The Basics of Regular Expressions in Tableau

Inger Bergom & Christina Butler
Tufts University Office of Institutional Research
2019 Northeast Association for Institutional Research Annual Conference
Why use regular expressions in Tableau?

- One of the five **Duties and Functions of Institutional Research** stated by AIR is collecting, analyzing, interpreting and reporting data and information (airweb.org, 2019)

- In order to achieve this in a rapidly-changing data environment, IR professionals need to continually **adapt to new technologies and data querying tools**

- Tableau is **increasingly used** in IR for data management, analysis, and visualization
What are regular expressions?

- **Not new or unique to Tableau**; originated in the 1950s and standard in many computer languages and tools, such as Python and MySQL
  - Released in Tableau 9.0

- Help execute **complex queries of text data**—such as addresses or birthdates

- **Extract patterns of text**, similar to wildcard notations, but can be more complex and offer additional functionality
What do regular expressions look like?

Find five consecutive digits:

\[ [0-9]{5} \]

Look for digits

5 in a row
What do regular expressions look like?

Find two consecutive capital letters:

\[ [A-Z]\{2} \]

Look for capital letters only 2 in a row

Regular Expression

\ [/ [A-Z]\{2\] /g \]

Test String

63 Pendergast Lane
Plainfield, NJ 07060
194 Division Dr.
Cranston, RI 02920
28 Westminster Street
Hanover, PA 17331
7351 North La Sierra Ave.
Staunton, VA 24401
97 Bald Hill Street
Shirley, NY 11967
458 Plumb Branch St.
West Deptford, NJ 08096
Regular expressions in Tableau

**REGEXP_REPLACE**(string, pattern, replacement): Returns a copy of the given string where the regular expression pattern is replaced by the replacement string.

**REGEXP_MATCH**(string, pattern): Returns true if a substring matches the regex pattern.

**REGEXP_EXTRACT**(string, pattern): Returns the portion of the string matching the regular expression pattern.

**REGEXP_EXTRACT_NTH**(string, pattern, index): Returns the portion of the string that matches the regular expression pattern. The substring is matched to the nth capturing group, where n is the given index.

Source: [https://www.tableau.com/about/blog/2015/6/become-regular-regular-expressions-39802](https://www.tableau.com/about/blog/2015/6/become-regular-regular-expressions-39802)
Use Case #1:

Reporting Student Addresses
Reporting student addresses

- Bi-annual reporting of student addresses to comply with city ordinances
- Student-entered data (messy!)
- Request for specific format:

<table>
<thead>
<tr>
<th>Street Number</th>
<th>Street Name</th>
<th>Suffix</th>
<th>Unit</th>
<th>Zip Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>61</td>
<td>ADAMS</td>
<td>ST</td>
<td>2</td>
<td>02144</td>
</tr>
<tr>
<td>27R</td>
<td>ALBION</td>
<td>ST</td>
<td>1</td>
<td>02144</td>
</tr>
<tr>
<td>27R</td>
<td>ALBION</td>
<td>ST</td>
<td>1</td>
<td>02144</td>
</tr>
<tr>
<td>27R</td>
<td>ALBION</td>
<td>ST</td>
<td>2</td>
<td>02144</td>
</tr>
<tr>
<td>9</td>
<td>ALDERSEY</td>
<td>ST</td>
<td>3</td>
<td>02144</td>
</tr>
<tr>
<td>21</td>
<td>ALDERSEY</td>
<td>ST</td>
<td>2</td>
<td>02144</td>
</tr>
<tr>
<td>297</td>
<td>ALEWIFE</td>
<td>PKWY</td>
<td>2</td>
<td>02144</td>
</tr>
<tr>
<td>26</td>
<td>APPLETION</td>
<td>ST</td>
<td>2</td>
<td>02144</td>
</tr>
</tbody>
</table>

- Old way in Excel = slow, tedious, reliant on manual checks
- New way in Tableau = more automated, faster, built-in checks
Reporting student addresses

Zip code: Return 5 consecutive digits

<table>
<thead>
<tr>
<th>Address</th>
<th>Zip</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Amherst Way, Princeton Junction, NJ 08550</td>
<td>08550</td>
</tr>
<tr>
<td>1 Austin Road West, Cullinan Ocean Sky 22C, Kowloon HKG</td>
<td>Null</td>
</tr>
<tr>
<td>1 Bellevue St, Saugus, MA 01906-2223</td>
<td>01906</td>
</tr>
<tr>
<td>1 Bond Str., Apt. 6D, Ny, NY 10012</td>
<td>10012</td>
</tr>
</tbody>
</table>

Calculation:

```
REGEXP_EXTRACT([Address], "([0-9]{5})")
```

Tableau function

variable

Regular expression (needs to be in quotes for Tableau)

Find digits (the “quantifier”)

5 in a row

Field we want to create

Returns Null if no match

Data we have to work with
Reporting student addresses

Identify addresses not starting with number

<table>
<thead>
<tr>
<th>Address</th>
<th>Starts with non-number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bush Hall 316</td>
<td>True</td>
</tr>
<tr>
<td>PO Box 64, Quincy, MA 02170</td>
<td>True</td>
</tr>
<tr>
<td>44241 Topaz Way, Fremont, CA 94539</td>
<td>False</td>
</tr>
<tr>
<td>The Fletcher School, 160 Packard Ave Box 76, Medford, MA 02155-5815</td>
<td>True</td>
</tr>
<tr>
<td>39917 Chalon Ct, Temecula, CA 92591</td>
<td>False</td>
</tr>
</tbody>
</table>

Tableau function

```
NOT REGEXP_MATCH(LEFT([Address],1),"\d")
```

Look at left-most character in string

Check if it’s a digit

Returns value of True or False

variable

Regular expression
Reporting student addresses

Street suffix: Identify any of a specific set of strings (returns first match)

Data we have to work with
Extract any of these strings (abbreviated list for demo)
Field we want to create

Calculation:

```
REGEXP_EXTRACT ([Address], '(Street|St |St.|Blvd)')
```

Tableau function variable Regular expression
Regex vs. String Calculation

This regular expression...

```
REGEXP_EXTRACT (Address],[",(Street|St \|St.|Blvd)\'])
```

returns the same result as this string calculation...

```
IF CONTAINS([Address],"Street") THEN "Street"
ELSEIF CONTAINS([Address],"St ") THEN "St 
ELSEIF CONTAINS([Address],"St." ) THEN "St."
ELSEIF CONTAINS([Address],"Blvd") THEN "Blvd"
END
```

(but the regular expression is more concise).
Use Case #2:

Removing special characters for naming PDFs
Removing special characters for naming PDFs

- Every year, over 200 advising reports are generated from Tufts’ Senior Survey

- This process was moved to Tableau and using Tableau’s tabcmd command line utility, PDF reports can be created from a single dashboard

- Each PDF is given a custom file name based on Major and Faculty Name

- In order for this process to run successfully, certain characters must be removed from the Major and Faculty Name strings
Removing special characters for naming PDFs

Remove non-alpha characters for a field used to name PDFs (PDFs cannot accept special characters)

```
REGEX_REPLACE([Major], '[^a-zA-Z]',"")
```

Find any non-alpha characters

Caret ("^") means "is not" when inside the brackets

Replace with nothing
Use Case #3:

Cleaning admissions data
Cleaning admissions data

- OIR received a request to report on admissions data by **Term**

- However, it was discovered that the **Term** field in the data source was unreliable and did not always align with when an application was actually being reported

- The solution was to derive a more accurate **Term** field by considering the year of the corresponding **Admissions Round**
Cleaning admissions data

Fixing data anomalies:
Override term field based on admissions round string field

Find 4 consecutive digits in Term field

REGEXP_REPLACE([Term], '(\d{4})', REGEXP_EXTRACT([Round], '(\d{4})'))

Replace with 4 consecutive digits found in Round field

<table>
<thead>
<tr>
<th>Term</th>
<th>Round</th>
<th>Term (modified)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fall 2015</td>
<td>2015 Doctor of Philosophy (PhD)</td>
<td>Fall 2015</td>
</tr>
<tr>
<td></td>
<td>Fall 2015</td>
<td></td>
</tr>
<tr>
<td>Fall 2016</td>
<td>2016 DVM First Year Student</td>
<td>Fall 2016</td>
</tr>
<tr>
<td></td>
<td>2019 Sackler Non Degree Coursework, Programs, and Visiting...</td>
<td>Fall 2019</td>
</tr>
<tr>
<td>Fall 2017</td>
<td>Certificate - Fall 2017</td>
<td>Fall 2017</td>
</tr>
<tr>
<td>Fall 2018</td>
<td>2018 Exchange</td>
<td>Fall 2018</td>
</tr>
<tr>
<td></td>
<td>2019 Friedman School of Nutrition Science and Policy</td>
<td>Fall 2019</td>
</tr>
<tr>
<td></td>
<td>2019 Sackler Non Degree Coursework, Programs, and Visiting...</td>
<td>Fall 2019</td>
</tr>
<tr>
<td></td>
<td>Fall 2020</td>
<td>Fall 2020</td>
</tr>
<tr>
<td>Fall 2021</td>
<td>2020 Sackler PhD &amp; MS Programs</td>
<td>Fall 2020</td>
</tr>
</tbody>
</table>
Resources

Regex testers:
https://www.regextester.com/
https://www.regexpal.com/
https://regex101.com/

Regular expressions information, library of expressions:
https://github.com/ziishaned/learn-regex
https://www.regular-expressions.info/

Regular Expressions Quick Start Guide:
https://www.regular-expressions.info/quickstart.html

Regular Expressions and Tableau
https://www.tableau.com/about/blog/2015/6/become-regular-regular-expressions-39802
https://code.tutsplus.com/tutorials/8-regular-expressions-you-should-know--net-6149
Tips for improving performance

Materialize Calculations in Your Extracts

Other lists of tips
- https://www.tableau.com/about/blog/2016/1/5-tips-make-your-dashboards-more-performant-48574