

Datenjournalismus HS19 - FHNW

```
library(tidyverse)
```

```
## -- Attaching packages ----- tidyverse 1.3.0 --
```

```
## <U+2713> ggplot2 3.2.1      <U+2713> purrr  0.3.3
## <U+2713> tibble  2.1.3      <U+2713> dplyr  0.8.3
## <U+2713> tidyr   1.0.0      <U+2713> stringr 1.4.0
## <U+2713> readr   1.3.1      <U+2713> forcats 0.4.0
```

```
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(dplyr)
library(ggplot2)
library(readxl)
```

Das Vorgehen bei der Analyse der Daten für potentielle neue Storys

1. Daten laden und untersuchen
2. Daten untersuchen und bereinigen (wenn nötig und wo sinnvoll)
3. Daten aggregieren und grafisch darstellen
4. Daten interpretieren
5. Iteration → Neustarten bei Punkt 2 - Daten bereinigen

1. Daten laden und untersuchen

Für die Bearbeitung der Challenge, stehen 4 Datensätze zur Verfügung. Jeder von ihnen mit einem anderen Detailgrad und dementsprechend auch mit unterschiedlicher Datengrösse. In der "Readme" Datei der Datensätze wird ersichtlich, dass die Unterschiede und Grössen der Files auf der Anzahl Spalten und damit möglichen Details basieren.

Full release = 67 Variablen (Spalten)

Thin release = 8 Variablen (Spalten)

Donor/Recipient/Year Aggregated Release = 4 Variablen (Spalten)

Donor/Recipient/Year/Purpose Aggregated Release = 6 Variablen (Spalten)

In allen 4 Dokumenten wurde die gleiche Höhe von Hilfsgeldern aufzeigen, jedoch durch die Wahl der Spalten gibts es unterschiedlich grosse Dokumente.

Da gemäss der Readme Datei die kleineren Dokumente eine Zusammenfassung der kompletten Datei sind wollte ich gleich mit allen Daten anfangen um keine Datenverluste zu haben und die Einteilungen in Kategorien, wenn nötig, selber durchführen kann.

```
# Import necessary data sets
```

```
all_data_full <- read.table("AidDataCoreFull_ResearchRelease_Level1_v3.1.csv", header=TRUE, sep=",")
```

2. Daten untersuchen und bereinigen

In dieser Arbeit werden die Daten zu Hilfgeldzahlung zum Land Kroatien untersucht. Aus diesem Grund wird im ersten Schritt ein Subset an Daten erstellt wobei Kroatien als Empfänger ersichtlich ist.

```
subset_recipient_croatia <- subset(all_data_full, recipient == "Croatia")
```

Um eine möglichst gute Vergleichbarkeit der Daten herstellen zu können, ist es wichtig die Daten inhaltlich aggregieren zu können auf den Ebenen der Spender, Empfänger und Gründe der Hilfgeldzahlungen. Basierend auf der Analyse in der Vorschau das vollumfängliche Dokument werden 8 Spalten extrahiert und die Anzahl leere Felder analysiert um die bestmögliche Datengrundlage zu schaffen.

```
subset_condensed <- select(subset_recipient_croatia, year, donor, recipient, aiddata_sector_name, crs_sector_name, commitment_amount_usd_constant)

check_all_data_condensed <- summary(subset_condensed)

check_all_data_condensed
```

##		year	donor	
##	Min.	:1989	Germany	: 484
##	1st Qu.:	:2001	United States	: 394
##	Median	:2004	France	: 359
##	Mean	:2004	Norway	: 303
##	3rd Qu.:	:2007	European Communities (EC)	: 251
##	Max.	:2011	Italy	: 200
##			(Other)	:1453
##			recipient	
##	Croatia			:3444
##	Afghanistan			: 0
##	Africa, North of Sahara, Regional Programs			: 0
##	Africa, Regional Programs Multi-Country			: 0
##	Africa, Regional Programs, Regional Programs:			0
##	Africa, South of Sahara Multi-Country			: 0
##	(Other)			: 0
##			aiddata_sector_name	
##				: 622
##	Government and civil society, general			: 561
##	Other			: 293
##	Other social infrastructure and services:			281
##	Emergency response			: 187
##	Post-secondary education			: 179
##	(Other)			:1321
##			crs_sector_name	
##	I.5.a. Government & Civil Society-general			: 679
##	I.1.d. Post-Secondary Education			: 399
##	I.6. Other Social Infrastructure & Services:			343
##	IV.2. Other Multisector			: 302
##	VIII.1. Emergency Response			: 196
##				: 191
##	(Other)			:1334
##	commitment_amount_usd_constant			
##	Min.	:	19	
##	1st Qu.:	:	18111	
##	Median	:	83636	
##	Mean	:	4264646	
##	3rd Qu.:	:	575662	
##	Max.	:	414177461	
##				

von den 3444 Observationen gibt es folgende Anzahl leere oder "other" in den Spalten "crs_sector_name" und "aiddata_sector_name":

- aiddata_sector_name: 2236
- crs_sector_name: 1827

Bei näherer Betrachtung in der Vorschau ist zu sehen, dass oft der Fall vorzufinden ist, dass entweder die eine oder die andere Spalte keinen Wert besitzt. Die Variablen Land, Jahr und Beitrag haben keine leeren Zellen oder "NA"s.

```
data_croatia <- subset_condensed
```

Durch Zeitrestriktionen und relativ komplexen Strukturen in den Daten, werden im nächsten Schritt, Zwecks Effizienz, die Daten in ein CSV geschrieben und im Excel so formatiert, dass die Kategorien einfach vergleichbar sind. Dazu gehört:

1. Zusätzliche Spalte erstellen für "high_level_category"
2. Kategorie wenn in beiden Spalten (aiddata_sector_name + crs_sector_name) die selbe Info besteht, übernehmen
3. Falls nur ein Feld die Information hat diese übernehmen
4. Falls die Felder unterschiedliche Informationen haben, ist die Spalte "aiddata_sector_name" führend
5. Die Datei in drei Unterkategorien teilen:

5.1 Basierend auf Spenden per Nation

5.2. Basieren auf Spenden per Organisation

5.3. Auf keiner Zusammenfassun basierend

```
write.csv(data_croatia, "data_croatia_to_clean.csv")
```

Im Exce File wurde die Kategorisierung in folgende Sparten eingeteilt:

1. Administrative Cost
2. Business and Production
3. Education
4. Emergency Aid
5. Environment
6. Health
7. Infrastructure
8. Multisector
9. Other
10. Social, Security and Government

```
data_croatia_donor_nations_clean <- read_excel("data_croatia_clean_nations.xlsx")
data_croatia_donor_organisations_clean <- read_excel("data_croatia_clean_organisations.xlsx")
data_croatia_donor_all_clean <- read_excel("data_croatia_clean_all.xlsx")
```

3. Daten aggregieren und grafisch darstellen

Nachdem nun die Daten in einer korrekten Form zur Verfügung stehen, können im nächsten Schritt die Vorbereitungen für die Grafiken durchgeführt werden um dann im Nachgang die Grafiken selbst zu erstellen.

```
category_sum_nations = group_by(data_croatia_donor_nations_clean, year, high_level_category)
category_sum_nations = summarise(category_sum_nations,
                                  sum_amount = sum(amount_in_USD))

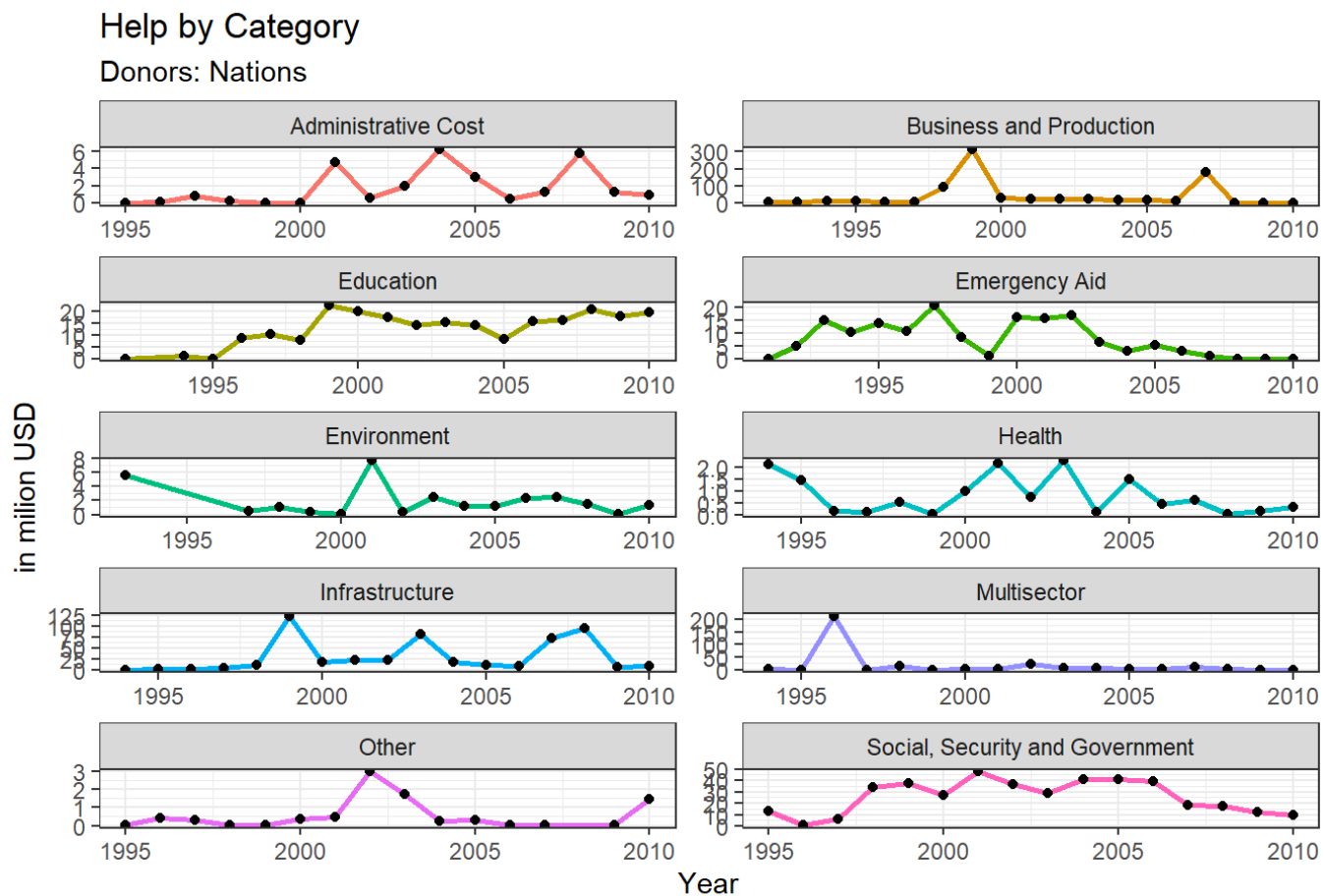
category_sum_organisations = group_by(data_croatia_donor_organisations_clean, year, high_level_category)
category_sum_organisations = summarise(category_sum_organisations,
                                       sum_amount = sum(amount_in_USD))

donation_sum_all_by_year = group_by(data_croatia_donor_all_clean, year)
donation_sum_all_by_year = summarise(donation_sum_all_by_year,
                                     sum_amount = sum(amount_in_USD))

donation_sum_nations = group_by(data_croatia_donor_nations_clean, donor)
donation_sum_nations = summarise(donation_sum_nations,
                                  sum_amount = sum(amount_in_USD))
```

```
distribution_help_nations <- ggplot(data = category_sum_nations, aes (x = year, y = sum_amount/1000000
)) +
  geom_line(aes(color = high_level_category), show.legend = FALSE, size = 1) +
  geom_point() +
  facet_wrap(~high_level_category, ncol = 2, scales = "free") +
  labs(title = "Help by Category",
       subtitle = "Donors: Nations",
       caption = "Source: AidData, 2017")
  )+
  xlab("Year") +
  ylab("in milion USD")

distribution_help_nations +theme_bw() +scale_fill_discrete(name = "Category")
```



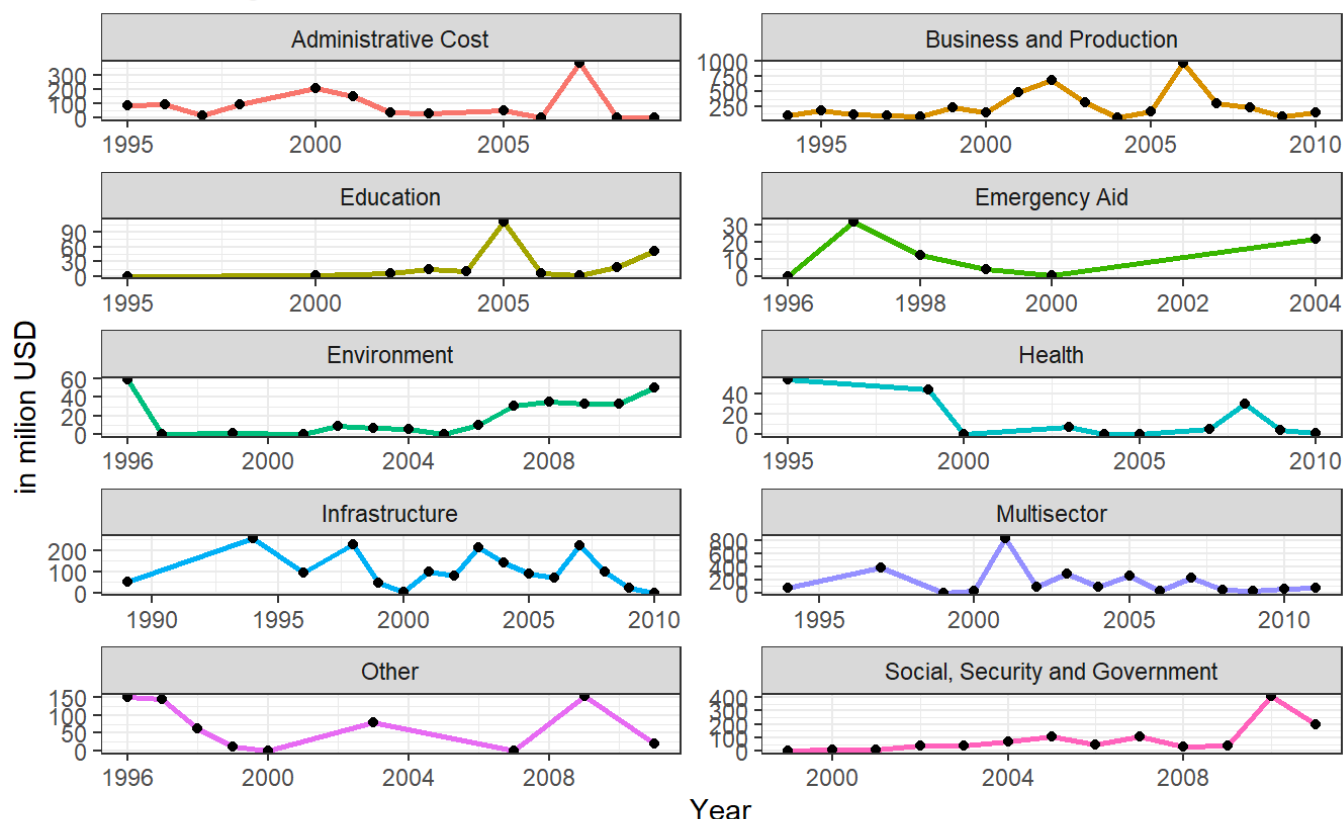
Source: AidData, 2017

```
distribution_help_organisations <- ggplot(data = category_sum_organisations, aes (x = year, y = sum_amount/1000000)) +
  geom_line(aes(color = high_level_category), show.legend = FALSE, size = 1) +
  geom_point() +
  facet_wrap(~high_level_category, ncol = 2, scales = "free") +
  labs(title = "Help by Category",
       subtitle = "Donors: Organisations",
       caption = "Source: AidData, 2017")+
  xlab("Year") +
  ylab("in milion USD")
```

```
distribution_help_organisations + theme_bw()
```

Help by Category

Donors: Organisations



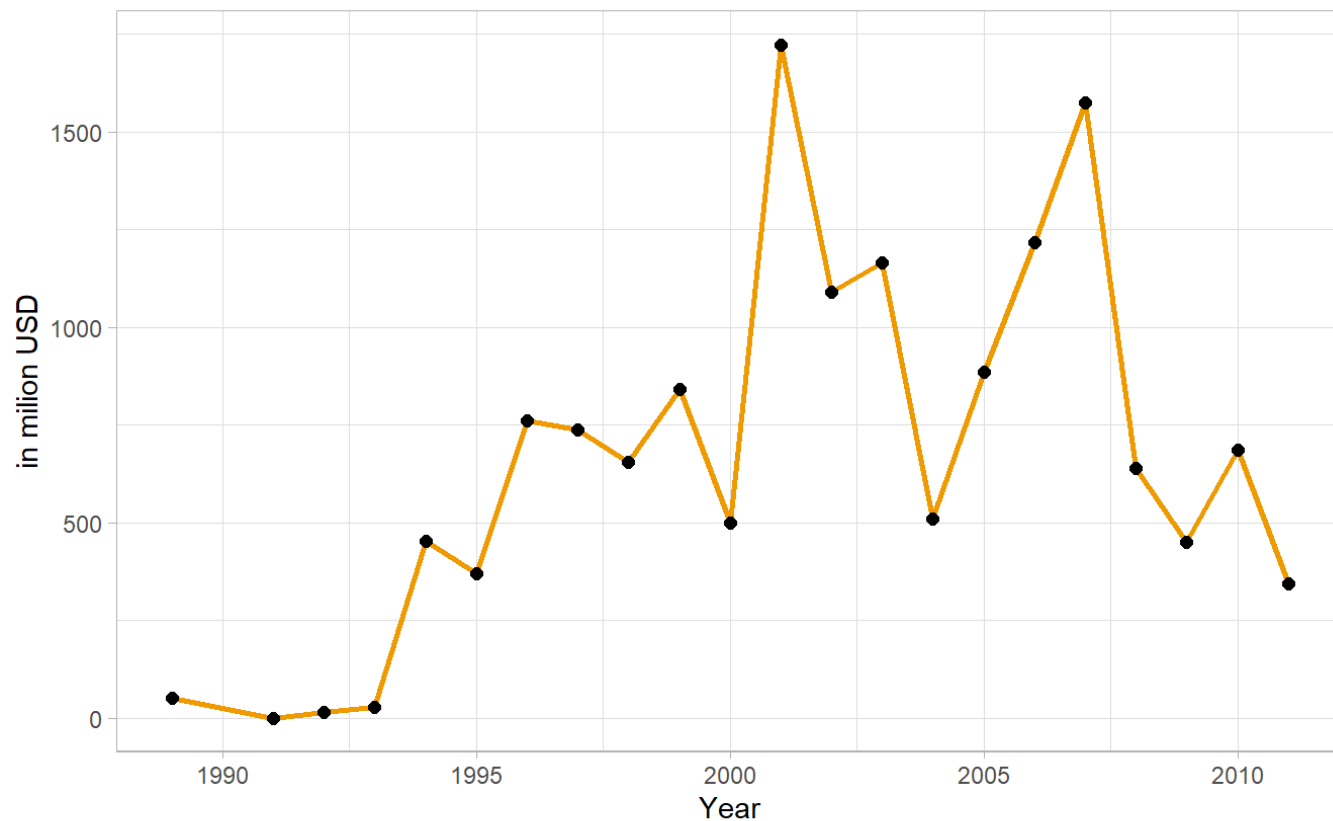
Source: AidData, 2017

```
international_aid <- ggplot(data = donation_sum_all_by_year) +
  geom_line( mappin = aes (x = year, y = sum_amount/1000000), linetype=1, size=1, color="orange2") +
  geom_point(mapping = aes( x = year, y = sum_amount/1000000), shape = 19, size=2) +
  labs(title = "International Aid (in milion USD)",
       subtitle = "Donors: All",
       caption = "Source: AidData, 2017")+
  xlab("Year") +
  ylab("in milion USD")

international_aid + theme_light()
```

International Aid (in million USD)

Donors: All



Source: AidData, 2017

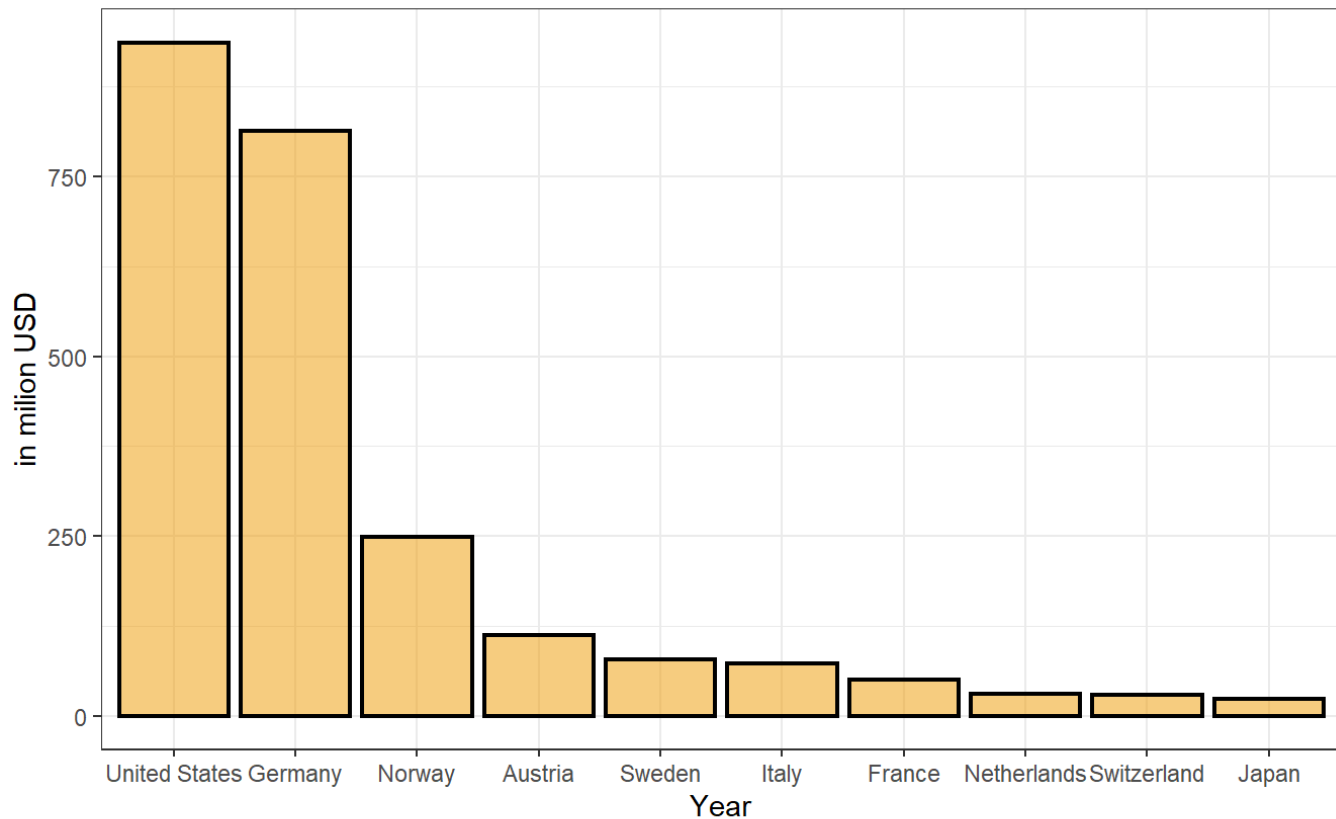
```
donation_sum_nations_top_10 <- donation_sum_nations %>% top_n(10)
```

```
## Selecting by sum_amount
```

```
donors_top_10 <- ggplot (data = donation_sum_nations_top_10, aes(reorder(donor, -sum_amount), sum_amount/1000000)) +  
  geom_bar(stat='identity', show.legend = FALSE, fill="orange2", colour="black", size=0.8, alpha=0.5)  
+  
  labs(title="Top 10 Amounts of Aid by Nations",  
        subtitle="Descending order in mil. USD",  
        caption="Source: AidData, 2017")+  
  xlab("Year") +  
  ylab("in milion USD")  
  
donors_top_10 + theme_bw()
```

Top 10 Amounts of Aid by Nations

Descending order in mil. USD



Source: AidData, 2017

4. Daten interpretieren

Die Interpretation der Daten wird in einer Story auf Medium durchgeführt. Die Story an sich kann unter folgendem Link gelesen werden:

Go to Medium- Story (<https://medium.com/@eugen.cuic/why-was-croatia-supported-after-the-balkan-war-and-by-who-fc78edc0c4cb?sk=27cd804cc22b6f399f3b0d6616724bc4>)

5. Iteration

Weitere zusätzliche Iterationen können stattfinden nachdem die Daten online publiziert worden sind und möglicherweise neue Informationen durch Kommentare auf Medium hinzugefügt werden. Andererseits können in einer vertiefenderer Arbeit auch zusätzliche Aspekte beleuchtet werden. In dieser Arbeit gibt es aber keine weitere Iteration