What can we learn from the visualization?

At what longitude the settlements with the longest period of longevity are located. It is also possible to see the average period of the selected settlements with a variable being displayed at the top left of the plot. This shows that the settlements with the longest periods are located in Turkey and the Middle East, while settlements with shorter periods dominate Western Europe.

What is the name for the type of visualization(s) used?

Scatter Plot.

What are all visual mappings used?

x position:

- Longitude Category

y position:

- Time Range Category (maxDate - minDate)

colour:

- Time Periods Keys Category

tooltip:

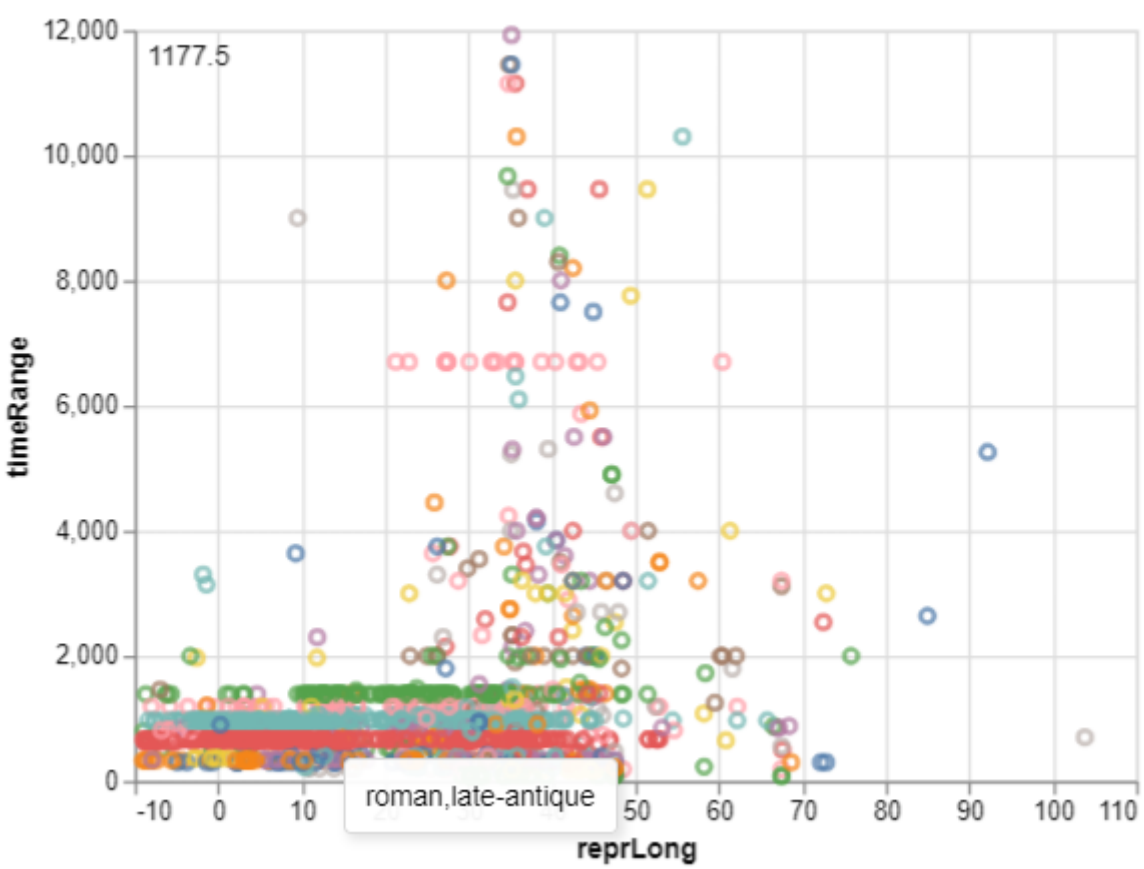
- Time Periods Keys Category

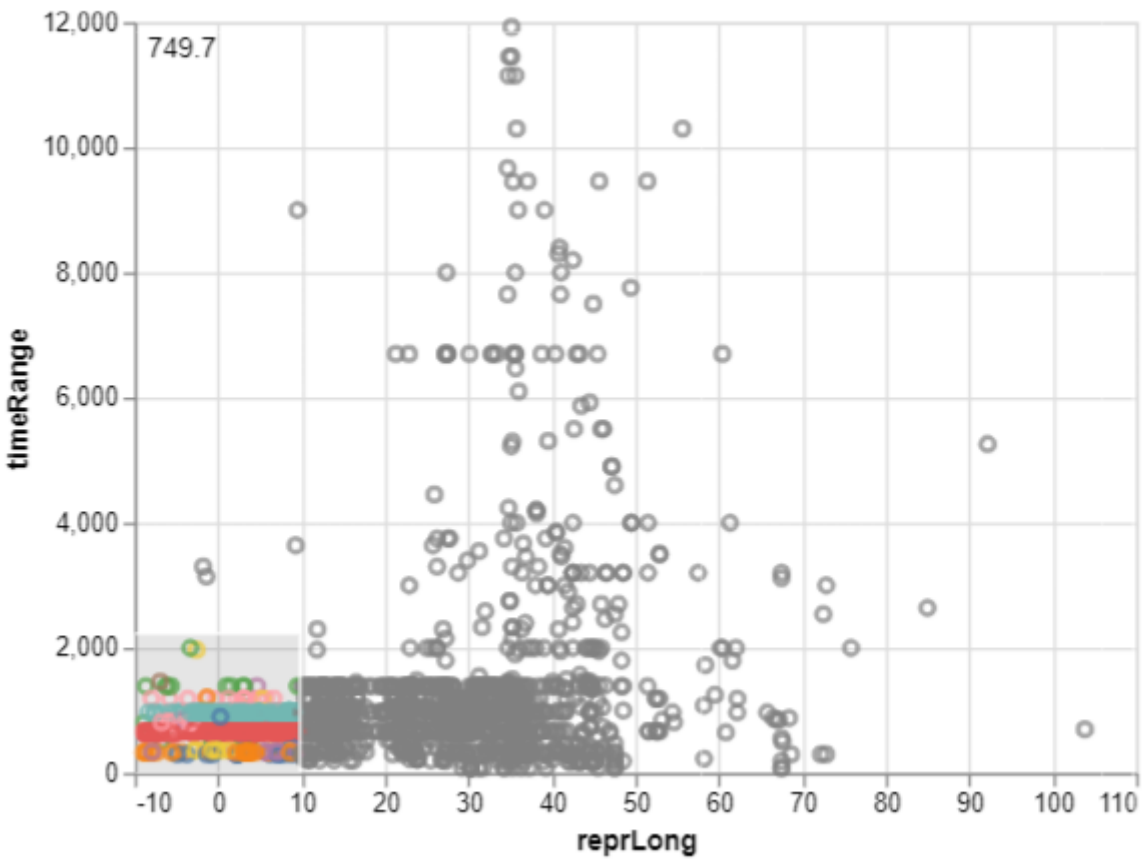
Was there any special data preparation done?

A new column was added, named timeRange, which is the subtraction of maxDate and minDate. Any dataset that wasn't a settlement was dropped. Finally, any settlement that was older than 60000 years ago (which represents the beginning of human civilization) was also dropped.

What are the limitations of your design?

the scatter plot only plots the longitude to timeRange, and there is no information provided on the latitude.


1177.5
12,000
10,000
8,000
6,000
4,000
2,000
0
timeRange
-10
0
10
50
60
70
80
90
100
110
reprLong
roman,late-antique


749.7
12,000
10,000
8,000
6,000
4,000
2,000
0
timeRange
-10
0
10
20
30
40
50
60
70
80
90
100
110
reprLong

```
import IOReader as io
import altair as alt
from vega_datasets import data

#Opening the dataset

alt.renderers.enable('altair_viewer')
alt.data_transformers.disable_max_rows()
locations_data = io.open_file("pleiades-locations-latest.csv")

#dropping unnecessary data

locations_data['timeRange'] = locations_data['maxDate'] -
locations_data['minDate']
locations_data.drop(
    locations_data[locations_data['timeRange'] > 60000].index, inplace=True)
locations_data.drop(locations_data[~(
    locations_data['featureType'] == "settlement")].index, inplace=True)

#selection for interaction

sel = alt.selection_interval()

#chart specification

chart = alt.Chart(locations_data).mark_point().encode(
    x='reprLong:Q',
    y='timeRange',
    color=alt.condition(sel, alt.Color('timePeriodsKeys:N', legend=None),
alt.value('gray')),
    tooltip='timePeriodsKeys'
).add_selection(
        sel
)

#average calculation

text = alt.Chart(locations_data).transform_filter(sel).      mark_text(
    align='left',
    baseline='top',
).encode(
    x=alt.value(5),
    y=alt.value(5),
    text=alt.Text('average(timeRange):Q', format='.1f'),
)

#show in Altair Viewer

(chart + text).show()
```

What can we learn from the visualization?

The aim of this visualization is to see the development of settlements over time. We can see that the number and variety of settlements increases over time. This shows the progress of human civilization over time.

What is the name for the type of visualization(s) used?

Geomap with samples located with latitude and longitude.

What are all visual mappings used?

x position:

- Longitude Category

y position:

- Latitude Category

colour:

- Time Periods Keys Category

tooltip:

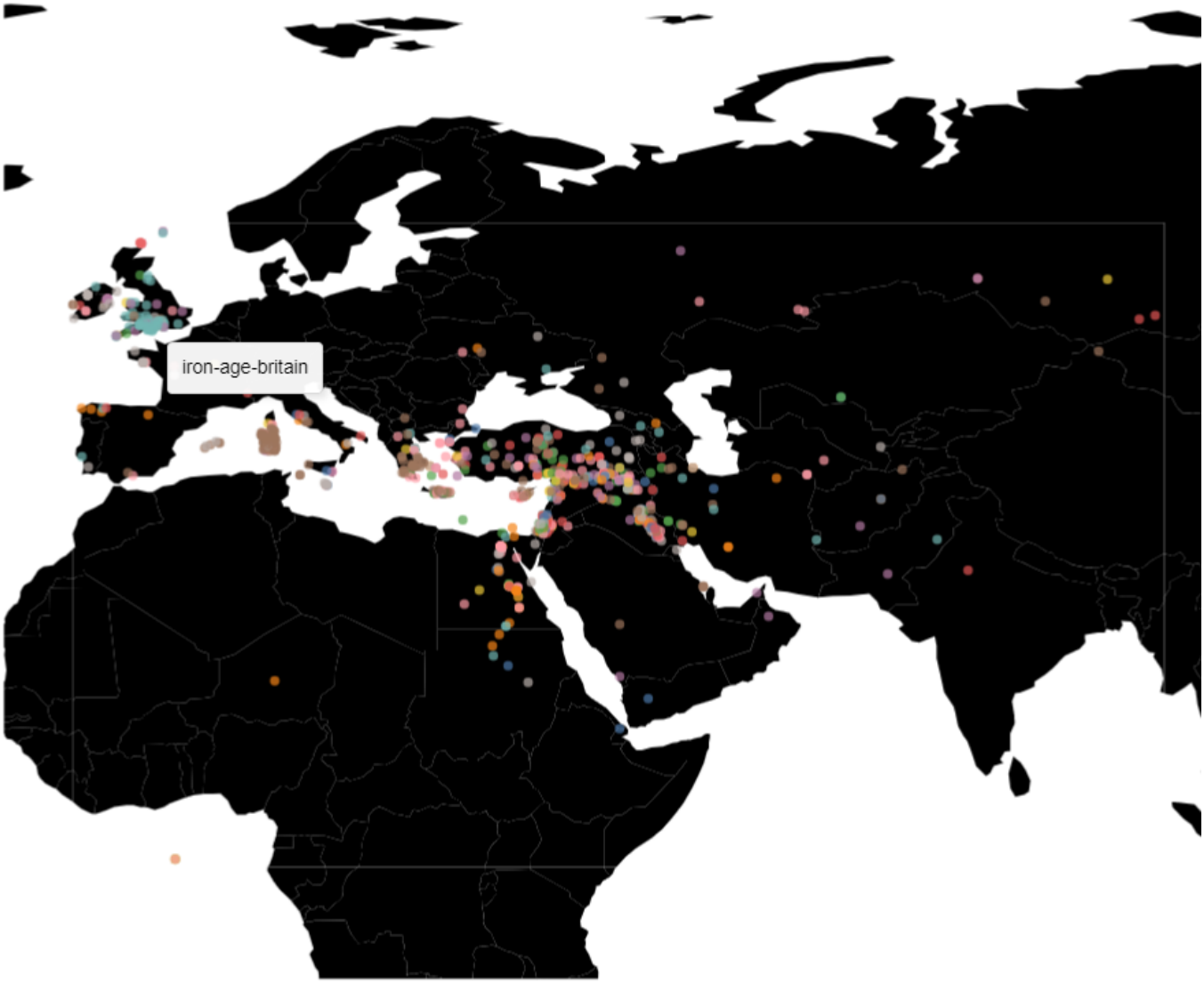
- Time Periods Keys Category

Was there any special data preparation done?

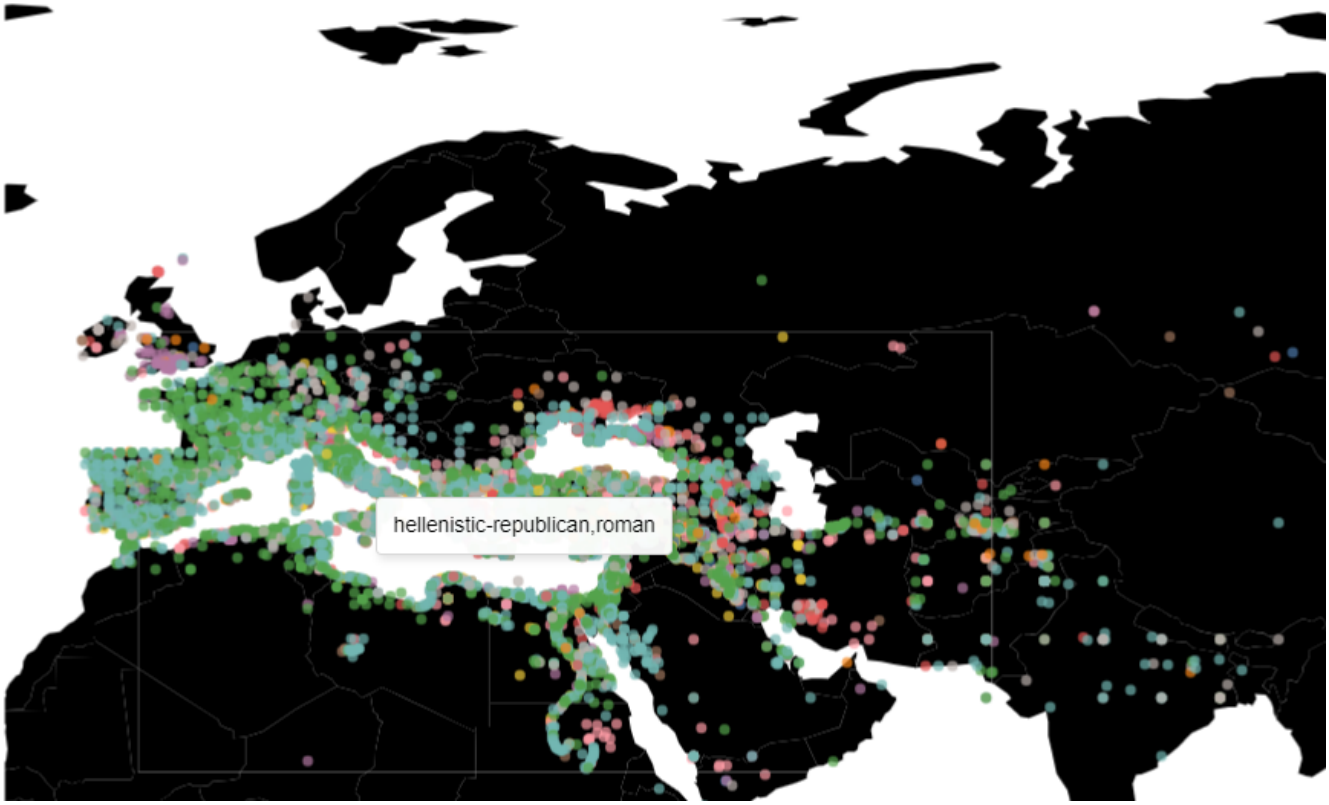
A new column was added, named timeRange, which is the subtraction of maxDate and minDate. Any settlement located above the equator was dropped. Finally, any settlement that was older than 60000 years ago (which represents the beginning of human civilization) was also dropped.

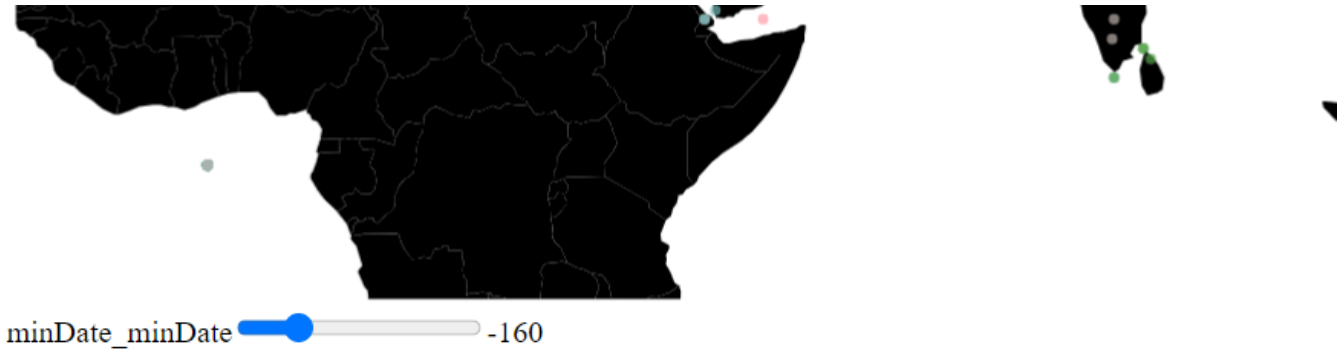
What are the limitations of your design?

The data that is linked to the slider is not continuous, so there are steps in which nothing changes, and steps where the information on the screen changes dramatically. Furthermore, the density of samples allows samples to cover each other, decreasing the amount of information that can be shown effectively.

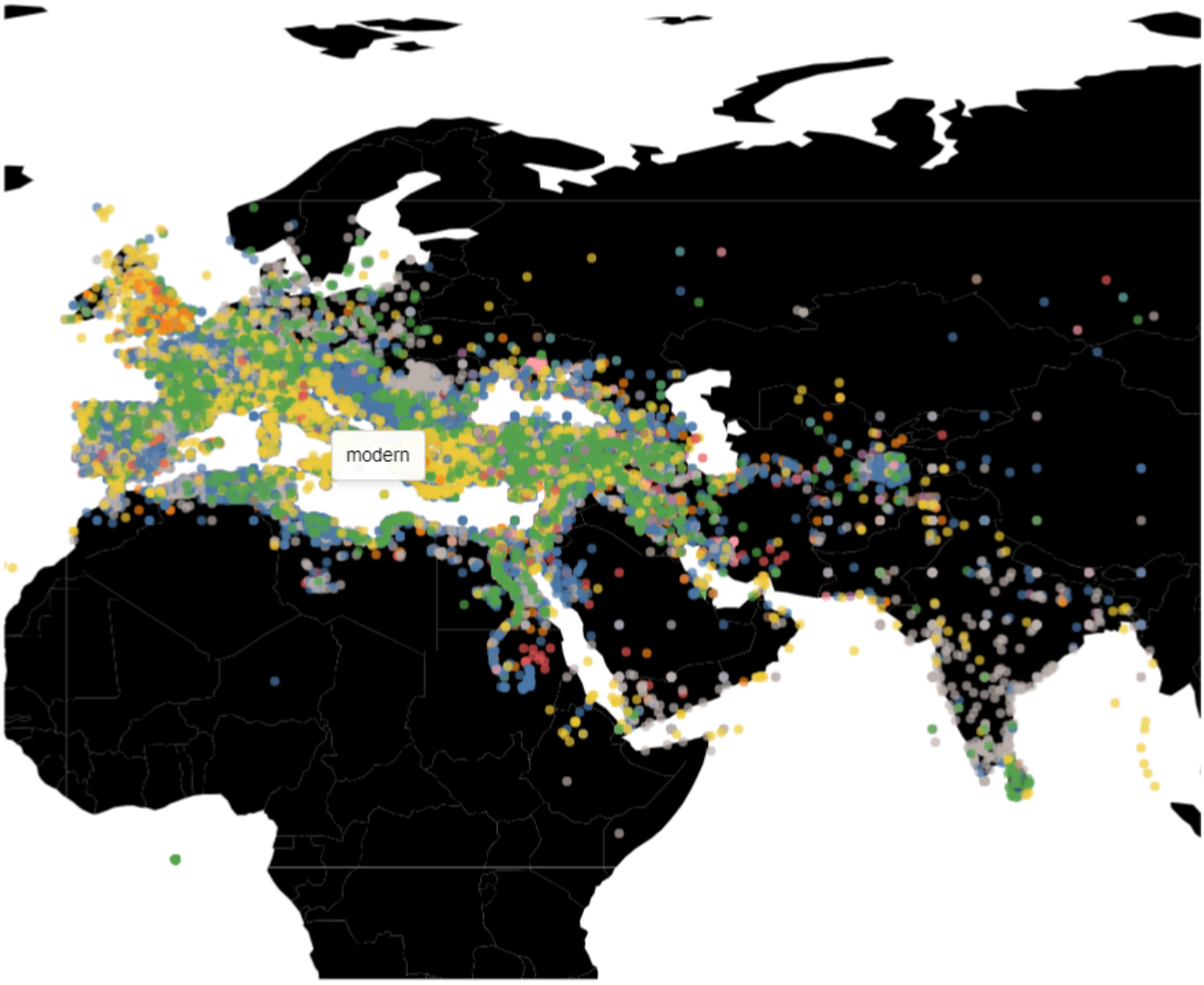


minDate_minDate -750





minDate_minDate -160



minDate_minDate 2000

```
import IOReader as io
import altair as alt
from vega_datasets import data

alt.renderers.enable('altair_viewer')
alt.data_transformers.disable_max_rows()
dataset = io.open_file("pleiades-locations-latest.csv")

countries = alt.topo_feature(data.world_110m.url, 'countries')

#preprocessing
```

```
dataset['timeRange'] = dataset['maxDate'] - dataset['minDate']
dataset.drop(
    dataset[dataset['timeRange'] > 60000].index, inplace=True)
dataset.drop(
    dataset[dataset['reprLat'] < 0].index, inplace=True)


#width and height of window
widthValue = 700
heightValue = 700

#slider
minDateSlider = alt.binding_range(min=-750, max=2000, step=10)
minDateSel = alt.selection_single(name="minDate", fields=['minDate'],
                                  bind=minDateSlider, init={'minDate': -750})

#chart specification
chart = alt.Chart(dataset).mark_circle().encode(
    longitude='reprLong',
    latitude='reprLat',
    color=alt.Color('timePeriodsKeys:N', legend=None),
    tooltip = 'timePeriodsKeys'
).properties(
    width=widthValue,
    height=heightValue,
).add_selection(
    minDateSel
).transform_filter(
    alt.datum.minDate < minDateSel.minDate
)

#geomap parameters
scaleValue = 350
translation = [100, 630]

#geomap specification
mapData = alt.layer(
    alt.Chart(countries).mark_geoshape(fill='black'),
).project(
    type='equirectangular', scale=scaleValue, translate=translation
).properties(width=widthValue, height=heightValue).configure_view(stroke=None)

(mapData + chart).show()
```

What can we learn from the visualization?

The aim of this visualization is to show the relationship between settlements of different time periods, in terms of their location, surface area, dates.We can see that most settlements have a similar surface area, and there is relatively little variation in terms of location. the periods in which each settlement lived changes dramatically.

What is the name for the type of visualization(s) used?

Parallel Coordinates.

What are all visual mappings used?

y position:

- Average values across different time periods.

x position:

- maxDate Category
- minDate Category
- scaledArea Category
- scaledReprLat Category
- scaledReprLong Category

colour:

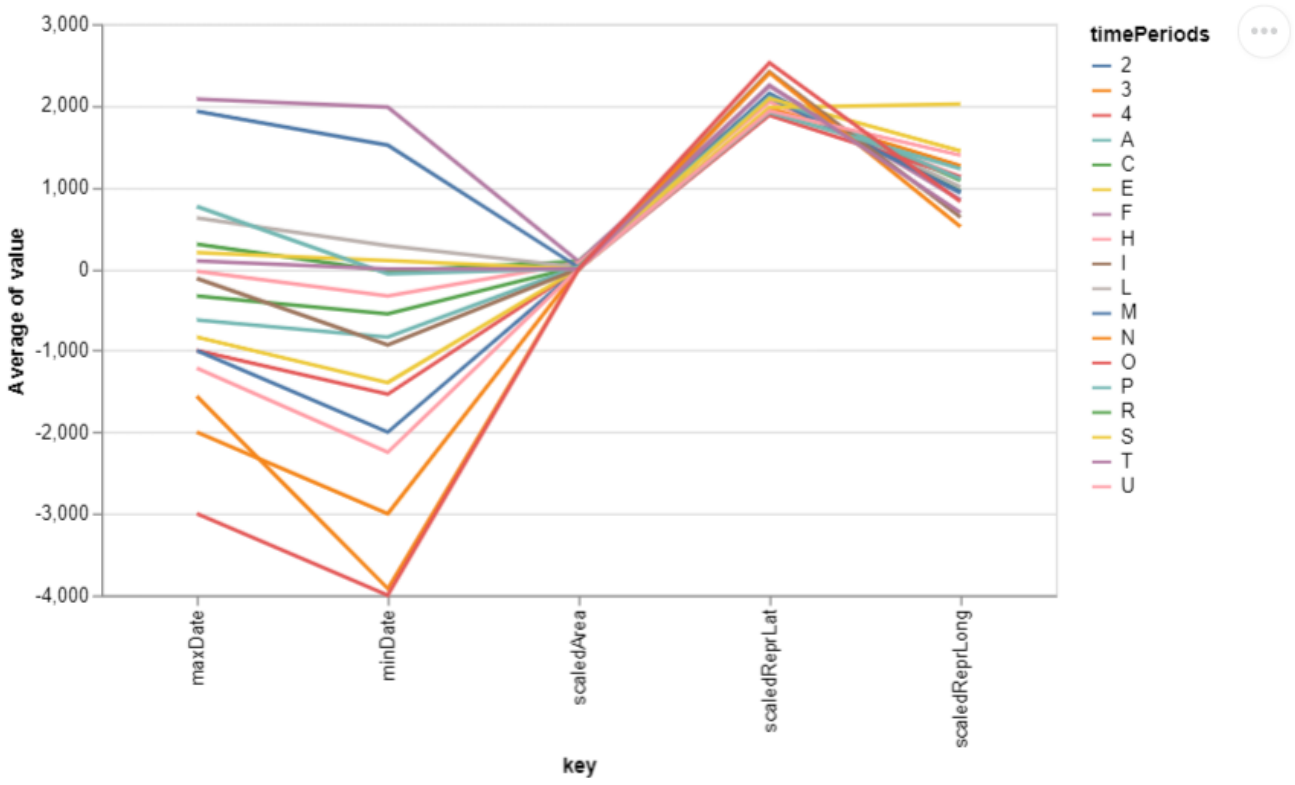
- Time Periods Category

Was there any special data preparation done?

A new column was added, named timeRange, which is the subtraction of maxDate and minDate. Any settlement that was older than 60000 years ago (which represents the beginning of human civilization) was dropped. Any settlement located above the equator was dropped. Furthermore, 4 more columns were added by separating the contents of the bbox column. This was done on Excel by following the tutorial at https://www.youtube.com/watch?v=QKM7q4fHYOU . These 4 columns were then used to calculate the surface area of each settlement. Settlements with a surface area of zero were then dropped. Moreover, the areas and coordinates of each settlement were rescaled so that the variation of those values would be more visible on the visualization. The amount of data on the screen was decreased to make the visualization cleaner and easier to read. This was done by removing all samples with a timePeriods value that had more than one character. This means we only retain settlements that are clearly identified by one time period. Finally, the y axis of the visualization is the average values of all settlements in the same time period.

What are the limitations of your design?

The relationships between axis that are visible greatly depend on the order in which the axis are presented. If the user were able to change the order of the axis, more interesting patterns could appear. Furthermore, because the values for area and coordinates needed to be rescaled, it's hard to relate them to their original values. Finally, to make the visualization cleaner, a lot of data needed to be dropped, which may have revealed other patterns in the dataset.




```
#excel reference at: https://www.youtube.com/watch?v=QKM7q4fHYOU

import IOReader as io
import pandas as pd
import altair as alt
from vega_datasets import data

alt.renderers.enable('altair_viewer')
alt.data_transformers.disable_max_rows()
df = pd.read_csv("pleiades-locations-latest.csv")

#the following values are approximations and they may vary
#depending on the position on the planet.
distanceBetweenLatitudes = 111000
distanceBetweenLongitudes = 97000

df['timeRange'] = df['maxDate'] - df['minDate']
df.drop(
    df[df['timeRange'] > 60000].index, inplace=True)

df['area'] = (df['topRightX'] - df['bottomLeftX']) * (df['topRightY'] -
df['bottomLeftY'])

#rescaling area
maxArea = df['area'].max()
minArea = df['area'].min()

originalRange = maxArea - minArea
```

```
scaledMax = 3000
scaledMin = 0

scaledRange = scaledMax - scaledMin

df['scaledArea'] = (((df['area'] - minArea) * scaledRange) / originalRange) +
scaledMin

#rescaling reprLat
maxReprLat = df['reprLat'].max()
minReprLat = df['reprLat'].min()

originalRange = maxReprLat - minReprLat

scaledMax = 3000
scaledMin = 0

scaledRange = scaledMax - scaledMin

df['scaledReprLat'] = (((df['reprLat'] - minReprLat) * scaledRange) /
originalRange) + scaledMin

#rescaling reprLong

maxReprLong = df['reprLong'].max()
minReprLong = df['reprLong'].min()

originalRange = maxReprLong - minReprLong

scaledMax = 3000
scaledMin = 0

scaledRange = scaledMax - scaledMin

df['scaledReprLong'] = (((df['reprLong'] - minReprLong) * scaledRange) /
originalRange) + scaledMin

#dropping unnecessary data

df.drop(df[df['scaledArea'] == 0].index, inplace=True)
df = df[df['timePeriodsKeys'].str.split(",").str.len().lt(2)]

#chart specification
chart = alt.Chart(df).transform_window(
    index='count()'
).transform_fold(
    ['scaledArea', 'maxDate', 'minDate', 'scaledReprLat', 'scaledReprLong']
).mark_line().encode(
    x='key:N',
    y='average(value):Q',
    color='timePeriods'
).properties(width=500)
```

```
chart.show()
```

What can we learn from the visualization?

The aim of this visualization is to see patterns of settlements from different time periods in Tuscany, Italy. We can see that roman settlements have had a lot of influence in the region. Moreover, modern settlements are mostly located in the eastern part of Tuscany. Finally, we can see that the dataset does not identify Etrurian settlements as Etrurian. Places like "Belvedere Tomb" and "Tomba della Montagnola" are identified under the umbrella term of "archaic" instead.

What is the name for the type of visualization(s) used?

Geomap with samples located with latitude and longitude.

What are all visual mappings used?

x position:

- Longitude Category.

y position:

- Latitude Category.

colour:

- Time Periods Keys Category.

tooltip:

- Title Category.

Was there any special data preparation done?

all settlements that weren't near the region of Tuscany were dropped. Furthermore, settlements that aren't clearly identified by one time period were dropped.

What are the limitations of your design?

Not all samples in the visualization are in Tuscany, as it was not possible to only select settlements in that region. For example, "Galeata" (which is visible in the visualization) is not in Tuscany, but it is in the neighbouring region of Emilia Romagna.


```
#Siena UID = 62adf57abfdf2645abe652f7caa19880
#Siena ID = 413293
#Siena connectsWith = 702,604,147,136,661,000
#Siena coordinates (lat, long) = 43.318695,11.330502

import pandas as pd
import altair as alt
import numpy as np
from vega_datasets import data

#loading data
alt.renderers.enable('altair_viewer')
alt.data_transformers.disable_max_rows()
df = pd.read_csv("pleiades-places-latest.csv")

#geomap
countries = alt.topo_feature(data.world_110m.url, 'countries')

#dropping samples outside of Tuscany
df.drop(df[df['reprLat'] < 42].index, inplace=True)
df.drop(df[df['reprLat'] > 44].index, inplace=True)

df.drop(df[df['reprLong'] < 10].index, inplace=True)
df.drop(df[df['reprLong'] > 12].index, inplace=True)
```

```
#dropping samples that aren't clearly identified by one time period
df = df[df['timePeriodsKeys'].str.split(",").str.len().lt(2)]

#window height and width
widthValue = 700
heightValue = 700

#chart specifications
chart = alt.Chart(df).mark_circle().encode(
    longitude='reprLong',
    latitude='reprLat',
    color='timePeriodsKeys',
    tooltip='title'
)

#geomap parameters
scaleValue = 5300
translation = [-700, 4300]

#geomap specification
mapData = alt.layer(
    alt.Chart(countries).mark_geoshape(fill='black'),
).project(
    type='equirectangular', scale=scaleValue, translate=translation
).properties(width=widthValue, height=heightValue).configure_view(stroke=None)

(mapData + chart).show()
```

What can we learn from the visualization?

The aim of this visualization is to show a possible correlation between the longevity of a settlement and its vicinity to a volcano. Volcanos are shown in red, while the colour of the settlements represents their longeivity. It seems that there isn't an obvious pattern appearing in the visualization. In the caucasian region, there is a high density of volcanos and a low density of settlements, which might suggest a possible correlation between the two. However, this might also be due to the region's unforgiving climate. Furthermore, there is a few settlements located near volcanos that have existed for a long period of time. This might suggest that the settlement took advantage of fertile lands around the volcano, or it might just indicate that the volcano was not active.

What is the name for the type of visualization(s) used?

Scatter plots placed on top of a geomap.

What are all visual mappings used?

x position:

- Longitude Category.

y position:

- Latitude Category.

colour:

- Time Range Category (for settlements).
- red (for volcanos)

tooltip:

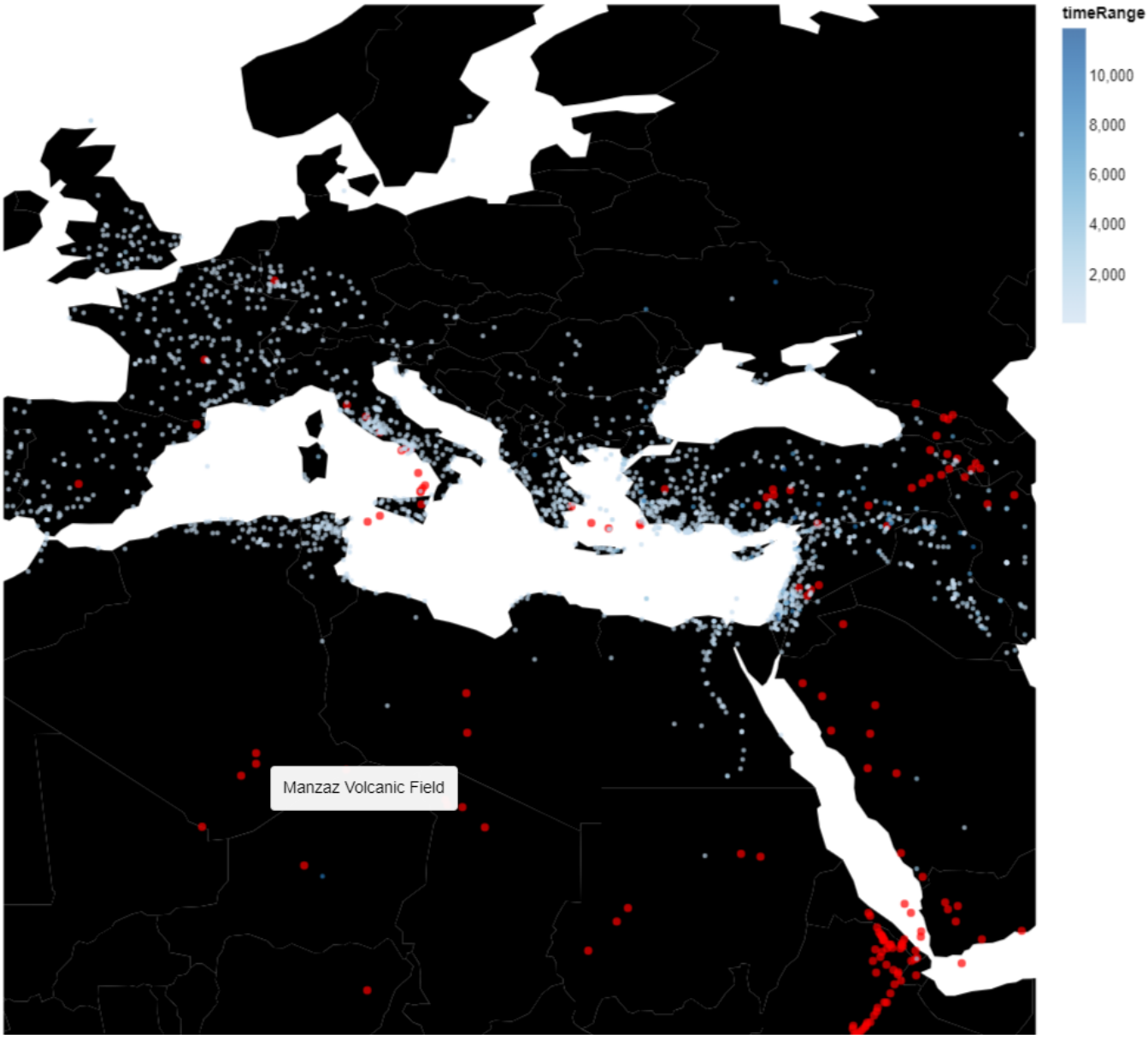
- Time Periods Keys Category (for settlements).
- V_Name Category (for volcanos).

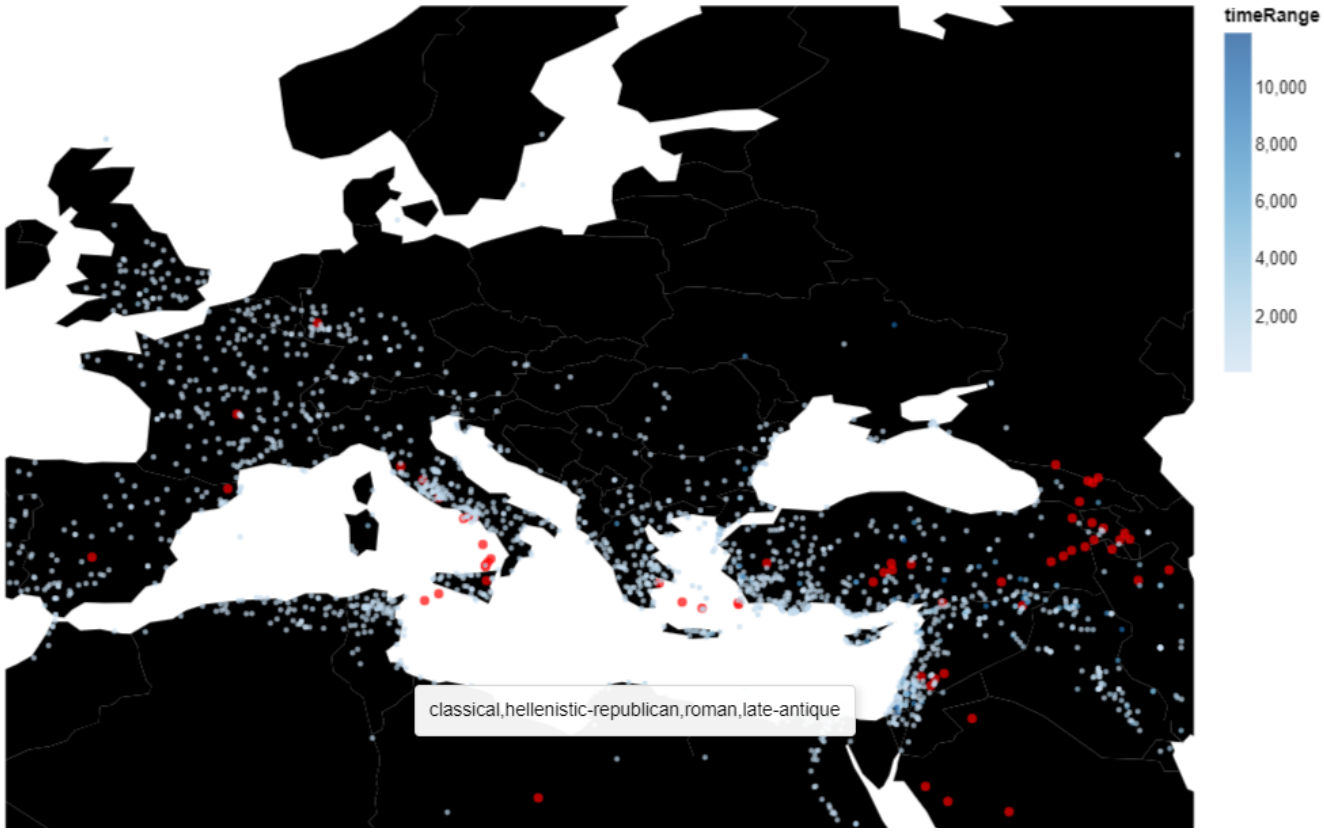
Was there any special data preparation done?

The locations of the volcanos was gathered by the dataset available at: https://data.humdata.org/dataset/volcano-population-exposure-index-gvm/resource/e3b1ecf0-ec47-49f7-9011-6bbb7403ef6d. The volcanos are shown in red, while the colour of the settlements represents their longevity. The longevity data is produced by subtracting values from the maxDate column and the minDate column. Settlements that are older than 60000 years were dropped. Samples that aren't settlements were also dropped. Tooltips are provided both for volcanos and settlements. Hovering the mouse over a volcano shows its name, hovering the mouse over a settlement shows its time period.

What are the limitations of your design?

The datasets used are not detailed enough to shoe any obvious pattern. It would be more useful to use a dataset that contains a volcano's activity over time. This way, one could only show volcanos that were active during the time period of the settlements around it. Furthermore, many settlements have low longevity, making it harder to spot differences between them. Furthermore, some areas in the dataset have a really high density of settlements, which might be covering some important information to the viewer.


timeRange
10,000
8,000
6,000
4,000
2,000
Manzaz Volcanic Field


timeRange
10,000
8,000
6,000
4,000
2,000
classical,hellenistic-republican,roman,late-antique

```
#Volcano dataset at: https://data.humdata.org/dataset/volcano-population-exposure-index-gvm/resource/e3b1ecf0-ec47-49f7-9011-6bbb7403ef6d

import IOReader as io
import pandas as pd
import altair as alt
import numpy as np
from vega_datasets import data

alt.renderers.enable('altair_viewer')
alt.data_transformers.disable_max_rows()
locationsData = pd.read_csv("pleiades-locations-latest.csv")
volcanoData = pd.read_csv("volcano.csv")
countries = alt.topo_feature(data.world_110m.url, 'countries')

#preprocessing
locationsData['timeRange'] = locationsData['maxDate'] - locationsData['minDate']

#dropping unnecessary data
locationsData.drop(locationsData[~(
    locationsData['featureType'] == "settlement")].index, inplace=True)
locationsData.drop(
    locationsData[locationsData['timeRange'] > 60000].index, inplace=True)

#scaling specification
scaling = alt.selection_interval(bind='scales')

#width and height of window
widthValue = 700
heightValue = 700

#volcano chart specification
volcanoChart = alt.Chart(volcanoData).mark_circle().encode(
    longitude='Longitude',
    latitude='Latitude',
    color = alt.value('red'),
    tooltip = 'V_Name'
).properties(
    width=widthValue,
    height=heightValue,
)
```

```
#pleiades chart specification
pleiadesChart = alt.Chart(locationsData).mark_point(
    filled=True,
    size=10
).encode(
    longitude='reprLong',
    latitude='reprLat',
    color='timeRange:Q',
    tooltip = 'timePeriodsKeys'
).properties(
    width=widthValue,
    height=heightValue
)

#geomap parameters
scaleValue = 700
translation = [100, 800]

#geomap specification
mapData = alt.layer(
    alt.Chart(countries).mark_geoshape(fill='black'),
    volcanoChart,
    pleiadesChart
).project(
    type='equirectangular', scale=scaleValue, translate=translation
).properties(width=widthValue, height=heightValue).configure_view(stroke=None)

mapData.show()
```