

# LENA 2.0 Manual

## Aufbau

Die neue Version von Lena kann Abstimmungsergebnisse in Echtzeit verarbeiten, Texte basierend auf den Wahlergebnissen schreiben und direkt auf Datawrapper-Karten darstellen. Zudem ist neu ein historischer Vergleich mit früheren Abstimmungen möglich. Lena 2.0 wurde mit der Statistiksoftware R erstellt und ist folgendermassen aufgebaut:

### [lena\\_run.R](#)

Der Hauptprozess von Lena. Hier werden die nötigen Funktionen und Daten geladen, die Daten angepasst, die Storybausteine evaluiert, die Texte zusammengestellt und optimiert sowie der Output generiert für Datawrapper (csv-Files)

### [config.R](#)

Hier sind die nötigen Bibliotheken und Datenquellen hinterlegt.

### [functions\\_readin.R](#)

Hier sind alle Funktionen hinterlegt, die es zum Einlesen der Daten braucht.

### [functions\\_storyfinder.R](#)

Hier sind die Funktionen hinterlegt, die LENA zum finden der passenden Storybausteine braucht, z.B. Intros, Lena Classics und historische sowie innerkantonale Vergleiche. Diese Funktionen sind je nach Bedürfnis beliebig erweiterbar.

### [functions\\_storybuilder.R](#)

Hier sind die Funktionen hinterlegt, mit denen LENA schlussendlich basierend auf dem Storyfinder die Texte generiert und optimiert.

## Flexible Elemente / Anpassungen für zukünftige Abstimmungen

### Textbausteine

Die einzelnen Textbausteine sind in einem Excel hinterlegt und können von der Redaktion beliebig angepasst werden. Die einzelnen Registerkarten (unten) sind nach den Abstimmungen benannt und müssen mit dem Kurznamen Vorlagen (siehe unten) übereinstimmen. Das Excel wird in in lena\_run.R unter «Textvorlagen laden» eingelesen. Beispiel Abstimmungen September 2020: [https://docs.google.com/spreadsheets/d/13NcWyQ\\_wr0WfstCruuMRW7zE6NqHHify0noJhPdXXu0/edit?usp=sharing](https://docs.google.com/spreadsheets/d/13NcWyQ_wr0WfstCruuMRW7zE6NqHHify0noJhPdXXu0/edit?usp=sharing)

### JSON-Link zu den Echtzeitergebnissen

Der Link zu den aktuellen Abstimmungsergebnissen muss in config.R hinterlegt werden unter «Link zu JSON-Daten». Der Link ist in der Regel mehrere Wochen vor der Abstimmung hier zu finden: <https://opendata.swiss/de/dataset/echtzeitdaten-am-abstimmungstag-zu-eidgenoessischen-abstimmungsvorlagen>

### Metadaten Gemeinden und Kantone

Im CSV-File «MASTERFILE\_GDE.csv» sind alle notwendigen Daten zu den Gemeinden und Kantonen hinterlegt (Gemeindenummern, Kurznamen, französische Namen ect.) . Dieses File muss angepasst werden, wenn es Veränderungen bei den Gemeinden gibt, wie z.B. Fusionen. Das File wird in config.R unter «Metadaten Gemeinden und Kantone» eingelesen.

### Kurznamen Vorlagen

Für jede Vorlage muss ein Kurzname definiert werden (z.B. «Kampfjet»). Diese werden unter «Kurznamen Vorlagen» in config.R hinterlegt (Achtung: Reihenfolge beachten, ist in JSON vorgegeben!)

### Historische Daten

Wird ein historischer Vergleich gewünscht, müssen die Abstimmungsdaten zur entsprechenden Vorlage im Excel-Format zur Verfügung gestellt werden. Die Daten sind auf der Seite des BFS zu finden. Hier z.B. für die Masseneinwanderungs-Initiative: <https://www.bfs.admin.ch/asset/de/je-d-17.03.03.cz.580.c>

Die Daten aus dem Excel werden in der config.R Datei geladen unter «Vorhandene Daten laden»

In lena\_run.R wird unter «Historischer Vergleich» definiert, bei welchen Vorlagen der historische Vergleich stattfindet. Hier muss die korrekte Vorlagen-ID (vierstellige Zahl) hinterlegt werden, wie sie im JSON zu finden ist. Bei der Zuwanderungs-Initiative wäre dies z.B. «6310»

### Innerkantonale Vergleiche

In lena\_run.R muss unter «Vergleich innerhalb des Kantons» definiert werden, bei welchen Vorlagen dieser Vergleich stattfinden soll. Auch hier muss die Vorlagen-ID hinterlegt werden.

## Output

Lena erzeugt für jede Vorlage ein csv-File mit den nötigen Angaben für Datawrapper. Diese Files werden im Ordner «Output» gespeichert. Damit die Live-Anbindung funktioniert, weist das File immer gleich viele Zeilen aus. D.h. auch wenn bei einer Gemeinde noch keine Dateien vorhanden sind, wird trotzdem eine Zeile erzeugt.

## Verknüpfung mit Github

Damit Datawrapper online auf die Daten zugreifen kann, werden die Output-Files auf Github gespeichert und dann geupdated. Dies kann durch manuelles Hochladen in Github geschehen oder aber durch eine automatische Verknüpfung eines R-Projekts mit Github (mehr dazu hier:

<https://happygitwithr.com/rstudio-git-github.html>)

Beispiel der Abstimmung im September 2020: <https://github.com/awp-finanznachrichten/lena/tree/master/Output>

## Live-Feed zu Datawrapper-Karten

In Datawrapper muss nun der Link zum Github-File mit den Daten zur Abstimmung hinterlegt werden. Wie eine Live-Anbindung der Karten genau funktioniert, kann hier entnommen werden:

<https://academy.datawrapper.de/article/236-how-to-create-a-live-updating-symbol-map-or-choropleth-map>

**Achtung:** Damit die Daten korrekt gelesen werden, ist es entscheidend, dass die Spalte «Gemeinde\_Nr» als Text, und **nicht** als Zahl erkannt wird. Dies lässt sich in Datawrapper unter <https://app.datawrapper.de/chart/xxxxx/describe> einstellen. Einfach Spaltentyp anwählen und als Spaltentyp «Text» definieren.

Unter Schritt 2 «Füge Daten Hinzu» muss zudem beim Geo-Code «ID» oder «Codes» hinterlegt sein, und nicht «Namen».

## Github Lena 2.0

Alle Bestandteile von Lena 2.0 (Stand September 2020) ist in folgendem Github-Repository zu finden:

<https://github.com/awp-finanznachrichten/lena>