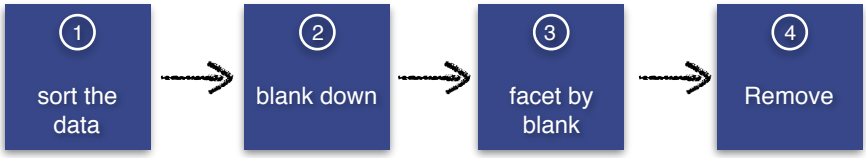
DATENSCHULE
Lernen, wie man Daten richtig nutzt.

OPEN REFINE CHEAT SHEET

QUICK LOOK

Remove Duplicates

For duplicated column:



Function Invocation

Functions can be invoked like the following:

1. <functionName>(arg0, arg1,...)
   or
2. arg0.<functionName>(arg1, ...)

Example:
trim(value) **or** value.trim()

GREL

Replace string

value.replace(<replacement>, <replacer>)

Example: value.replace("&", "and")
replaces ampersand through the string „and"

Remove leading or trailing spaces

value.trim()

Example: " money " —> "money"

Conditional statement

if(<expression>,<if yes>, <if no>)

Example: if (value > 10000, "big", "small")
5000 —> „small", 100000 —> „big"

Note: Conditions can be nested into each other

Access values in other columns

cells['<column name>'].value

Example: cells[‚Amount'].value

Remove unnecessary information

value.match(<replacement>, <replacer>)

Example: value.match("[0-9]{4}")[0]
"founded in 1983" —> "1983"

Difference of two dates

value.diff(<compare_date>, <time unit>)

Example: value.diff("1999-01-11".toDate(), "days")
2000-01-11T00:00:00Z —> 365

Note: Possible time units are amongst others: „days", „minutes", „years", „seconds", „weeks"

Decoding HTML Entities

value.unescape("url")

Example: My%20Awsome%20%26%20only%20url —>
"My Awesome & only url"

Regular Expressions

Format

/<regular expression/

Matching

Expression	Matches
.	one arbitrary character
html$	„html" at the end of a line
^my_	my_ at the beginning of the line
[a-z]	any lowercased letter
[1234]	either 1 or 2 or 3 or 4
[^a-z]	anything execept lowercase letters

Special Characters

Expression	Matches
\d	digits
\s	whitespaces like tabs, spaces or newlines
\w	any word character (letter or number)
\b	word boundary

Note: Big special characters negate the meaning. For example **\D** matches anything but digits, **\S** anything but whitespace characters etc.

Quantifiers

Expression	Matches
X?	at most one time
X*	arbitrary occurences (even zero)
X+	at least one occurrence

DATENSCHULE
Lernen, wie man Daten richtig nutzt.

OPEN REFINE CHEAT SHEET

BEST PRACTICE

CHECK INCONSISTENCIES

Transform columns to their according datatypes (text, number, date) and check whether all values can be converted to find inconsistencies.

You can use facets to indicate inconsistent data.

FIND OUTLIERS

Outliers make your dataset special. By finding these outliers you can check whether the data is reasonable.

You can use more sophisticated functions like standard deviations to find outliers.

OPERATE ON COLUMNS

Always work on columns rather than rows or even single values.

Doing this, you make sure, that all the steps you made are not specific to the document and can be applied to similar datasets.

LEARN REGULAR EXPRESSIONS

Regular expressions are a powerful tool to work on any kind of data. By specifying simple rules, the cleaning of the data gets much easier.

With the use of wildcards and groups your expression get flexible.

USE EXPORTS

All steps performed on columns in Open Refine can be automatically reproduced.

Use exports to replay your modifications to the document to similar datasets. Executed work doesn't have to be repeated.

EAT YOUR OWN DOGFOOD

By consuming your own data you can verify if its clean.

Play around with your data.
Ask questions to the document.
Ask yourself what the dataset should be used for and check the structure

SEPARATE MULTI - VALUED COLUMNS

Every cell should only contain one piece of information. Multi-valued columns are hard to consume. Separate them by using regular expressions

CHECK COLUMN HEADERS

Give each column a reasonable name. The header describes the data in the shortest way.

Pay attention if one could guess the kind of data by only reading the column names.

HELP & DOCUMENTATION

http://openrefine.org/

https://github.com/OpenRefine/OpenRefine/wiki

DATENSCHULE
Lernen, wie man Daten richtig nutzt.

OPEN REFINE CHEAT

BEST PRACTICE

USE CHAINING OVER NESTING

In Open Refine you can use either chaining or nesting of functions. Chaining is far more readable then nesting. Instead of

value.replace()

LOREM IPSUM

Outliers make your dataset special. By finding these outliers you can check whether the data is reasonable.

You can use more sophisticated functions like standard deviations to find outliers

LOREM IPSUM

Always work on columns rather than rows or even single values.

Doing this you ensure, that all the steps you made are not specific to the document and can be applied to similar datasets.

LOREM IPSUM

Regular Expressions are a powerful tool to work on any kind of data. By specifying simple rules the cleaning of the data gets much easier.

With the use of wildcards and groups your expression get flexible.

LOREM IPSUM

Use exports to replay your modifications to the document to similar dataset. Executed work don't have to be repeated.

All Steps performed on Columns in Open Refine can be automatically reproduced.

LOREM IPSUM

Play around with your data. Ask questions to the document.
Try to find out how someone would use the dataset and if the structure is reasonable.

By consuming your own data you can verify if its clean.

LOREM IPSUM

Every cell should only contain one piece of information. Multi-Valued columns are hard to consume. Try to seperate them by using regular expressions

LOREM IPSUM

Give each column a reasonable Name. The Header describes the data in the shortest way. Pay attention if one could guess the kind of data by only reading the column names

LOREM IPSUM

http://openrefine.org/

https://github.com/OpenRefine/OpenRefine/wiki

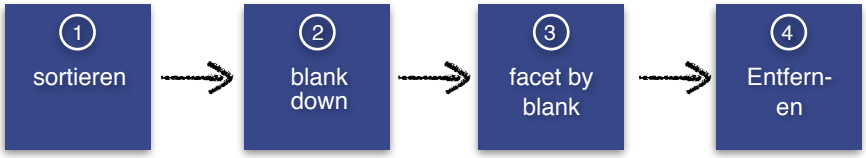
DATENSCHULE
Lernen, wie man Daten richtig nutzt.

OPEN REFINE CHEAT SHEET

QUICK LOOK

Duplikate entfernen

Für die duplizierte Spalte:



Aufrufen von Funktionen

Funktionen können auf verschiedene Arten aufgerufen werden:

1. <funktionsname>(arg0, arg1,...)

oder

2. arg0.<funktionsname>(arg1, ...)

Beispiel:
trim(value) **oder** value.trim()

GREL

Ersetzen von Zeichenketten

value.replace(<ersetzte zeichen>, <Ersetzungstext>)

Beispiel: value.replace("&.", "und")
Ersetzt das & durch die Zeichenkette „und"

Entfernen von voran- und nachgestellten Leerzeichen

value.trim()

Example: " geld " —> "geld"

Bedingte Anweisungen

if(<prüfbedingung>,<wenn ja>, <wenn nein>)

Example: if (value > 10000, "groß", "klein")
5000 —> „groß", 100000 —> „klein"

Notiz: Bedingte Anweisungen können verschachtelt werden.

Zugriff auf andere Spalten

cells['<spaltenname>'].value

Beispiel: cells['Wert'].value

Entfernen unnötiger Informationen

value.match(<ersetzter ausdruck>, <ersetzungsausdruck>)

Beispiel: value.match("[0-9]{4}")[0]
"1983 gegründet" —> "1983"

Rechnen mit Zeitwerten

value.diff(<vergleichsdatum>, <zeitgröße>)

Beispiel: value.diff("1999-01-11".toDate(), "days")
2000-01-11T00:00:00Z —> 365

Notiz: Mögliche Werte sind unter anderem: „days", „minutes", „years", „seconds", „weeks"

Decodieren von HTML Entitäten

value.unescape("url")

Beispiel: My%20Awsome%20%26%20only%20url —>
"My Awesome & only url"

Reguläre Ausdrücke

Format

/<reguläre Ausdruck>/

Matching

Ausdruck	Matches
.	beliebiges Zeichen
html$	„html" am Ende der Zeile
^my_	„my_„ am Anfang der Zeile
[a-z]	beliebiger Kleinbuchstabe
[1234]	entweder 1, 2, 3 oder 4
[^a-z]	Alle Zeichen außer Kleinbuchstaben

Spezialzeichen

Ausdruck	Matches
\d	Zahlen
\s	whitespaces wie Tabs, Leerzeichen oder Leerzeichen
\w	beliebige Worte
\b	Wortgrenzen

Notiz: Spezialzeichen mit großen Buchstaben negieren die Bedeutung. Zum Beispiel matcht **\D** alles AUßER Zahlen, **\S** alles AUßER Whitespaces etc.

Quantifizierer

Ausdruck	Matches
X?	maximal einmal
X*	beliebig oft (auch 0 mal)
X+	mindestens einmaliges Auftauchen

DATENSCHULE
Lernen, wie man Daten richtig nutzt.

OPEN REFINE CHEAT SHEET

BEST PRACTICE

ÜBERPRÜFE INKONSISTENZEN

Konvertiere die Spalten in ihren jeweiligen Datentyp (Text, Zahl, Zeiteinheit) und überprüfe, ob alle Werte konvertiert werden können

Facets können zur Überprüfung eingesetzt werden.

FINDE AUßENSEITER

Außenseiter machen deine Daten speziell. Wenn die Außenseiter identifiziert werden können.

You can use more sophisticated functions like standard deviations to find outliers.

ARBEITE AUF BASIS VON SPALTEN

Always work on columns rather than rows or even single values.

Doing this, you make sure, that all the steps you made are not specific to the document and can be applied to similar datasets.

LERNE REGULÄRE AUSDRÜCKE

Regular expressions are a powerful tool to work on any kind of data. By specifying simple rules, the cleaning of the data gets much easier.

With the use of wildcards and groups your expression get flexible.

NUTZE DEN VERLAUF

All steps performed on columns in Open Refine can be automatically reproduced.

Use exports to replay your modifications to the document to similar datasets. Executed work doesn't have to be repeated.

TESTE DEIN DATENSET

By consuming your own data you can verify if its clean.

Play around with your data. Ask questions to the document.
Ask yourself what the dataset should be used for and check the structure

TEILE ZELLEN MIT MEHREREN WERTEN

Every cell should only contain one piece of information. Multi-valued columns are hard to consume. Separate them by using regular expressions

ÜBERPRÜFE SPALTEN - ÜBERSCHRIFTEN

Give each column a reasonable name. The header describes the data in the shortest way.

Pay attention if one could guess the kind of data by only reading the column names.

HILFE UND DOKUMNETATION

http://openrefine.org/

https://github.com/OpenRefine/OpenRefine/wiki

OPEN REFINE CHEAT SHEET

BEST PRACTICE

USE CHAINING OVER NESTING

In Open Refine you can use either chaining or nesting of functions. Chaining is far more readable then nesting.
Instead of

value.replace()

LOREM IPSUM

Outliers make your dataset special. By finding these outliers you can check whether the data is reasonable.

You can use more sophisticated functions like standard deviations to find outliers

LOREM IPSUM

Always work on columns rather than rows or even single values.

Doing this you ensure, that all the steps you made are not specific to the document and can be applied to similar datasets.

LOREM IPSUM

Regular Expressions are a powerful tool to work on any kind of data. By specifying simple rules the cleaning of the data gets much easier.

With the use of wildcards and groups your expression get flexible.

LOREM IPSUM

Use exports to replay your modifications to the document to similar dataset. Executed work don't have to be repeated.

All Steps performed on Columns in Open Refine can be automatically reproduced.

LOREM IPSUM

Play around with your data. Ask questions to the document.
Try to find out how someone would use the dataset and if the structure is reasonable.

By consuming your own data you can verify if its clean.

LOREM IPSUM

Every cell should only contain one piece of information. Multi-Valued columns are hard to consume. Try to seperate them by using regular expressions

LOREM IPSUM

Give each column a reasonable Name. The Header describes the data in the shortest way. Pay attention if one could guess the kind of data by only reading the column names

LOREM IPSUM

http://openrefine.org/

https://github.com/OpenRefine/OpenRefine/wiki