Projektbericht

Webmapping

Master-Studiengang:

Geographie – Globaler Wandel, regionale Nachhaltigkeit

Universität Innsbruck

Betreuer:

Klaus Förster

Bernhard Oeggl

VerfasserInnen:

Marlene Benzinger

Marcel Haring

Karolin Vossbeck

Innsbruck, am 16.06.2021

Inhaltsverzeichnis

Abbildungsverzeichnis

Konzeptidee

Für das Innsbrucker Unternehmen coffeekult, hatten wir uns vorgestellt eine globale Karte zu erstellen, um die jeweiligen Kaffeesorten und ihre Anbaugebiete darzustellen. Ziel ist es, eine interaktive Kartenwebsite zu gestalten, die weiterführende Informationen über Kaffee und nachhaltigen Anbau bietet.

Zur Umsetzung brauchen wir die globale Openstreetmap, die wir mit Hilfe der JavaScript library Leaflet darstellen und um Plugins erweitern können, wie etwa einer mini-map um einen Überblick zu bieten. Die Mainpage soll Informationen für die einzelnen Kaffeesorten bereitstellen, hier sind selektierbare Polygone bzw. Popups an den jeweiligen Anbaugebieten geplant. Die Anbaugebiete können über ein Drop-down Menü ausgewählt werden und miteinander verglichen werden. Das passende Plugin müssen wir uns über Leaflet noch suchen.

Zwei weitere HTML Unterseiten sollen interessante Fakten zu Kaffee und nachhaltigen Anbau allgemein bieten und wenn geeignet durch thematische Webmaps aufbereitet werden. Denkbar wäre beispielsweise die Darstellung von Kaffeekopfverbrauch für jedes Land und die Import/Export Wege von Kaffee global gesehen. Für das CSS Design wollten wir uns an der coffeekult Website orientieren.

Umsetzung

Index.html

Zuerst wurde die index.html erstellt und ein Bootstrap framework für CSS und JavaScript verwendet. Das Bootstrap framework stellte die Vorlage für die Navigationsbar im Header und die aufklappbaren Informationsseiten am Seitenrand. Darüber hinaus wurde eine custom.css Datei erstellt, um den Style der Seite zu definieren und weitere .js-Dateien.

```
<!-- BOOTSTRAP CSS -->
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.1/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-+0n0xVW2eSR5OomGNYDnhzAbDsOXxcvSN1TPprVMTNDbiYZCxYbOOl7+AMvyTG2x" crossorigin="anonymous">
<!-- BOOTSTRAP JS -->
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.1/dist/js/bootstrap.bundle.min.js" integrity="sha384-gtEjrD/SeCtmISkJkNUaaKMoLD0//ElJ19smozuHV6z3Iehds+3Ulb9Bn9Plx0x4" crossorigin="anonymous">  </script>
```

Danach wurden die Fremdbibliotheken eingebunden, Google Fonts und Google Icons für Schriftdesign und das Logo im Header der Seite und die Leafletbibliotheken für CSS und JS.

```
<!-- Google Fonts -->
  <link rel="preconnect" href="https://fonts.gstatic.com">  <link href="https://fonts.googleapis.com/css2?family=Lexend+Deca&family=Noto+Sans+TC&display=swap" rel="stylesheet">
<!--Google Icons-->
<link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet">
<!-- Leaflet CSS -->
<link rel="stylesheet" href="https://unpkg.com/leaflet@1.7.1/dist/leaflet.css" integrity="sha512-xodZBNTC5n17Xt2atTPuE1HxjVMSvLVW9ocqUKLsCC5CXdbqCmblAshOMAS6/keqq/sMZMZ19scR4PsZChSR7A==" crossorigin=""/>  <!-- Leaflet JS -->  <script src="https://unpkg.com/leaflet@1.7.1/dist/leaflet.js" integrity="sha512-XQoYMqMTK8LvdxXYG3nZ448hOEQiglfqkJs1NOQV44cWnUrBc8PkAOcXy20w0vlaXaVUearIOBhiXZ5V3ynxwA==" crossorigin=""></script>
```

Als nächstes wurden drei Plugins eingebaut. Das Leaflet Provider Plugin wurde zum Einbinden der Basemap-Layer verwendet, das Leaflet Grayscale Plugin gibt der Karte ihre graue Farbe. Durch das iconLayers Plugin werden die beiden unterschiedlichen Layer als kleine Symbole auf der Karte anzeigt und können dadurch angesteuert werden. Für die Karte wurden geoJson Daten heruntergeladen, um die Polygone der Anbauländer für Kaffee anzuzeigen und später die jeweiligen Produktionsmengen darstellen zu können.

```
<!-- Leaflet Providers Plugin -->

  <script src="https://cdnjs.cloudflare.com/ajax/libs/leaflet-providers/1.12.0/leaflet-providers.min.js"    integrity="sha512-LixflAm9c0/qONbz9F1Ept+QJ6QBpb7wUlPuyv1EHloTVgwSK8j3yMV3elnElGQcv7Y5QTFlF/FqyeE/N4LnKQ=="    crossorigin="anonymous"></script>

<!-- Leaflet Grayscale Plugin -->

  <script src="TileLayer.Grayscale.js"></script>

<!-- iconLayers Plugin -->

    <script src="srcbasemap/iconLayers.js"></script>    <link href="srcbasemap/iconLayers.css" rel="stylesheet">
```

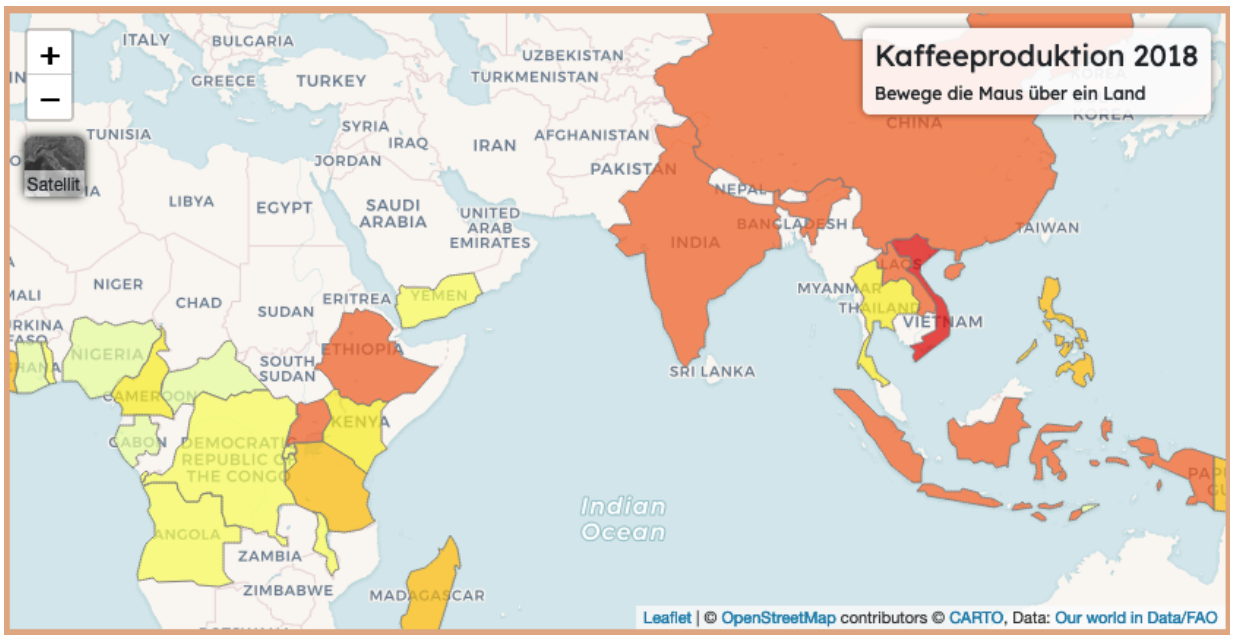

Abbildung 1: Dartellung des verwendeten Plugins mit zwei unterschiedlichen Datensätzen

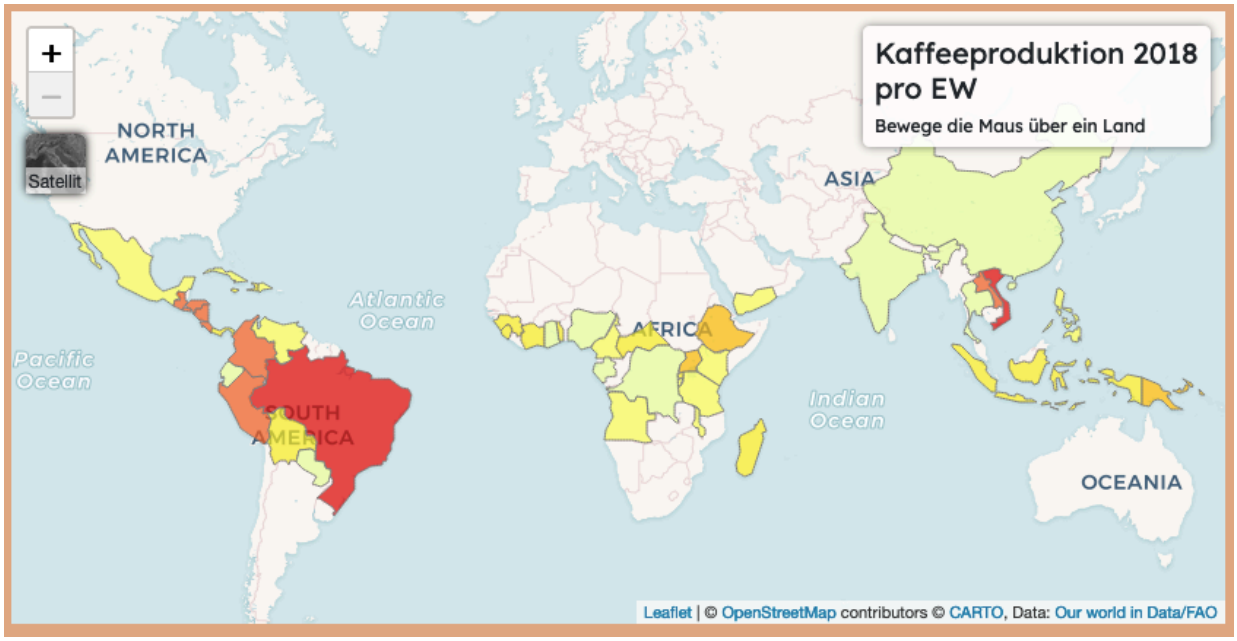

Abbildung 2: Dartstellung des verwendeten Plugins mit unterschiedlichen Datensätzen

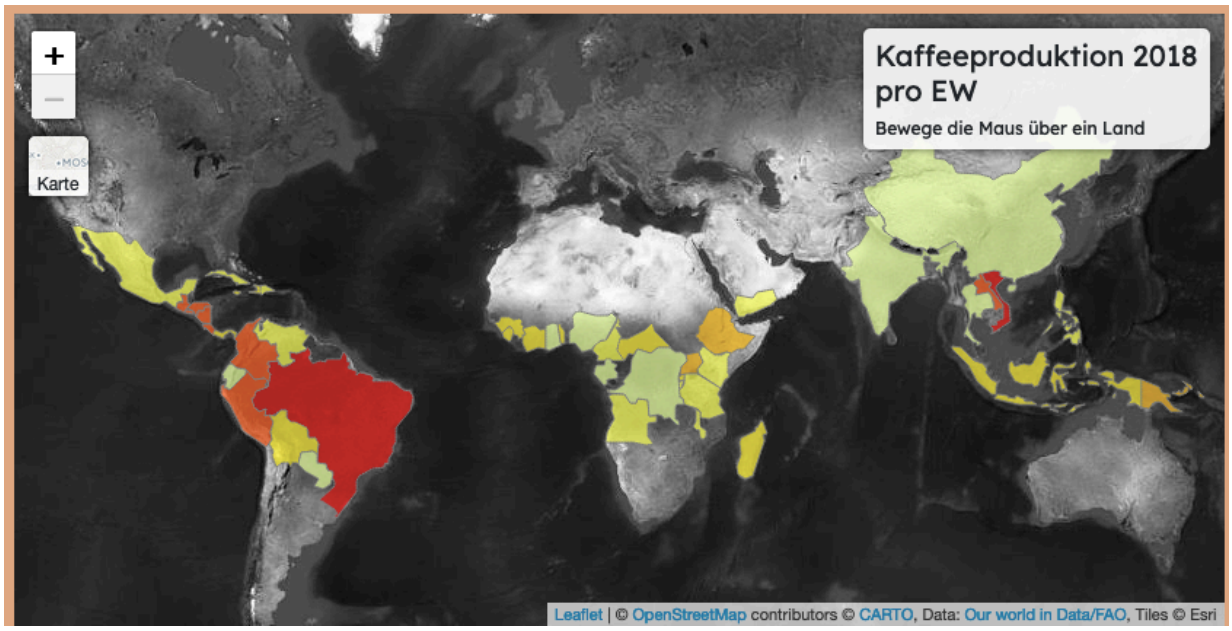

Abbildung 3: Darstellund des aktivierten iconLayer Plugins

Die Karten wurden für die Weiterarbeit in der .js-Datei in einem div-Container mit einer ID gekennzeichnet.

```
<!-- Die Karten -->
<div id="mapproduction"></div>
<div id="mapproductionperson"></div>          </div>
```

Export.html

Die export hmtl Seite nutzt das gleiche Gerüst wie schon in der index.html Seite definiert: Das Leaflet Provider Plugin wurde zum Einbinden der Basemap-Layer verwendet, das Leaflet Grayscale Plugin gibt der Karte ihre graue Farbe. Durch das iconLayers Plugin werden die beiden unterschiedlichen Layer als kleine Symbole auf der Karte anzeigt und können dadurch angesteuert werden. Zusätzlich wurde noch das Fullscreen Plugin hinzugefügt.

Um die Exportwege mit Pfeilen darzustellen, wurde das Leaflet-Plugin „Arrowheads“ verwendet. Das Leaflet-Plugin „TextPath“ beschriftet die Polylines.

```
<!-- export JS zur Darstellung der Pfeile -->
  <script defer src="export.js"></script>
<!-- Arrowheads -->
  <script src="https://cdn.jsdelivr.net/npm/leaflet-arrowheads@1.2.3/src/leaflet-arrowheads.js"></script>
  <!-- geometry util -->
<script src="https://cdn.jsdelivr.net/npm/leaflet-arrowheads@1.2.3/src/leaflet-geometryutil.js"></script>
    <!-- Text Path -->
<script src="https://cdn.jsdelivr.net/npm/leaflet-textpath@1.2.3/leaflet.textpath.js"></script>
</head>
  <!-- Leaflet fullscreen -->
  <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/leaflet.fullscreen/2.0.0/Control.FullScreen.css"
    integrity="sha512-KQk/GTCcAywe4iEOhnZ6ZmWDNv/3NvOE6f0iUvrNVCX/oD7+sPXAeDP90aOX3EXHmerC6gDYUlUBaSd4hxjWEw=="
    crossorigin="anonymous" />
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/leaflet.fullscreen/2.0.0/Control.FullScreen.js"
    integrity="sha512-TaNrKSd5TOm4PfgJMFYkDpQ1X8GmqXBlh+Kmk83Mfrcx9sjTzqI1zcZd0n0xpdJevfPL7voAcbblrUaYx+LM1g=="
    crossorigin="anonymous"></script>
```

Um die Kartendarstellung in export.html einzubinden, wurde die eigene export.js-Datei erstellt, mit einer ID gekennzeichnet und so verknüpft:

<!-- Die Map für Export Darstellungen -->

* *

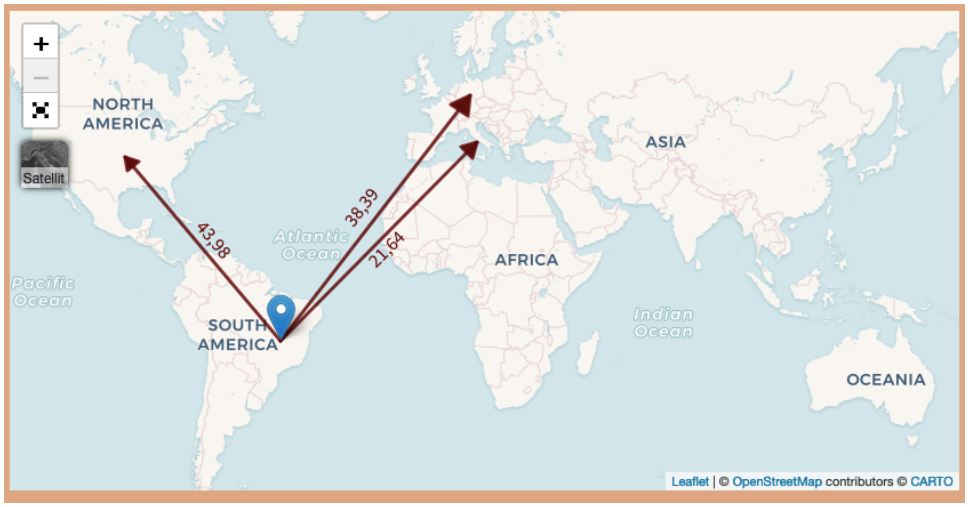

Abbildung 4: Darstellung des verwendeten Arrow Plugins

Export.js

Das Fullscreen-Option wurde der makemap-Funktion hinzugefügt.

Die Koordinaten der Länder wurden in einer Liste mit der Bezeichnung *latlngs* gespeichert, um damit die Exportwege von Brasilien zu den drei Hauptabnehmern darzustellen.

```
// create polylines from an array of arrays of LatLng points
// brazil to ...
var latlngs = [
[  //usa
[-15.749997, -47.9499962],        [37.6, -95.665]    ],
[ //germany
[-15.749997, -47.9499962],        [51.1642292, 10.4541194]    ],
[ //Italy
[-15.749997, -47.9499962],        [41.29246, 12.5736108]    ]];
```

Für Brasilien, das Ausgangsland, wurde ein Marker hinzugefügt.

```
var marker = L.marker(    [-15.749997, -47.9499962])
.addTo(expomap);
```

Die Exportwege in die drei Länder wurde mit Polylines (.polyline) dargestellt, dies sollten jedoch Pfeile verdeutlichen. Dafür wurde das Leaflet-Plugin „Arrowheads“ verwendet (.arrowheads).

Um die Menge des Exports darzustellen, wurde das Leaflet-Plugin „TextPath“ eingebunden. So konnten die Polylines mit den jeweiligen Zahlen beschriftet werden. Die Polylines mit den Pfeilspitzen, wie auch die Beschriftung (setText), wurden dann an das Design der Website mit jeweiligen Optionen angepasst.

```
// Brazil to USA
var usaline = L.polyline(latlngs[0], {
color: '#5C0000',
opacity: 0.8    }).
arrowheads({       size: '5%',       proportionalToTotal: true,       fill: true,    })
.addTo(expomap);

// adjust Text
usaline.setText('43,98', {
   repeat: false,
    offset: 20,
   attributes: {
        fill: '#5C0000',
        'font-weight': 'light',
        'font-size': '15',
       'font-family': 'Noto Sans TC',
   },
   center: true,
orientation: 180,
below: true,});
```

Verbrauch.html

Die verbrauch.html ist wie die index.html aufgebaut, besitzt aber keine plugins und dient zur Information.

Main.js

Im main.js wurde zunächst die Karte dargestellt und die Layer Control definiert. Für die Karten der globalen Kaffeeproduktion wurde im main.js noch die Choropletenfarbe bestimmt, ein Hovereffekt eingefügt und die geoJson Daten hinterlegt.

```
// Karte darstellen
function makemap(mapid) {
let map = L.map(mapid, {
      center: [12, 15],
```

```
zoom: 2,
   layers: L.tileLayer.provider('CartoDB.Voyager'),
       maxZoom: 6,
minZoom: 2,        fadeAnimation: false
    });
    return map
  }
```

Durch die Funktion makemap (mapid) kann die zweite Karte zu Produktion pro Kopf und später die Karte in der export.js über eine id-Klassifizierung den Code übernehmen.

```
// Karte Produktion total mit Funktion erzeugen
let mapproduction = makemap("mapproduction");
makeLayerControl (mapproduction);

// Choropletenfarbe bestimmen
function getProductionColor(prod) { //Funktion mit JS ternary operator (wie if eine conditional Abfrage)
return prod > 1000000 ? '#DF0000' :
      prod > 100000 ? '#FA5D20' :
      prod > 50000 ? '#FABB00' :
   prod > 30000 ? '#FAEB20' :
prod > 10000 ? '#FAFF56' :
'#E8FF9C';
};
function productionStyle(feature) {
return { //Style Funktion
      fillColor: getProductionColor(feature.properties.coffe_production),
      weight: 1,
     opacity: 1,
     color: 'grey',
dashArray: '1',
      fillOpacity: 0.7,
```

```
attribution: "Data: <a href=\"https://ourworldindata.org/grapher/coffee-bean-production\">Our world in Data/FAO</a>"
    };
// geoJsonDaten hinterlegen
selectedMap = L.geoJson(coffeeProducerCountries, {
style: productionStyle;
 onEachFeature: onEachFeature
}).addTo(mapproduction);
```

Um dem Hovereffekt noch eine Infobox hinzufügen, die den jeweiligen Produktions/Exportwert bei Mausbewegung über das Landpolygon anzeigt, wurde die Funktion infoPop verwendet.

```
// Wert anzeigen bei Hover
let infoPop = L.control();
infoPop.onAdd = function () {
  this._div = L.DomUtil.create('class', 'infobox'); // create a div with a class
"info"
  this.update();
 return this._div;
};
infoPop.update = function (feature) { // method that we will use to update the control based on feature properties passed
this._div.innerHTML = '<h5>Kaffeeproduktion 2018 <br>pro EW</h5>' +  (feature ? // Conditional ob Hover
        '<b>' + feature.formal_de + '</b><br />' + Math.round((feature.coffe_production/feature.pop_est)*10000)/10 + ' Kilo/Einwohner'
: 'Bewege die Maus über ein Land');
};
infoPop.addTo(mapproductionperson);
```

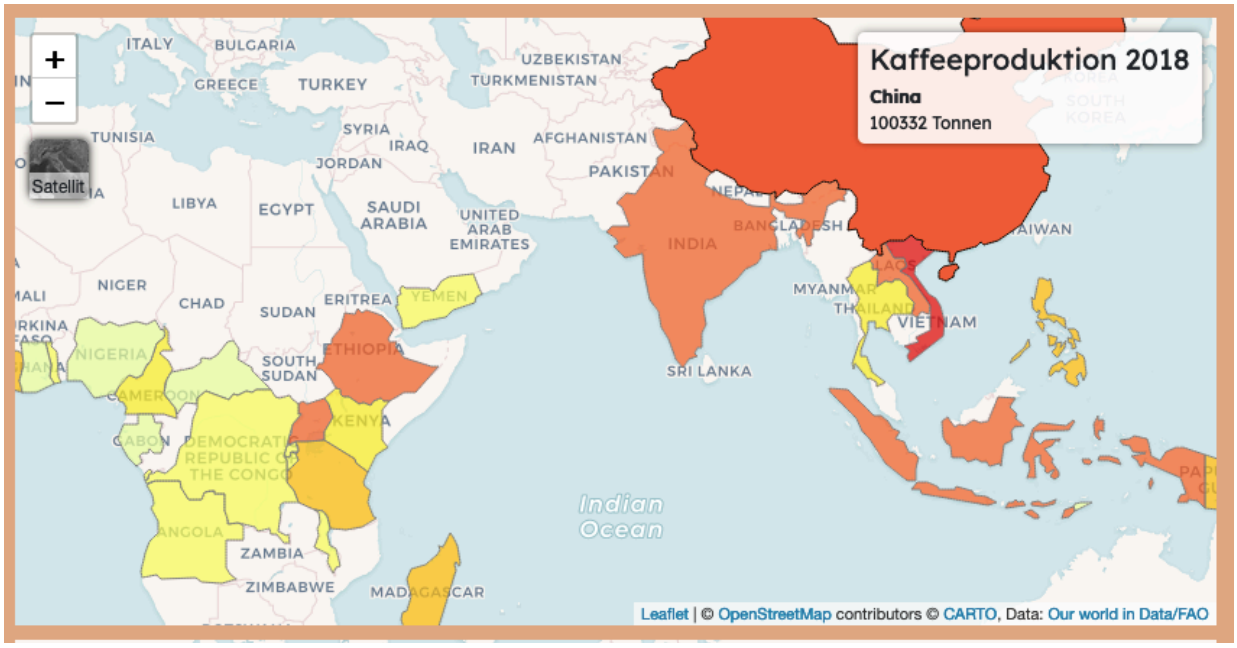

Abbildung 5: Darstellung der infoPop Funktion, hier die Maus auf China, Wert angezeigt in der Box oben rechts.

Coffee_countries.js

In dieser JavaScript-Datei wurden die Variablen *coffeeProducerCountries* mit den Informationen der jeweiligen Länder gespeichert, um dann in der main.js auf die jeweiligen features, vor allem die Polygone und Namen der Länder, zurückgreifen zu können.

TileLayerGrayscale.js

Für das Leaflet Grayscale Plugin wurde eine eigene .js-Datei erstellt. Es handelt sich um ein reguläres TileLayer mit einem Graysale makeover, sodass die Karte in Grautönen erscheint, was die Choroplethenfarben besser darstellen lässt.

Quellen

Bunn, Christian & Laderach, Peter & Ovalle Rivera, Oriana & Kirschke, Dieter. (2014). A bitter cup: climate change profile of global production of Arabica and Robusta coffee. Climatic Change. 129. 10.1007/s10584-014-1306-x.

Global Coffee Data: http://datastandard.globalcoffeeplatform.org/en/latest/explanation.html (15.6.2021)

https://www.indexmundi.com/agriculture/?commodity=green-coffee&graph=bean-exports&display=map (15.6.2021)

https://www.thecoffeeguide.org/coffee-guide/world-coffee-trade/conversions-and-statistics/ (15.6.2021)

https://worldcoffeeresearch.org/ (15.6.2021)

geoJSON countries: https://geojson-maps.ash.ms/ (15.6.2021)

Anbaugebiete Top 10: https://www.coffeecircle.at/de/b/kaffee-anbaugebiete (15.6.2021)

Plug-ins

leaflet-providers: https://cdnjs.com/libraries/leaflet-providers

Leaflet-arrowheads: https://www.jsdelivr.com/package/npm/leaflet-arrowheads

leaflet-grayscaleoverlay: https://www.jsdelivr.com/package/npm/leaflet-grayscaleoverlay

leaflet-iconlayers: https://www.jsdelivr.com/package/npm/leaflet-iconlayers

leaflet-textpath: https://www.jsdelivr.com/package/npm/leaflet-textpath

leaflet-fullscreen: https://cdnjs.com/libraries/leaflet.fullscreen