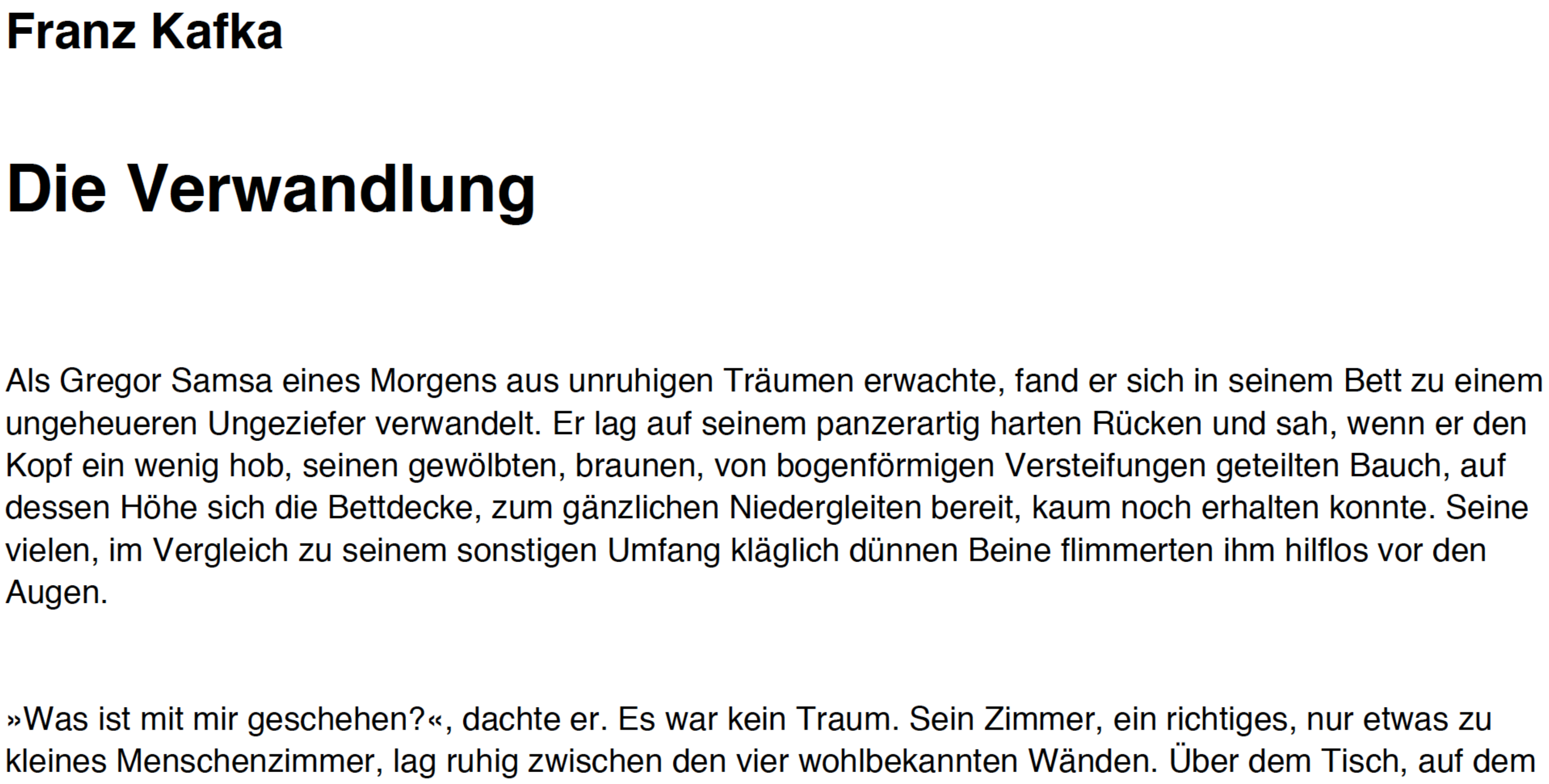
```python
from urllib.request import urlopen
from bs4 import BeautifulSoup
import re
import csv

baseURL   = "https://community.jmp.com"
fileRef   = open("jmperCables.csv","w+")
writeFile = csv.writer(fileRef,delimiter="\t")

def getNextPost(link,fIn):
    pageInfo  = urlopen(link)
    bsObj     = BeautifulSoup(pageInfo,"html5lib")
    nextTitle = bsObj.find("title").getText().strip()
    nextBody  = bsObj.find("div",{"class":"lia-message-body-content"}).getText().strip()
    if bsObj.find("li",{"class":"lia-component-previous"}) == None:
        nextLink = ""
    else:
        nextLink = bsObj.find("li",{"class":"lia-component-previous"}).find("a").attrs["href"]
    fIn.writerow([nextTitle, nextBody])
    return(nextLink)

nextLink = "/t5/JMPer-Cable/Saving-graphs-tables-and-reports-in-JMP/ba-p/29839"
while nextLink != "":
    nextLink = getNextPost(baseURL+nextLink,writeFile)
    print(nextLink)
fileRef.close()
```

PREPARING DATA FOR TEXT EXPLORER – DATA ACCESS



Franz Kafka

Die Verwandlung

Als Gregor Samsa eines Morgens aus unruhigen Träumen erwachte, fand er sich in seinem Bett zu einem ungeheueren Ungeziefer verwandelt. Er lag auf seinem panzerartig harten Rücken und sah, wenn er den Kopf ein wenig hob, seinen gewölbten, braunen, von bogenförmigen Versteifungen geteilten Bauch, auf dessen Höhe sich die Bettdecke, zum gänzlichen Niedergleiten bereit, kaum noch erhalten konnte. Seine vielen, im Vergleich zu seinem sonstigen Umfang kläglich dünnen Beine flimmerten ihm hilflos vor den Augen.

»Was ist mit mir geschehen?«, dachte er. Es war kein Traum. Sein Zimmer, ein richtiges, nur etwas zu kleines Menschenzimmer, lag ruhig zwischen den vier wohlbekannten Wänden. Über dem Tisch, auf dem

PREPARING DATA FOR TEXT EXPLORER – TUNING

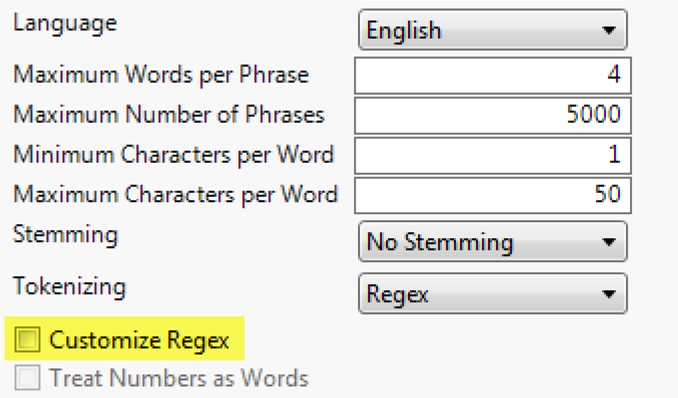
- Text Explorer prepares the text in this order:
  - Text strings are converted to lowercase.
  - Tokenization – Regex or Basic Words. By default, most punctuation is removed in this step. Custom regex can be used to include terms with punctuation.
  - Terms are recoded. While there are no built-in recodes they can be added by the user.
  - Phrases are collected. They cannot start or end with a stop word. Custom regex can be used to include phrases starting/ending with stop words.
  - Terms that are not a stop word and meet the criteria for maximum and minimum characters are collected.
  - Stemming and stem exceptions are applied.

PREPARING DATA FOR TEXT EXPLORER – REGEX

- What is it? A tool to capture terms that follow a systematic pattern.
- Why bother? Because the default tokenization misses terms that don't follow the rules (e.g., start with a stop word, have punctuation in them, contain more than one word, etc.).
- JMP provides a default list of regular expressions. If you add to that list, it is saved locally but exists in a file that can be shared.
  - Custom regular expressions must be reloaded when Text Explorer is relaunched unless it is run from a saved script.

PREPARING DATA FOR TEXT EXPLORER – REGEX EDITOR

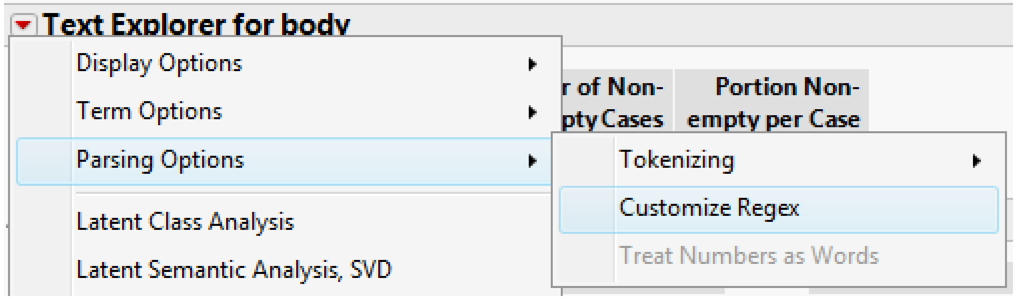
- Access the Editor by checking the Customize Regex box in the first dialog or by selecting Parsing Options > Customize Regex under the red hotspot menu. The first method lets you load customized regular expressions you have previous created before tokenizing the documents.



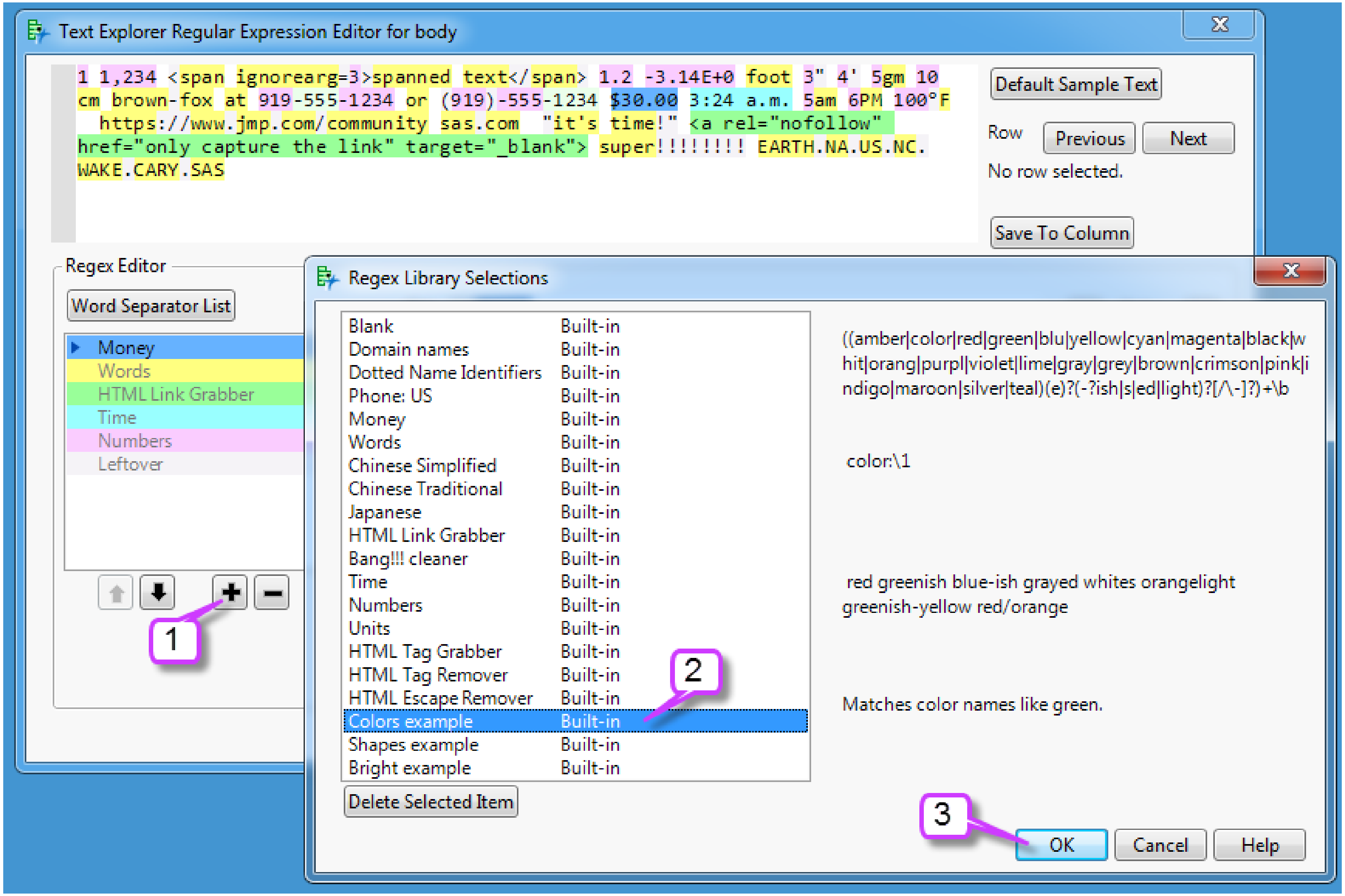
Language	English
Maximum Words per Phrase	4
Maximum Number of Phrases	5000
Minimum Characters per Word	1
Maximum Characters per Word	50
Stemming	No Stemming
Tokenizing	Regex

□ Customize Regex

□ Treat Numbers as Words



PREPARING DATA FOR TEXT EXPLORER – REGEX EDITOR



```
1 1,234 <span ignorearg=3>spanned text</span> 1.2 -3.14E+0 foot 3" 4' 5gm 10
cm brown-fox at 919-555-1234 or (919)-555-1234 $30.00 3:24 a.m. 5am 6PM 100°F
 https://www.jmp.com/community sas.com  "it's time!" <a rel="nofollow"
href="only capture the link" target="_blank"> super!!!!!!!! EARTH.NA.US.NC.
WAKE.CARY.SAS
```

No row selected.

- Money
- Words
- HTML Link Grabber
- Time
- Numbers
- Leftover

Blank	Built-in
Domain names	Built-in
Dotted Name Identifiers	Built-in
Phone: US	Built-in
Money	Built-in
Words	Built-in
Chinese Simplified	Built-in
Chinese Traditional	Built-in
Japanese	Built-in
HTML Link Grabber	Built-in
Bang!!! cleaner	Built-in
Time	Built-in
Numbers	Built-in
Units	Built-in
HTML Tag Grabber	Built-in
HTML Tag Remover	Built-in
HTML Escape Remover	Built-in
Colors example	Built-in
Shapes example	Built-in
Bright example	Built-in

((amber|color|red|green|blu|yellow|cyan|magenta|black|whit|orang|purpl|violet|lime|gray|grey|brown|crimson|pink|indigo|maroon|silver|teal)(e)?(-?ish|s|ed|light)?[/\-]?)+\b

color:\1

red greenish blue-ish grayed whites orangelight greenish-yellow red/orange

Matches color names like green.

PREPARING DATA FOR TEXT EXPLORER – REGEX

- Special characters – `() [] {} ^$*+.?|-\`
  - Have special functions, must be preceded by `\` to specify them literally.
  - Because `\[` has a special JSL meaning, use `\x5B` instead
  - `]` and `-` only need to be escaped after and unclosed `\[`
  - `}` only needs to be escaped after an unclosed `{`
- Character shorthands and classes
  - `\d` – Digits: 0 – 9
  - `\w` – Word characters: a – z, A – Z, 0 – 9, and _ (underscore)
  - `\s` – Whitespace (space, tab, line break, and form feed)
  - `\D`, `\W`, and `\S` are the match the negation of `\d`, `\w`, and `\s`
  - `[…]` – Match any (single) character in the square brackets
  - `[^…]` – Match any (single) character not in the square brackets
  - (*a*|*b*) – Match the string *a* or the string *b*
  - [*n*–*m*] – Any character/number between *n* and *m*, where n and m are both numbers or letters and are in the proper order.

PREPARING DATA FOR TEXT EXPLORER – REGEX

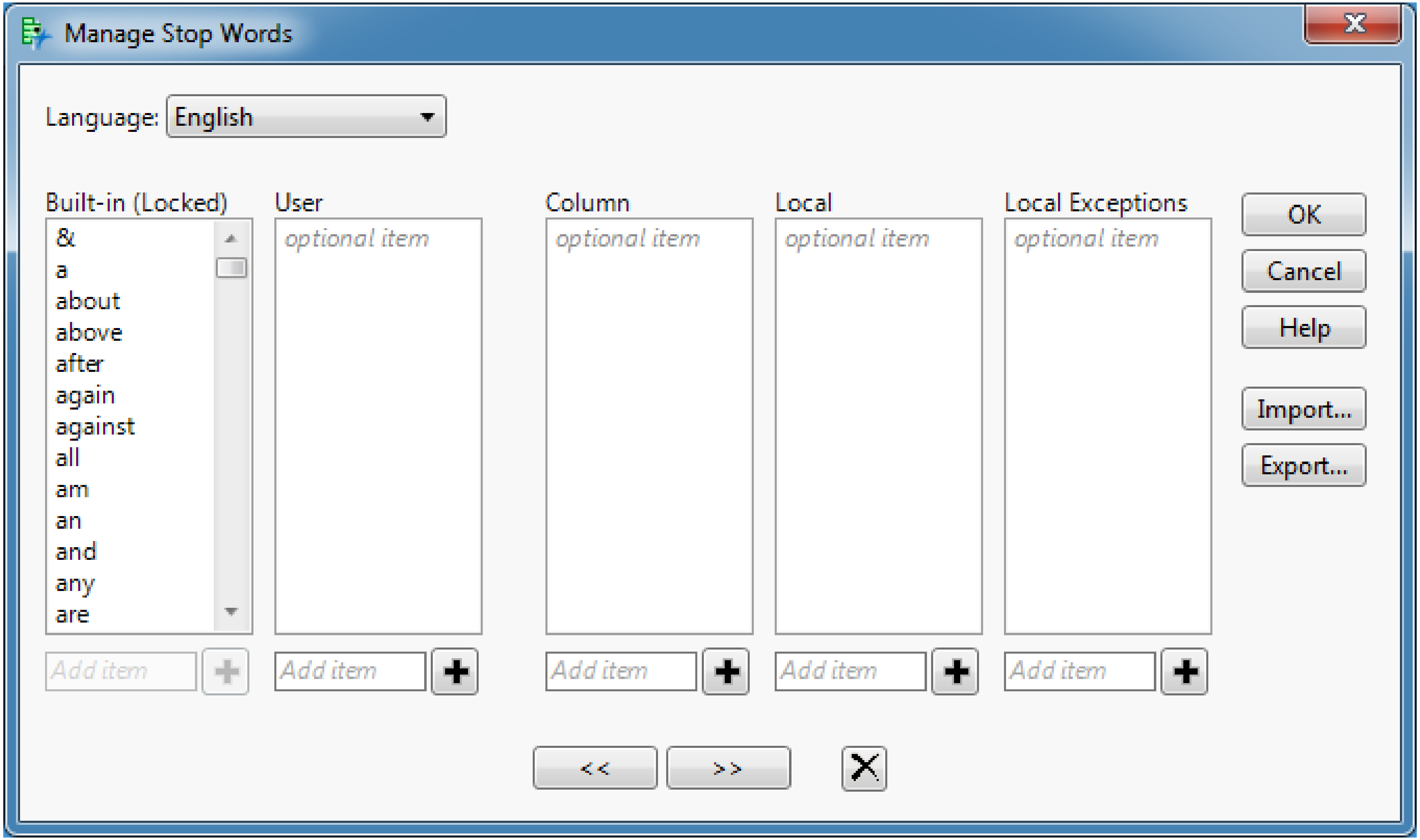
- Anchors
  - `^` – Beginning of a line (`\A` does the same)
  - `$` – End of a line (`\Z` does the same)
  - `\b` – Word boundary (ended with another `\b`)
- Quantifiers let you specify how often you match the character or expression immediately preceding them.
  - `*` – Zero or more times
  - `+` – One or more
  - `?` – Zero or one
  - {*i*} – Exactly *i* times.
  - {*i*, } – at least *i* times
  - {*i*,*j*} – Between *i* and *j* times.

PREPARING DATA FOR TEXT EXPLORER – RECODE, PHRASES, STOP WORDS, AND STEM EXCEPTIONS

- To manage recodes, phrases, stop words, and stem exceptions, select the appropriate item under the hotspot menu choice Term Options. The dialog box for each is similar and is shown on the next slide. Only stop words have built-in entries. Adding and removing items. Exporting
  - User – supplied by user.
  - Column – provided via column property.
  - Local – defined when Text Explorer is launched in a JSL script.
  - Local Exceptions – override built-in entries.
  - Project – defined when Text Explorer is launched from a JMP project.

    Items can be added by typing them in, copying from one list or another program and pasting, or importing from a text file.
- View Options (hotspot menu) displays the above properties.

PREPARING DATA FOR TEXT EXPLORER – RECODE, PHRASES, STOP WORDS, AND STEM EXCEPTIONS



Manage Stop Words

Language: English

Built-in (Locked)

&
a
about
above
after
again
against
all
am
an
and
any
are

PREPARING DATA FOR TEXT EXPLORER – RECODE, PHRASES, STOP WORDS, AND STEM EXCEPTIONS

- Regex – terms containing only numbers between delimiters are removed. Keep the log window open because some errors only appear in the log.
- Basic words – terms containing only numbers are removed unless treat numbers as terms is checked (must contain alpha or Unicode)

- Term and phrase colors:

Type	Color
Built-in	Red
User Library	Green
Project	Blue
Column Property	Orange
Local	Grey

PREPARING DATA FOR TEXT EXPLORER – INTRO

- Features new in JMP® 14
  - You can filter the displayed Stop Words, Specified Phrases, Stem Exceptions, Term List, Phrase List, and the Stem Report lists (Display Options > Show Filters for all Tables).
  - You can start with a blank regular expression.
  - You can remove a phrase from the term list through right clicking.
  - When you paste text from a script to a document that supports Rich Text Form (RTF), such as Microsoft Word, the syntax colors are preserved in the Show Text window. The background colors from a Text Explorer script are also preserved.

PREPARING DATA FOR TEXT EXPLORER – RESOURCES

- Accessing Data
  - Beautiful Soup 4.6.0 Python routines for accessing web pages.
  - Mitchell, R. (2015), Web Scraping with Python: Collecting Data from the Modern Web, O'Reilly.
  - pdfminer Python routines for accessing PDF files
  - Scrappy 1.5 web crawling routines for Python
- Regular Expressions
  - https://www.regular-expressions.info/
  - Goyvaerts, J. & Levithan, J. (2012), Regular Expression Cookbook, 2nd Ed., O'Reilly.
  - Stubblebine, T. (2009), Regular Expression Pocket Reference, 2nd Ed., O'Reilly.