

# Semesterarbeit Teil 2

AWD FS 2018

Simon Egli [simon.egli@students.ffhs.ch]

# 1 Inhaltsverzeichnis

---

2	Abbildungsverzeichnis .....	4
3	Tabellenverzeichnis .....	4
4	Vorbereitung .....	5
5	Konzept .....	6
5.1	Einleitung .....	6
5.2	Ausgangslage .....	6
5.3	Vorbereitung .....	6
5.3.1	Globale Erfüllungskriterien .....	6
5.3.2	Arbeitspakete .....	7
6	Realisierung .....	8
6.1	Arbeitspaket «Funktionsgraphen anzeigen» .....	8
6.1.1	Quellcode .....	8
6.1.2	Resultat .....	8
6.2	Arbeitspaket «Mehrere Funktionsgraphen in derselben Grafik» .....	9
6.2.1	Analyse .....	9
6.2.2	Quellcode .....	9
6.2.3	Resultat .....	9
6.3	Arbeitspaket «Forschungsdaten als CSV bereitstellen» .....	10
6.3.1	Analyse .....	10
6.3.2	Quellcode .....	10
6.3.3	Resultat .....	10
6.4	Arbeitspaket «Balkendiagramme» .....	12
6.4.1	Analyse .....	12
6.4.2	Quellcode .....	12
6.4.3	Resultat .....	12
6.5	Arbeitspaket «Tortendiagramme» .....	13
6.5.1	Analyse .....	13
6.5.2	Quellcode .....	13
6.5.3	Resultat .....	13
6.6	Arbeitspaket «Histogramme» .....	14
6.6.1	Analyse .....	14

6.6.2	Quellcode .....	14
6.6.3	Resultat .....	14
6.7	Arbeitspaket «Kann-Ziel» .....	15
6.7.1	Analyse.....	15
6.7.2	Quellcode.....	15
6.7.3	Resultat .....	15
6.8	Auswertung.....	16

## 2 Abbildungsverzeichnis

---

Figure 1: Darstellung der Sinus Funktion .....	8
Figure 2: Darstellung der Sinus und Cosinus Funktion .....	9
Figure 3: Häufigkeit der Ereignisse pro Jahr.....	12
Figure 4: Flächenvisualisierung Vulkangrösse und Erdbeben-Magnitude .....	13
Figure 5: Verteilung über die geografische Breite .....	14
Figure 6: Örtliche Analyse der Datensätze.....	15

## 3 Tabellenverzeichnis

---


Table 1: Definition der Arbeitspakete .....	7
Table 2: Metadaten der CSV Dateien .....	11

## 4 Vorbereitung

---

Dieses Dokument beinhaltet die Lösung der Semesterarbeit Teil 2. Das gestellte Problem wird nach dem Ansatz «Think first, then act» gelöst.

Erst nach der theoretischen Konzeption sollen Python Funktionen implementiert werden. Die so geschaffenen Quellcode-Referenzen werden wie folgt dargestellt:



```
from PythonDatei import *  
print( funktion( parameter ) )
```

Unter Berücksichtigung des Dateipfads, können diese kopiert und eigenständig ausgeführt werden.

Als Arbeitsumgebung (IDE) wird Visual Studio Code verwendet:

<https://code.visualstudio.com/>

Um Python entwickeln zu können, wird die von Microsoft zur Verfügung gestellte Extension verwendet:

<https://marketplace.visualstudio.com/items?itemName=ms-python.python>

Zudem wird der Funktionsumfang der IDE mit der Extension Code Runner erweitert:

<https://marketplace.visualstudio.com/items?itemName=formulahendry.code-runner>

Wichtige Erkenntnisse und Hinweise werden dargestellt als:



Dies ist ein wichtiger Hinweis.

## 5 Konzept

---

Um die Aufgabenstellung effizient zu lösen, soll die Arbeit in übersichtliche Arbeitspakete unterteilt werden.

### 5.1 Einleitung

Eine möglichst realistische Aufgabenstellung ist wünschenswert. Aus diesem Grund sollen öffentliche Forschungsdaten visualisiert werden. Im experimentellen Verfahren sollen mögliche Relationen zwischen Vulkanismus und Erdbeben grafisch erstellt werden.

### 5.2 Ausgangslage

Als Grundlage werden globale Daten folgender U.S Staatsorganisation herangezogen, welche wenn möglich verwendet werden:

The Smithsonian Institution: <https://volcano.si.edu>

U.S. Geological Survey: <https://earthquake.usgs.gov>



Um den Rahmen der Arbeit nicht zu sprengen, muss auf den spannendsten Teil, die Interpretation der Ergebnisse, verzichtet werden.

### 5.3 Vorbereitung

Unter Berücksichtigung der Aufgabenstellung, des Modulplans und den Richtlinien für die Semesterarbeit wird folgende Planung aufgestellt.

#### 5.3.1 Globale Erfüllungskriterien

- In Python geschrieben
- Graphen werden mittels matplotlib erzeugt
- Unit Tests sind nicht erforderlich
- Dokumentation ist zugleich dieses Dokument
- Alle relevanten Arbeitsabläufe und Resultate sind dokumentiert

### 5.3.2 Arbeitspakete

Die gestellte Aufgabe wird in folgende Arbeitspakete unterteilt:

Nr.	Name	Erfüllungskriterien
1	Funktionsgraphen anzeigen	<ul style="list-style-type: none"> <li>Die Sinus-Funktion wird als Grafik dargestellt</li> <li>Mindestens eine volle Periode ist ersichtlich</li> <li>X- und Y-Achse sind beschriftet</li> <li>Grafik wird angezeigt</li> </ul>
2	Mehrere Funktionsgraphen in derselben Grafik	<ul style="list-style-type: none"> <li>Die Sinus- und Cosinus-Funktion wird als Grafik dargestellt</li> <li>Mindestens eine volle Periode ist ersichtlich</li> <li>X- und Y-Achse sind beschriftet</li> <li>Grafik wird angezeigt</li> </ul>
3	Forschungsdaten als CSV bereitstellen	<ul style="list-style-type: none"> <li>Seismologische Forschungsdaten liegen als CSV bereit</li> <li>Eruptionsdaten liegen als CSV bereit</li> </ul>
4	Balkendiagramme	<ul style="list-style-type: none"> <li>Geeignete Daten zur Gegenüberstellung identifiziert</li> <li>Achsen sind beschriftet</li> <li>Titel ist vorhanden</li> <li>Grafik wird angezeigt</li> </ul>
5	Tortendiagramme	<ul style="list-style-type: none"> <li>Geeignete Daten zur Gegenüberstellung identifiziert</li> <li>Die Anteile der jeweiligen Gruppen sind ersichtlich</li> <li>Alle Tortendiagramm-Anteile sind beschriftet</li> <li>Grafik wird angezeigt</li> </ul>
6	Histogramme	<ul style="list-style-type: none"> <li>Geeignete Daten zur Gegenüberstellung identifiziert</li> <li>Achsen sind beschriftet</li> <li>Titel ist vorhanden</li> <li>Grafik wird angezeigt</li> </ul>
7	Kann-Ziel	<ul style="list-style-type: none"> <li>Weitere, spezielle Diagramme</li> </ul>

Table 1: Definition der Arbeitspakete

## 6 Realisierung

---

Im Folgenden werden die Arbeitspakete implementiert und dokumentiert.



Eine Detailgetreue Dokumentation der einzelnen Python Implementation sprengt den Rahmen von zwei bis drei A4 Seiten Text (1500 Wörter). Aus diesem Grund enthalten die Python Dateien aussagekräftige Inline-Dokumentationen.

### 6.1 Arbeitspaket «Funktionsgraphen anzeigen»

Gemäss dem Arbeitspaket wurde die Sinus-Funktion als Grafik abgebildet.

#### 6.1.1 Quellcode

Der Quellcode befindet sich in der Datei «SEgli\_02\_Sinus\_Graph.py» im Anhang.



```
from SEgli_02_Sinus_Graph import *  
show_sin()
```

#### 6.1.2 Resultat

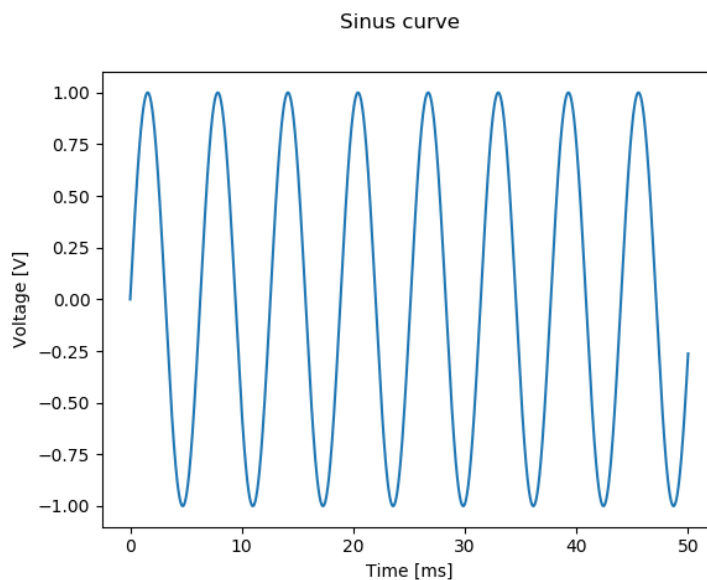


Figure 1: Darstellung der Sinus Funktion



## 6.2 Arbeitspaket «Mehrere Funktionsgraphen in derselben Grafik»


Der Hauptunterschied in Bezug zum Arbeitspaket 1 ist, dass hier zwei Graphen gezeichnet werden sollen.

### 6.2.1 Analyse

Der Source Code aus dem Arbeitspaket kann als Grundlage verwendet werden. Um die Unterscheidbarkeit der beiden Arbeitspakete zu gewährleisten, wird diese Aufgabe in einer neuen Quellcode-Datei eingepflegt. Gemäss Dokumentation, kann die Methode «plot» mehrmals aufgerufen werden, um mehrere Graphen innerhalb eines Diagramms darzustellen.

### 6.2.2 Quellcode

Der Quellcode befindet sich in der Datei «SEgli\_02\_Sinus\_Cosinus\_Graph.py» im Anhang.

```
 from SEgli_02_Sinus_Cosinus_Graph import *  
show_sin_cos_graph()
```

### 6.2.3 Resultat

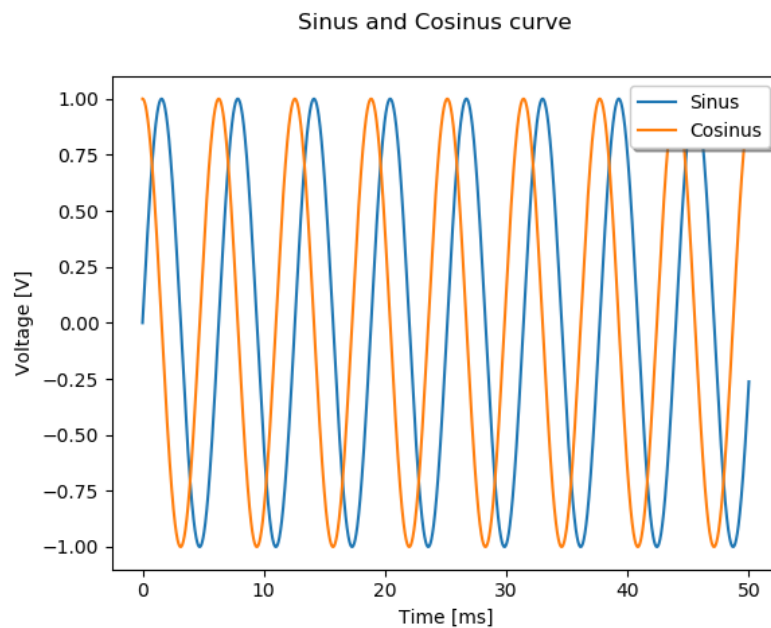


Figure 2: Darstellung der Sinus und Cosinus Funktion

## 6.3 Arbeitspaket «Forschungsdaten als CSV bereitstellen»

Ziel: Bereitstellung der CSV Dateien.

### 6.3.1 Analyse

Beide vorgängig eruierten Institutionen bieten einen Webservice, mit dessen Hilfe historische sowie Echtzeit-Daten abgefragt werden können:

The Smithsonian Institution: <https://volcano.si.edu/database/webservices.cfm>

U.S. Geological Survey: <https://earthquake.usgs.gov/fdsnws/event/1>

Die oben verlinkten Webservices können genutzt werden, um kostenlos Daten abzufragen und im CSV Format bereit zu stellen.

### 6.3.2 Quellcode

Der Quellcode zu dieser Aufgabe wurde nicht in Python implementiert und wird als nicht relevant eingestuft.

### 6.3.3 Resultat

Die Daten vom Smithsonian Institute liegen als «Eruption.csv», diejenigen vom U.S Geological Survey als «Earthquake.csv» vor.



Der Datensatz vulkanischer Aktivitäten und Ausbrüche beinhaltet ausschliesslich Ereignisse des Holozäns.



Der seismographische Datensatz enthält ausschliesslich signifikante Ereignisse. Der Begriff „signifikant“ wurde durch das U.S Geological Survey wie folgt definiert: <https://earthquake.usgs.gov/earthquakes/browse/significant.php>

Die CSV Dateien beinhalten folgende Metadaten:

Eruption.csv		Earthquake.csv	
Number	Numeric	Date	DateTime
Name	String	Time	DateTime
Country	String	Latitude	Numeric
Region	String	Longitude	Numeric
Type	String	Type	String
Activity Evidence	String	Depth	Numeric
Last Known Eruption	String	Depth Error	String
Latitude	Numeric	Depth Seismic Stations	String
Longitude	Numeric	Magnitude	Numeric
Elevation (Meters)	Numeric	Magnitude Type	String
Dominant Rock Type	String	Magnitude Error	String
Tectonic Setting	String	Magnitude Seismic Stations	String
		Azimuthal Gap	String
		Horizontal Distance,	String
		Horizontal Error	String
		Root Mean	String

Table 2: Metadaten der CSV Dateien

Um die Datensätze verwenden zu können, müssen diese weiter aufbereitet werden. Zum Beispiel sollen alle seismographischen Datensätze mit nuklearem Ursprung ausgeschlossen werden. Dazu sind erweiterte Filterfunktionen notwendig, die in Teilaufgabe 1b nicht implementiert wurden. Aus diesem Grund wird das Modul Pandas verwendet um CSV Daten zu lesen.

## 6.4 Arbeitspaket «Balkendiagramme»

Ziel: Bereitstellung der Balkendiagramme.

### 6.4.1 Analyse

Die Datensätze aus Arbeitspaket 3 werden genutzt um eine Gegenüberstellung mittels Balkendiagramm zu erstellen.

In einem Fenster sollen zwei Diagramme gezeigt werden, die ausgehend vom Jahr 1965 die Anzahl Ereignisse pro Jahr ausweisen.

### 6.4.2 Quellcode

Der Quellcode befindet sich in der Datei «SEgli\_02\_Count\_By\_Year.py» im Anhang.



```
from SEgli_02_Count_By_Year import *  
show_count_by_year()
```

### 6.4.3 Resultat

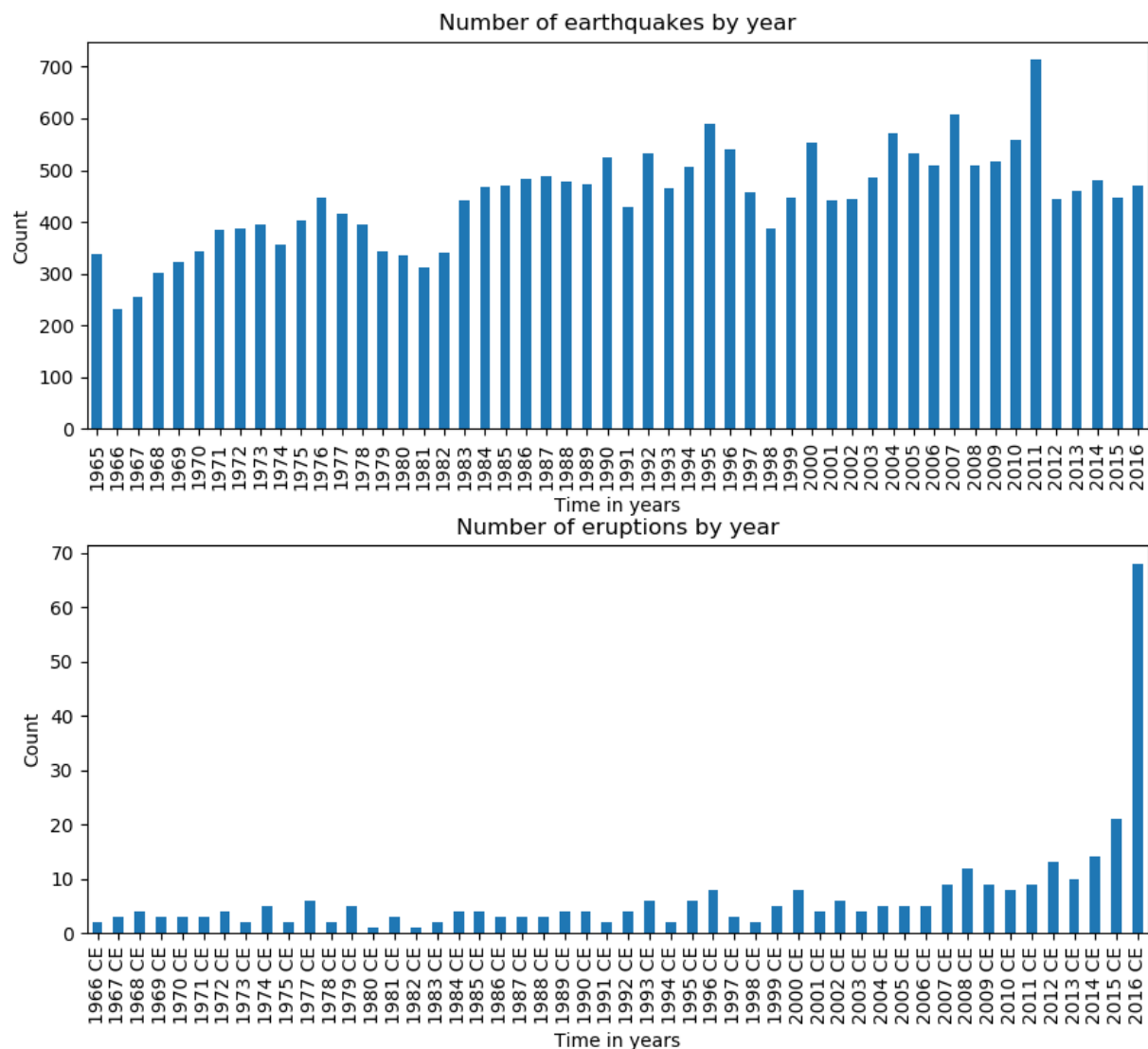


Figure 3: Häufigkeit der Ereignisse pro Jahr

## 6.5 Arbeitspaket «Tortendiagramme»

Ziel: Bereitstellung der Tortendiagramme.

### 6.5.1 Analyse

Die Datensätze aus Arbeitspaket 3 werden genutzt um eine Gegenüberstellung mittels Tortendiagramm zu erstellen.

In einem Fenster sollen zwei Diagramme gezeigt werden, die einerseits die Magnitude eines Erbebens (aufgerundet auf ganze Zahlen) und andererseits die Höhe der Vulkane (aufgerundet auf 1000 Meter) gruppiert in einem Tortendiagramm darstellen.

### 6.5.2 Quellcode

Der Quellcode befindet sich in der Datei «SEgli\_02\_Elevation\_Vs\_Magnitude.py» im Anhang.

```
from SEgli_02_Elevation_Vs_Magnitude import *  
show_pie()
```

### 6.5.3 Resultat

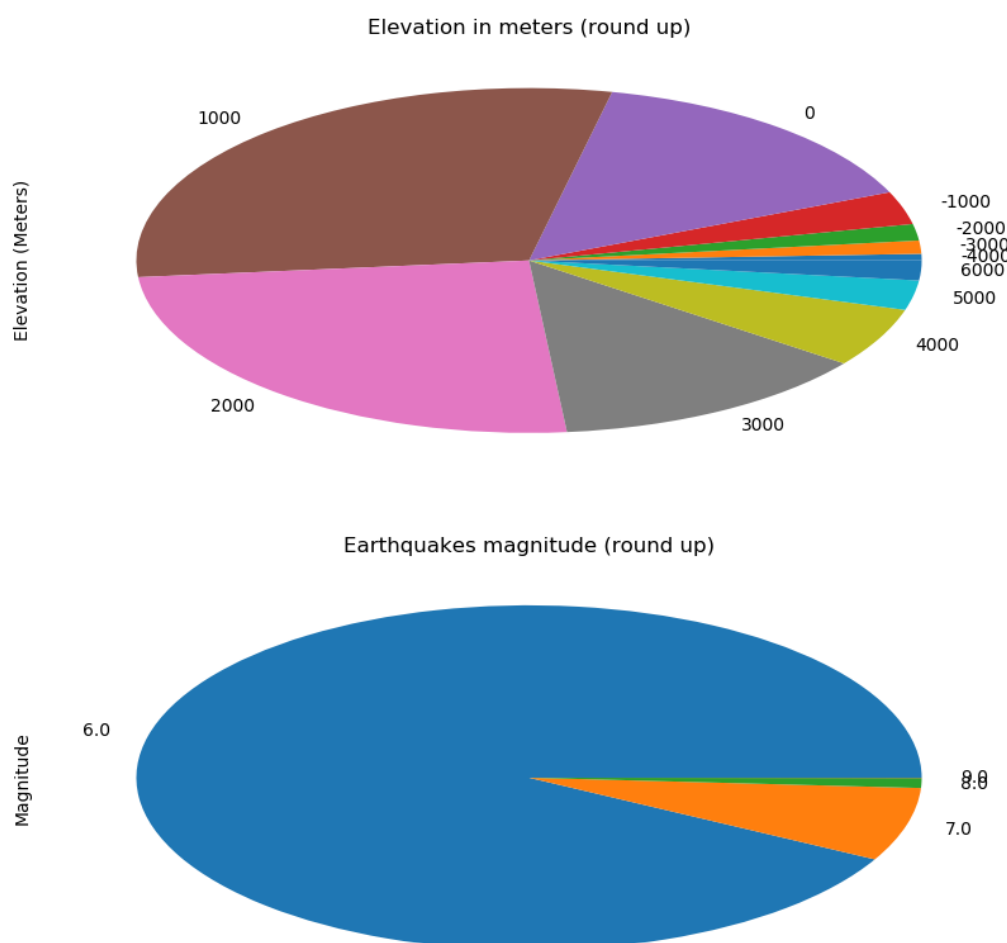


Figure 4: Flächenvisualisierung Vulkangrösse und Erdbeben-Magnitude

## 6.6 Arbeitspaket «Histogramme»

Ziel: Bereitstellung der Histogramme.

### 6.6.1 Analyse

Zwei Histogramme sollen im selben Fenster dargestellt werden. Diese zeigen die Häufigkeit der Ereignisse (einerseits Erbeben, andererseits Vulkanausbrüche) hinsichtlich ihrer geografischen Breite.

### 6.6.2 Quellcode

Der Quellcode befindet sich in der Datei «SEgli\_02\_Latitude\_Histogram.py» im Anhang.



```
from SEgli_02_Latitude_Histogram import *  
show_histograms_by_latitude()
```

### 6.6.3 Resultat

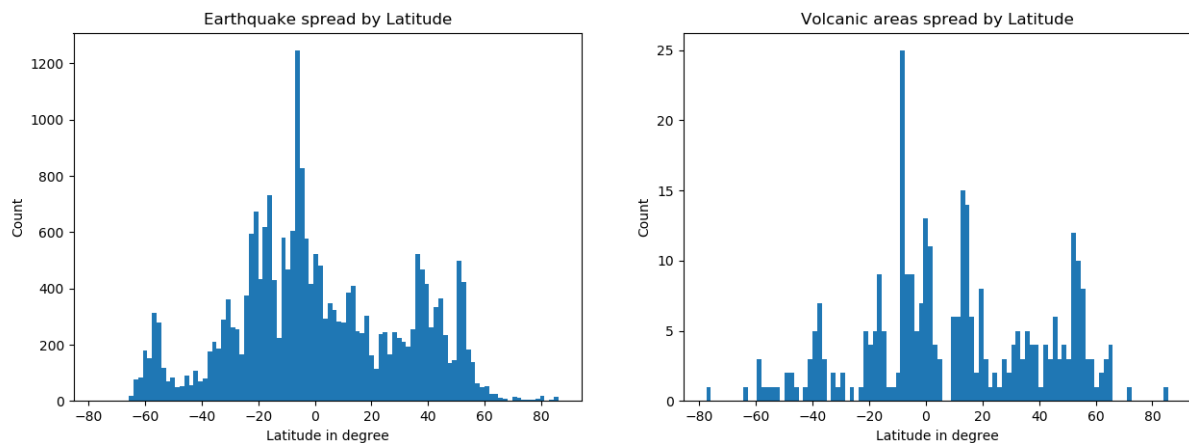


Figure 5: Verteilung über die geografische Breite

## 6.7 Arbeitspaket «Kann-Ziel»

### 6.7.1 Analyse

Die vorangegangenen Histogramme sind spannend. Ereignisse beider Quellen scheinen eine ähnliche Verteilung auf den Breitengraden aufzuweisen.

Im Folgenden soll diese Annahme gestärkt werden, in dem die Ortsangaben beider Datenquellen als Karte dargestellt werden.

### 6.7.2 Quellcode

Der Quellcode befindet sich in der Datei «SEgli\_02\_Map.py» im Anhang.

Zur Ausführung muss das Modul „Basemap“ installiert werden. Zum Zeitpunkt der Erstellung kann das Modul wie folgt installiert werden:

```
conda install -c conda-forge basemap=1.0.8.dev0
```

Anschliessend kann folgender Quellcode ausgeführt werden um die Karte zu generieren:

```
from SEgli_02_Map import *  
print_map()
```

### 6.7.3 Resultat

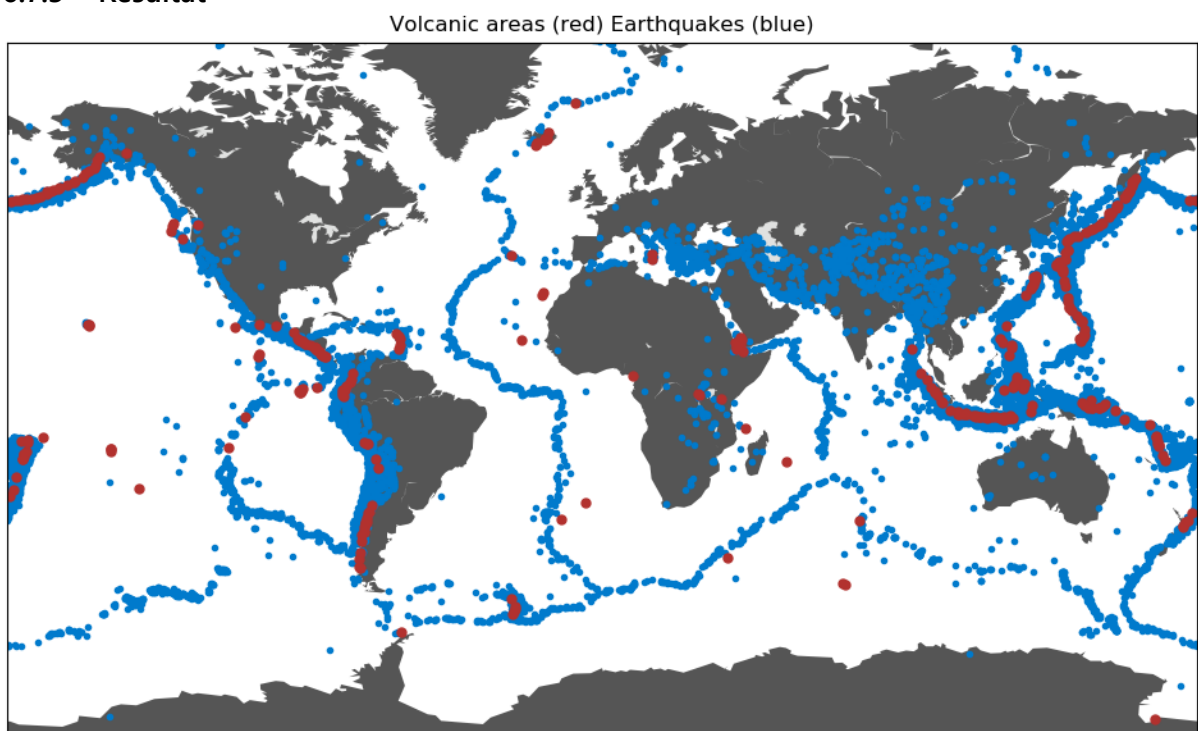


Figure 6: Örtliche Analyse der Datensätze

## 6.8 Auswertung

Die Aufgabenstellung war umfangreich. Zusätzlicher Aufwand war nötig, um die CSV Dateien bereitzustellen. Dieser Mehraufwand war jedoch einkalkuliert. Die korrekte Anwendung der einzelnen Diagramme benötigte einiges an Nachforschung. Schlussendlich konnten die Aufgaben erfolgreich umgesetzt werden.

Aufgrund der fehlenden Erfahrung in der Python Programmierung konnte der zeitliche Rahmen von 8h nicht ganz eingehalten werden.

Der effektive Inhalt dieses Dokumentes umfasst 1403 Wörter, was weniger als drei Seiten Text entspricht. Somit wurde das gestellte Kriterium bezüglich Dokumentationsumfang erfüllt.