Author: Mohammad Hadi Jalali Lak (932173)

```
locations = pd.read_csv("pleiades-locations-latest.csv")
names = pd.read_csv("pleiades-names-latest.csv")
places = pd.read_csv("pleiades-places-latest.csv")
alt.data_transformers.disable_max_rows()
```

Getting started

Before starting the each visualization I'd like to provide a block of python code where some data preperation was made. It's appropiate to include this block here because these preperations apply for almost all of the visualizations(4 out of 5) I've presented. This block converts multiple feature types to one (the first mentioned type), removes rows with no specified feature Type and at last filters out the feature types with too many or not enough counts. (not unkwnon but na)

```
names = names[names['nameTransliterated'].notna()]
locations_ = locations_[locations_['featureType'].notna()]
locations_.loc[locations_['featureType'].str.contains('fort'), 'featureType'] = 'fort'
locations_.loc[locations_['featureType'].str.contains('villa'), 'featureType'] = 'villa'
locations_.loc[locations_['featureType'].str.contains('settlement-modern'), 'featureType'] = 'settlement-modern'
locations_.loc[locations_['featureType'].str.contains('river'), 'featureType'] = 'river'
locations_.loc[locations_['featureType'].str.contains('temple'), 'featureType'] = 'temple'
locations_.loc[locations_['featureType'].str.contains('station'), 'featureType'] = 'station'
locations_.loc[locations_['featureType'].str.contains('cemetery'), 'featureType'] = 'cemetery'
locations_.loc[locations_['featureType'].str.contains('island'), 'featureType'] = 'island'
locations_.loc[locations_['featureType'].str.contains('city-gate'), 'featureType'] = 'city-gate'
locations_.loc[locations_['featureType'].str.contains('amphitheatre'), 'featureType'] = 'amphitheatre'
locations_.loc[locations_['featureType'].str.contains('theatre'), 'featureType'] = 'theatre'
locations_.loc[locations_['featureType'].str.contains('tomb'), 'featureType'] = 'tomb'
locations_.loc[locations_['featureType'].str.contains('mountain'), 'featureType'] = 'mountain'
locations_.loc[locations_['featureType'].str.contains('bridge'), 'featureType'] = 'bridge'
locations_.loc[locations_['featureType'].str.contains('sanctuary'), 'featureType'] = 'sanctuary'
locations_.loc[locations_['featureType'].str.contains('church'), 'featureType'] = 'church'
locations_.loc[locations_['featureType'].str.contains('bath'), 'featureType'] = 'bath'
locations_.loc[locations_['featureType'].str.contains('city-wall'), 'featureType'] = 'city-wall'
locations_.loc[locations_['featureType'].str.contains('architecturalcomplex'), 'featureType'] = 'architecturalcomplex'
locations_.loc[locations_['featureType'].str.contains('monument'), 'featureType'] = 'monument'
locations_.loc[locations_['featureType'].str.contains('cape'), 'featureType'] = 'cape'
locations_.loc[locations_['featureType'].str.contains('tumulus'), 'featureType'] = 'tumulus'
locations_.loc[locations_['featureType'].str.contains('lake'), 'featureType'] = 'lake'
locations_.loc[locations_['featureType'].str.contains('plaza'), 'featureType'] = 'plaza'
locations_.loc[locations_['featureType'].str.contains('townhouse'), 'featureType'] = 'tonwhouse'
locations_.loc[locations_['featureType'].str.contains('regiom'), 'featureType'] = 'region'
locations_.loc[locations_['featureType'].str.contains('nuraghe'), 'featureType'] = 'nuraghe'
locations_.loc[locations_['featureType'].str.contains('water-open'), 'featureType'] = 'water-open'
locations_.loc[locations_['featureType'].str.contains('port'), 'featureType'] = 'port'
locations_.loc[locations_['featureType'].str.contains('settlement,'), 'featureType'] = 'settlement'
locations_.loc[locations_['featureType'].str.contains('unknown'), 'featureType'] = 'unknown'
```

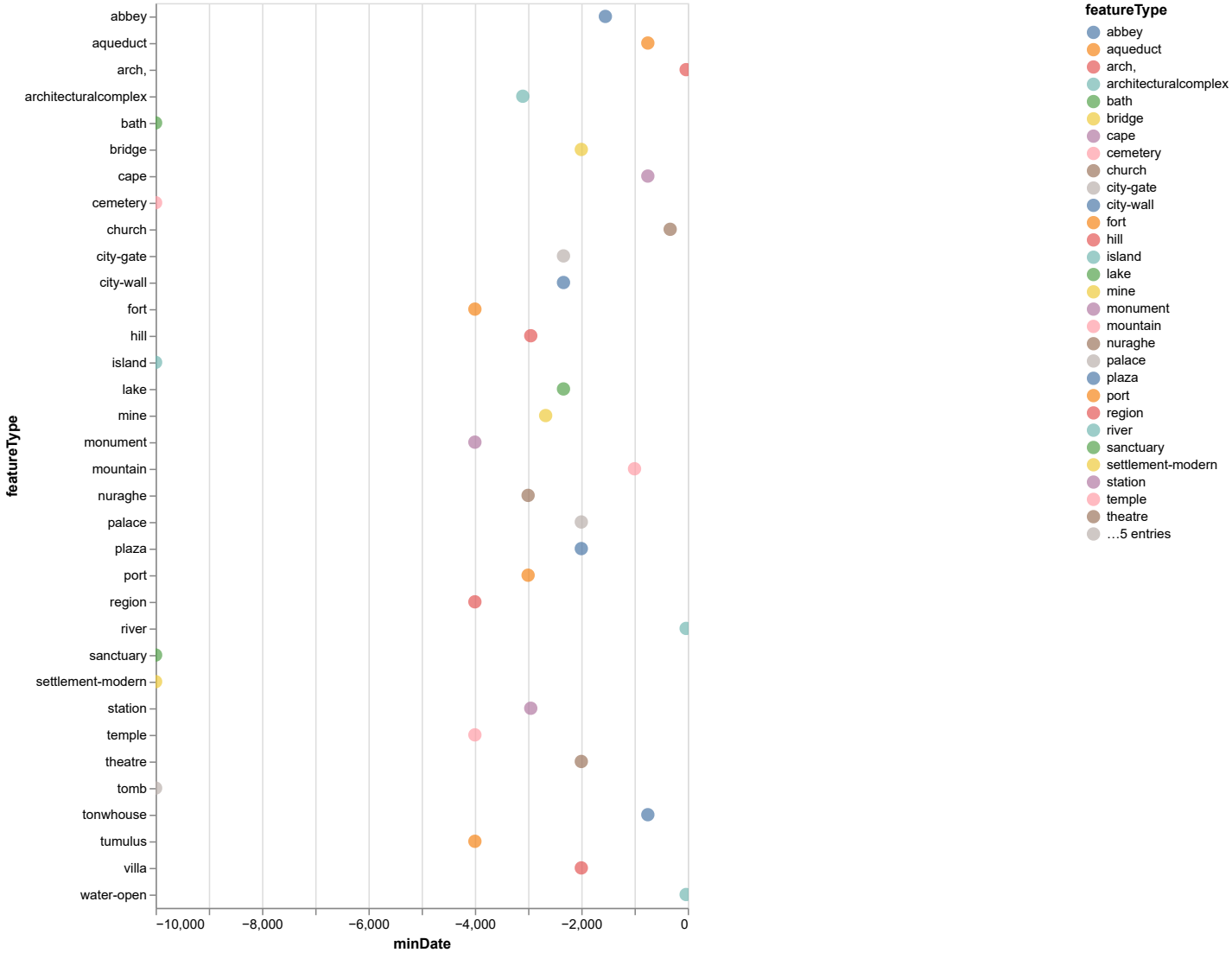
```
sub_locations= locations_[locations_.groupby('featureType').featureType.transform('count')>15].copy()
sub_locations= sub_locations[sub_locations.groupby('featureType').featureType.transform('count')<20_000].copy()
sub_locations['featureType'].value_counts()
sub_locations2 = sub_locations[sub_locations.groupby('featureType').featureType.transform('count')<1500].copy()
```

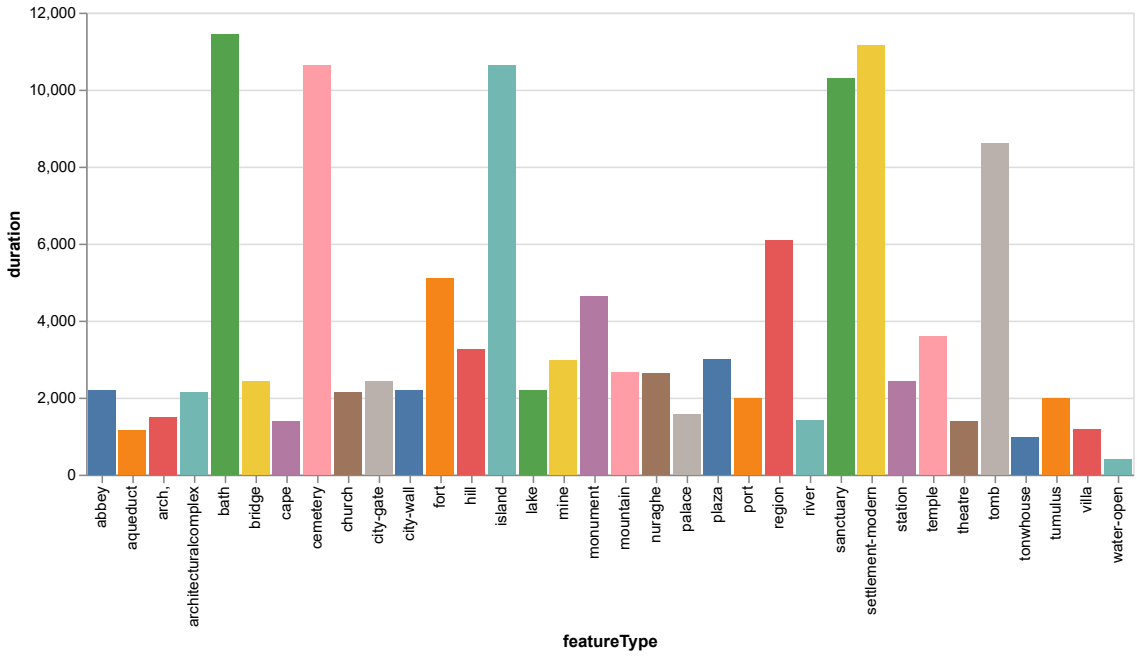
Oldest Types Of Civilizations

What can we learn from the visualization?

This combination of a bar chart and scatter plot show the first types of civilizations formed along with the longest standing ones. As represnted, tombs, settlements and sanctuaries are one of the first to form. They are also the categories with the longest duration. Tootltips are used to give more information on the time period and descrption of a specific record. The interactivity of these graphs makes it possible for the user to select and compare multiple feature types(by selecting on the y interval)

```
minimumDatesPerCategory = sub_locations2.loc[sub_locations2.groupby('featureType', sort=False)['minDate'].idxmin()]
interval = alt.selection_interval(encodings=['y'])


oldestLoc = alt.Chart(minimumDatesPerCategory).mark_circle(size = 100).encode(
    y='featureType',
    x='minDate',
    color=alt.condition(interval, 'featureType', alt.value('lightgray')),
    order = 'minDate',
    tooltip = ['featureType','description']
).properties(
    selection=interval
).interactive()

durationLoc = alt.Chart(sub_locations2).mark_bar().encode(
    y='duration',
    x='featureType',
    color='featureType',
    tooltip = ['timePeriodsRange']
).transform_filter(
    interval
).interactive()
plotLoc = oldestLoc & durationLoc
plotLoc
```





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

Linked interactive bar chart and scatter plot with available selection

What are all visual mappings used?

Scatter Plot:

y positions

featureType category

x positions

date of the oldest record (minDate)

colour

featureType category

tooltip

decription and feature type of the location and

Bar Chart:

x positons

featureType category

length

duration of the feature type (duration)

colour

featureType category

tooltipp

time range of the feature type

Was there any special data preparation done?

• Data were aggregated by category to generate counts and minimum date
• limited columns were selected to improve the efficiency
• New column 'duration' was made using minDate and MaxDate
• feature typee lists were reduced to one main type to make the graph more clear
• 'unknown' and 'settlement' types were removed becase of having too many counts and also not adding anything valuable to visualizations
• less popular feature types were removed to make the visualization more clear

What are the limitations of your design?

Despite removing a lot of feature types, there might still be one too many points on the graph for some people's liking so because of this, some colours may be too similar. However, with the help of tooltips and second graph this is not a huge problem. I personally believe this was the good amount of data to include. the dates might also not be too accurate

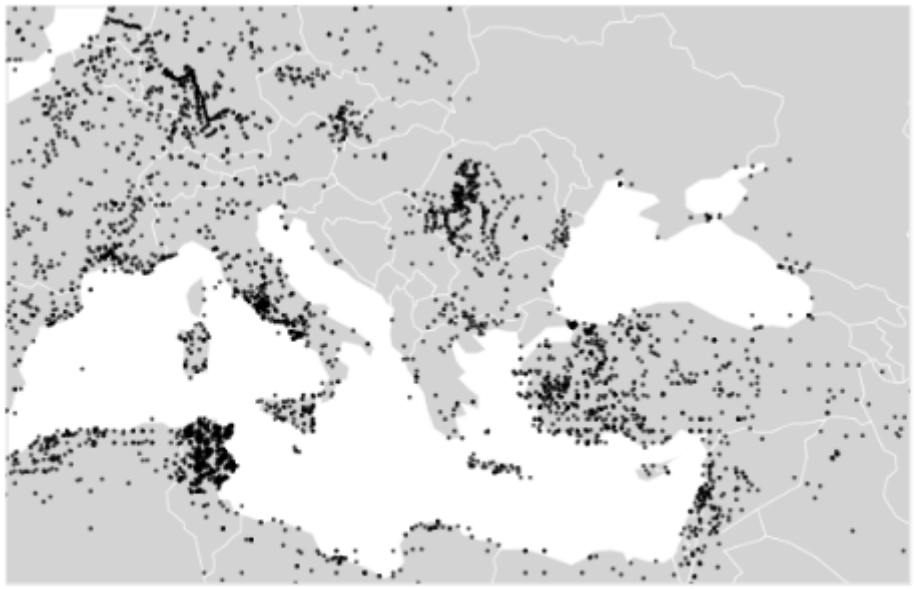
Distribution of Records in Different Time Periods

What can we learn from the visualization?

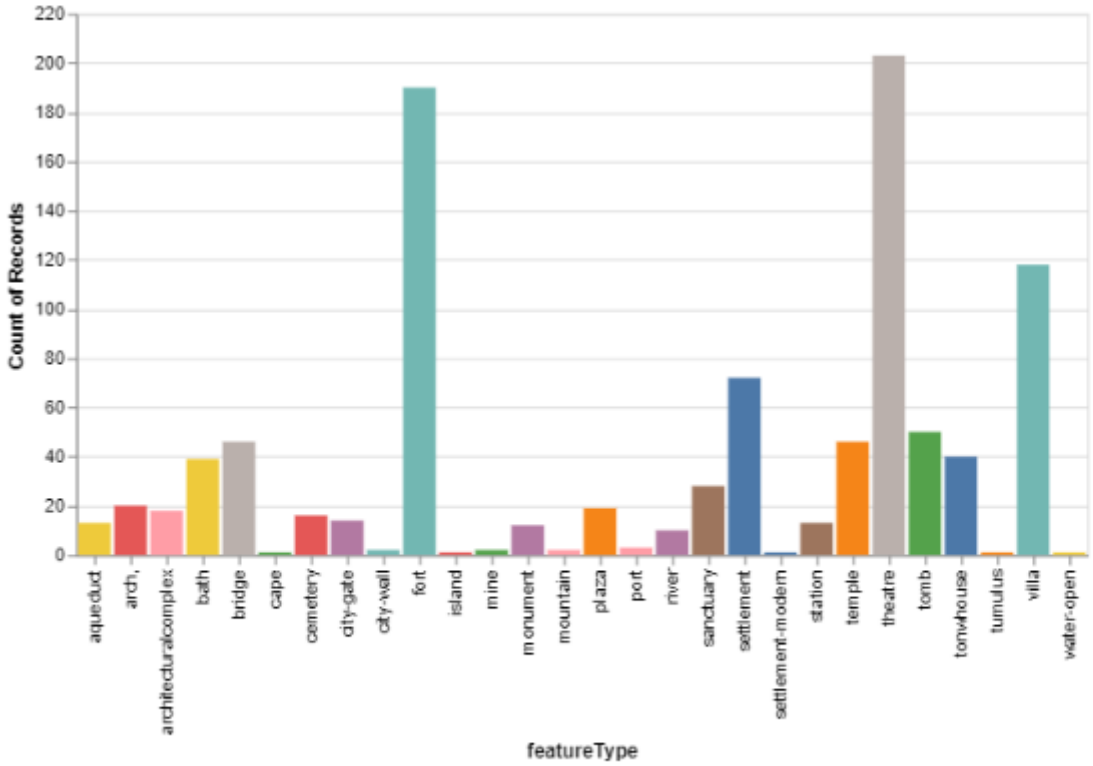
This combination of a map and a bar chart with an added selector, helps the user to get a better understanding of the distribution of the recorded civilizations(location wise and also feature type wise) in different time periods. The user can compare and see where most people were settled in any time period and what sort of locations were used and buildings were bulit in this specific time period. This is very helpful to see any type of immigration that's happened at any time. Users can go back and forth with the time periods using the drop down menu and observe different information. Tooltips are also used in the map to observe the feature type of any point on the map. It can be gathered from the map that the first civilizations were formed in Greece and Italy and spreaded to nearby locations quickly.

Because of the big amount of data (even with filtering out unused columns), this visualization wouldn't load in visual studio where it only took couple of seconds in jupyter. So I decided to include pictures instead of vega-lite code.

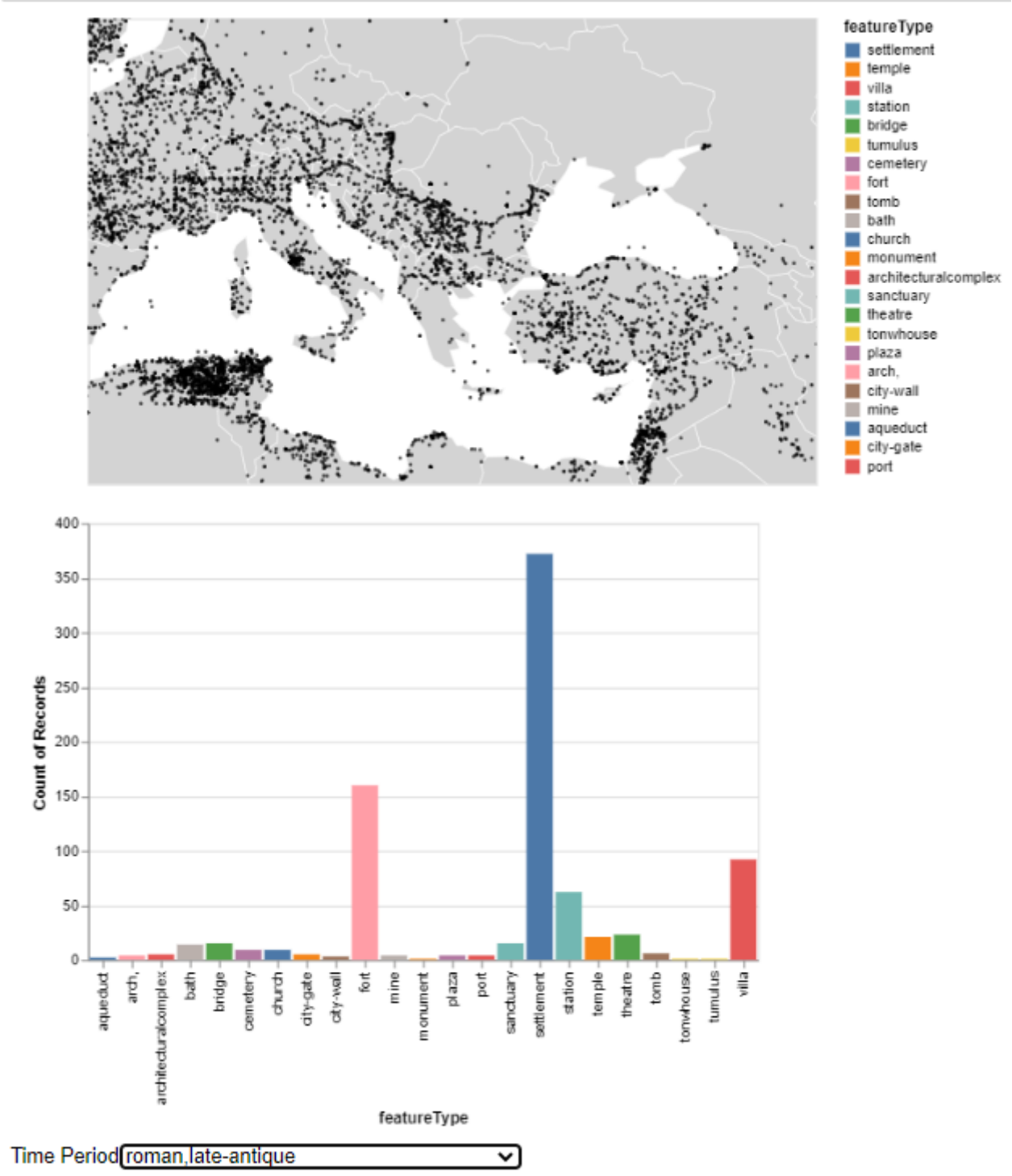
```
map =  alt.topo_feature(data.world_110m.url, 'countries')
data_vis2 = locations_[['reprLong','reprLat','featureType','timePeriodsKeys']]
data_vis2_ = sub_locations[['featureType','timePeriodsKeys']]
input_dropdown = alt.binding_select(options=['classical,hellenistic-republican,roman','hellenis
                                             'hellenistic-republican,roman',
                                             'hellenistic-republican,roman,late-antique',
                                             'roman','roman,late-antique','late-antique','moder
selection = alt.selection_single(fields=['timePeriodsKeys'], bind=input_dropdown)
color1 = alt.Color('featureType',
            sort=alt.EncodingSortField('count', order='descending'))
main= alt.layer(
    alt.Chart(map).mark_geoshape(
        fill='lightgray',
        stroke='white',
        strokeWidth = 0.5
    ),
    alt.Chart(data_vis2).mark_circle().encode(
        longitude='reprLong:Q',
        latitude='reprLat:Q',
        size=alt.value(5),
        tooltip='featureType',
        color = alt.value('black')
    ).add_selection(
    selection
).transform_filter(
    selection)
).properties(
    width=400,
    height=220
).project(
    type = "mercator",
    scale = 600,
    translate = [10,650]
).properties(
    width=500,
    height=320
)
bar3 = alt.Chart(data_vis2_).mark_bar().encode(
    y='count()',
    x='featureType',
    color = color1
).transform_filter(
    selection
).interactive()

main & bar3
```



featureType
- settlement
- temple
- island
- villa
- cape
- water-open
- river
- mountain
- station
- bridge
- settlement-modern
- tumulus
- cemetery
- fort
- tomb
- bath
- monument
- architecturalcomplex
- sanctuary
- theatre
- tonwhouse
- plaza
- arch,
- city-wall
- mine
- aqueduct
- city-gate
- port



Time Period roman



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

Linked map and bar chart with a drop down selector menu

What are all visual mappings used?

Map:

latitude

latitude of location

longitude

longitude of location

colour

black (to prevent confusion with bar chart colours)

tooltips

featureType category

Bar Chart:

x positons

featureType category

colour

featureType category

length

count of records with this specific feature type

Was there any special data preparation done?

• Data were aggregated by category to generate counts
• limited columns were selected to improve the efficiency
• 'unknown' and 'settlement' types were removed becase of being an being too big and also not adding anything valuable to visualizations
• less popular feature types were removed to make the visualization more clear
• less popular feature types were removed to make the visualization more clear

What are the limitations of your design?

This map only focuses on the most important part of the maps and may be not ideal for user trying to evaluate the distruibution though the whole world. The user can't really see the specific feature on the map unless using the tooltip. an improvement can be making multiple maps of different areas and also including an option to colour the points based on their category. however I decided not to include this in my map because it would be dominated by unknwon category. Removing unknown category wasn't something I opted for neither because the puropose of the map should be the distribution opf all the points not specific ones

Author Profiles and Comparison

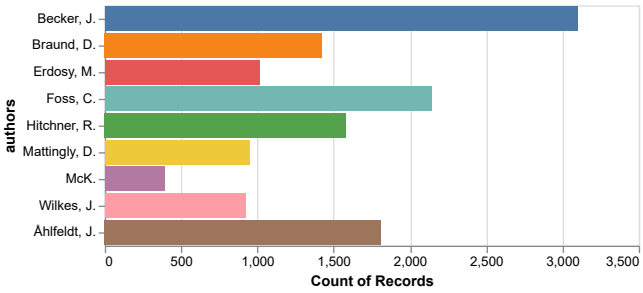
What can we learn from the visualization?

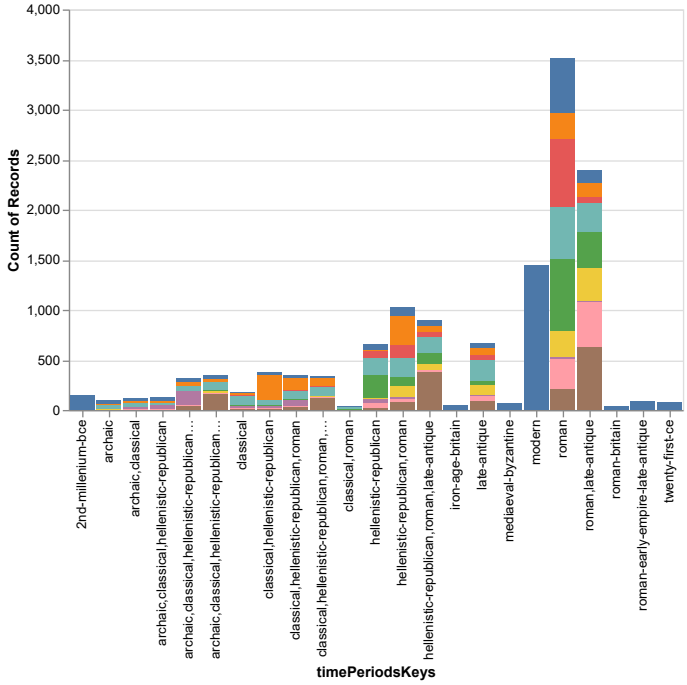
This presentation gives useful insight to the top 9 authors work. The user can see which time period or what type of civilization each author is interested in and make a profile based on these information. using a selector feature, it is also possible to compare the authors with each other by clicking their name in each graph(and holding shif to choose multiple). Even without selection, the colours on the second and third graph makes it easy to see which author dominates which feature type or time period. This can be very helpful for the authors or any other researcher. They may be interestied in working with other authors specialized in an specific time period or feature type. Users can check the credibility of a source hey've used by exploring the authors profile as well.

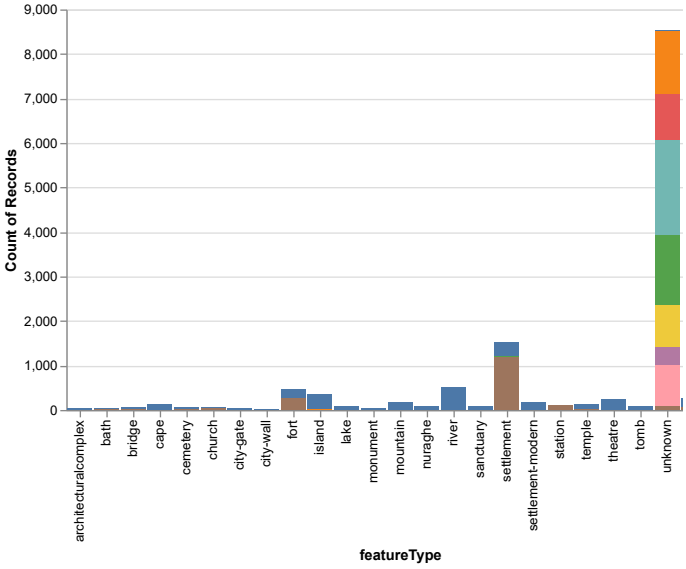
Tidying up authors column:

```
_locations_ = locations_
_locations_.loc[_locations_['authors'].str.startswith('Becker'), 'authors'] = 'Becker, J.'
_locations_.loc[_locations_['authors'].str.startswith('Hitchner'), 'authors'] = 'Hitchner, R.'
_locations_.loc[_locations_['authors'].str.startswith('Erdosy, M.'), 'authors'] = 'Erdosy, M.'
_locations_.loc[_locations_['authors'].str.startswith('Braund, D.'), 'authors'] = 'Braund, D.'
_locations_.loc[_locations_['authors'].str.startswith('McK'), 'authors'] = 'McK.'
_locations_.loc[_locations_['authors'].str.startswith('Wilkes, J.'), 'authors'] = 'Wilkes, J.'
_locations_.loc[_locations_['authors'].str.startswith('Mattingly, D.'), 'authors'] = 'Mattingly, D.'
_locations_.loc[_locations_['authors'].str.startswith('Bender, H.'), 'authors'] = 'Bender, H.'
_locations_.loc[_locations_['authors'].str.startswith('Haselgrove, C.'), 'authors'] = 'Haselgrove, C.'
_locations_.loc[_locations_['authors'].str.startswith('Foss, C.'), 'authors'] = 'Foss, C.'
_sub_locations2 = _locations_[_locations_.groupby('authors').authors.transform('count')>800].copy()
_sub_locations2 = _sub_locations2[_sub_locations2.groupby('featureType').featureType.transform('count')>30].copy()
_sub_locations2 = _sub_locations2[_sub_locations2.groupby('timePeriodsKeys').featureType.transform('count')>30].copy()
```

```
data_vis3 = _sub_locations2[['authors','featureType','timePeriodsKeys']]
selections = alt.selection_multi(fields=['authors'])
color2 = 'authors'
author_count = alt.Chart(data_vis3).mark_bar().encode(
    x='count()',
    y='authors',
    color = alt.condition(selections, 'authors' , alt.value('lightgrey'))
).properties(
selection = selections)

ww = alt.Chart(data_vis3).mark_bar().encode(
    y='count()',
    x='timePeriodsKeys',
    color = color2,
    tooltip = 'timePeriodsKeys'
).transform_filter(
    selections
).interactive()

we = alt.Chart(data_vis3).mark_bar().encode(
    y='count()',
    x='featureType',
    color = color2
).transform_filter(
    selections
).interactive()
author_count & (ww | we)
```







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

3 Linked bar charts

What are all visual mappings used?

Authors chart:

y positions

Authors name

length

number of records belonging to that author

colour

author category

Time Periods Chart:

x positons

time periods category

length

number of records belonging to that time period

colour

author category

tooltip

time periods (in case the full string doesn't fit on the plot)

Feature Type Chart:

x positons

feature type category

length

number of records belonging to that feature type

colour

author category

Was there any special data preparation done?

• Data were aggregated by category to generate counts and minimum date
• limited columns were selected to improve the efficiency
• multiple feature types were reduced to one main type to make the graph more clear
• multiple authors were reduced to one main author to make the graph more clear
• less popular feature types were removed to make the visualization more clear
• less active authors were removed to make the visualization more clear

What are the limitations of your design?

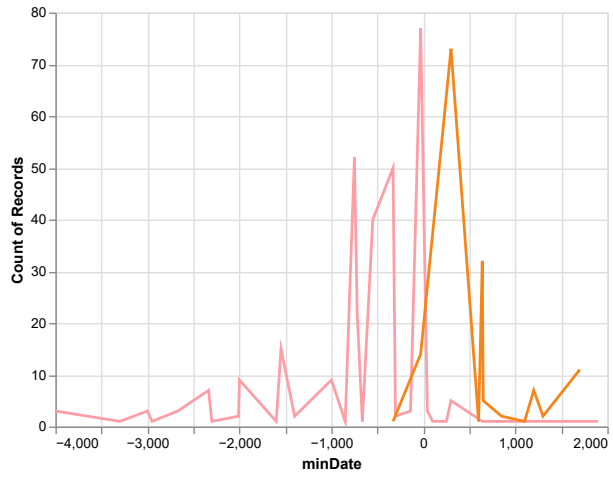
apart from only including nine authors, this visualization might apppear a little bit confusing at first, specially the time periods chart with multiple colours and labels. However, in a second look it should be clear to the user what the colours and labels mean. Another flaw in this presentation is the fact that an author is not given credit for some of the contributions they've had in project and mostly the main author or the more known author's name was taken. An improvment can be made in the data prepration part to make this cisualization more accurate.

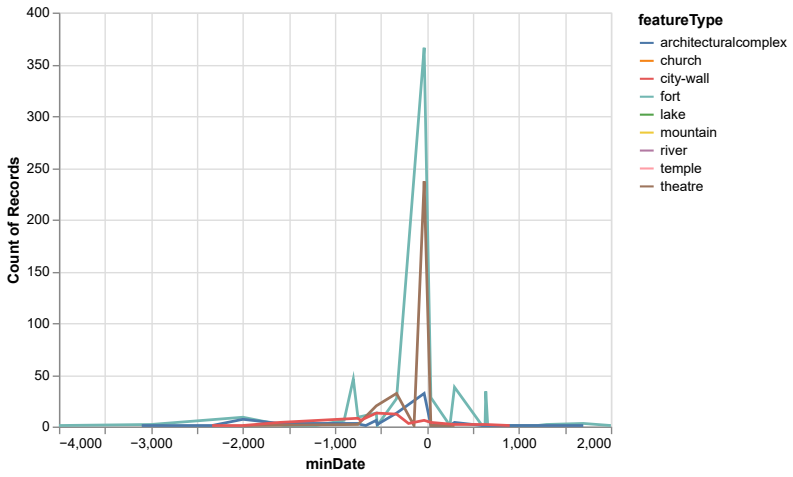
Changes of paradoxical or similar feature types

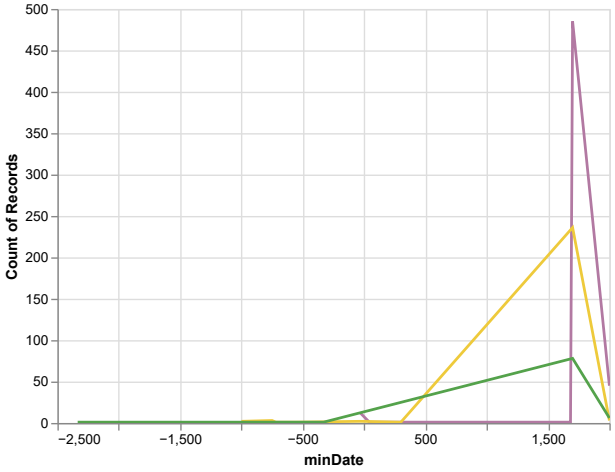
What can we learn from the visualization?

This presentation, despite being very simple but it is very effective. By comparing the some paradoxical feature types we can compare the lifes of people in different eras and also see the effects of various historical events. in the first line graph, the user can see the effects of appearence of christ and the popularization of christianity on the architectures of the cities and settlements. As seen in the first graph this was resulted in the decrease of temples and the number of churches increased.
By comparing the counts of more complex buldings we can see the time period where the human civilization was improved massively. From this graph we can understand in which time period with multiple wars going on by analyzing the peaks of number of forts.
And from the last we can see the people have moved close to rivers from mountain between the years of 1500 and 2000.

```
study_cases = ['church','temple']
data_vis3 = locations_ [['featureType','minDate']]
target_points = data_vis3.loc[data_vis3['featureType'].isin(study_cases)]

church_and_temple = alt.Chart(target_points).mark_line().encode(
    x='minDate',
    y='count()',
    color='featureType'
).interactive()

study_cases2 = ['fort','architecturalcomplex','city-wall','theatre']
target_points2 = data_vis3.loc[data_vis3['featureType'].isin(study_cases2)]

fort_and_theatre = alt.Chart(target_points2).mark_line().encode(
    x='minDate',
    y='count()',
    color='featureType',
    tooltip = 'minDate'
).interactive()

study_cases3 = ['mountain','river','lake']
target_points3 = data_vis3.loc[data_vis3['featureType'].isin(study_cases3)]

ups_and_downs = alt.Chart(target_points3).mark_line().encode(
    x='minDate',
    y='count()',
    color='featureType',
    tooltip = 'minDate'
).interactive()
(church_and_temple | fort_and_theatre) & ups_and_downs
```







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

3 Interactive multi series line chart

What are all visual mappings used?

y positions

number of records

x positions

date

colour

featurType

tooltip

date

Was there any special data preparation done?

• in each graph, data was filtered to only take the rows with targeted feature types
• limited columns were selected to improve the efficiency

What are the limitations of your design?

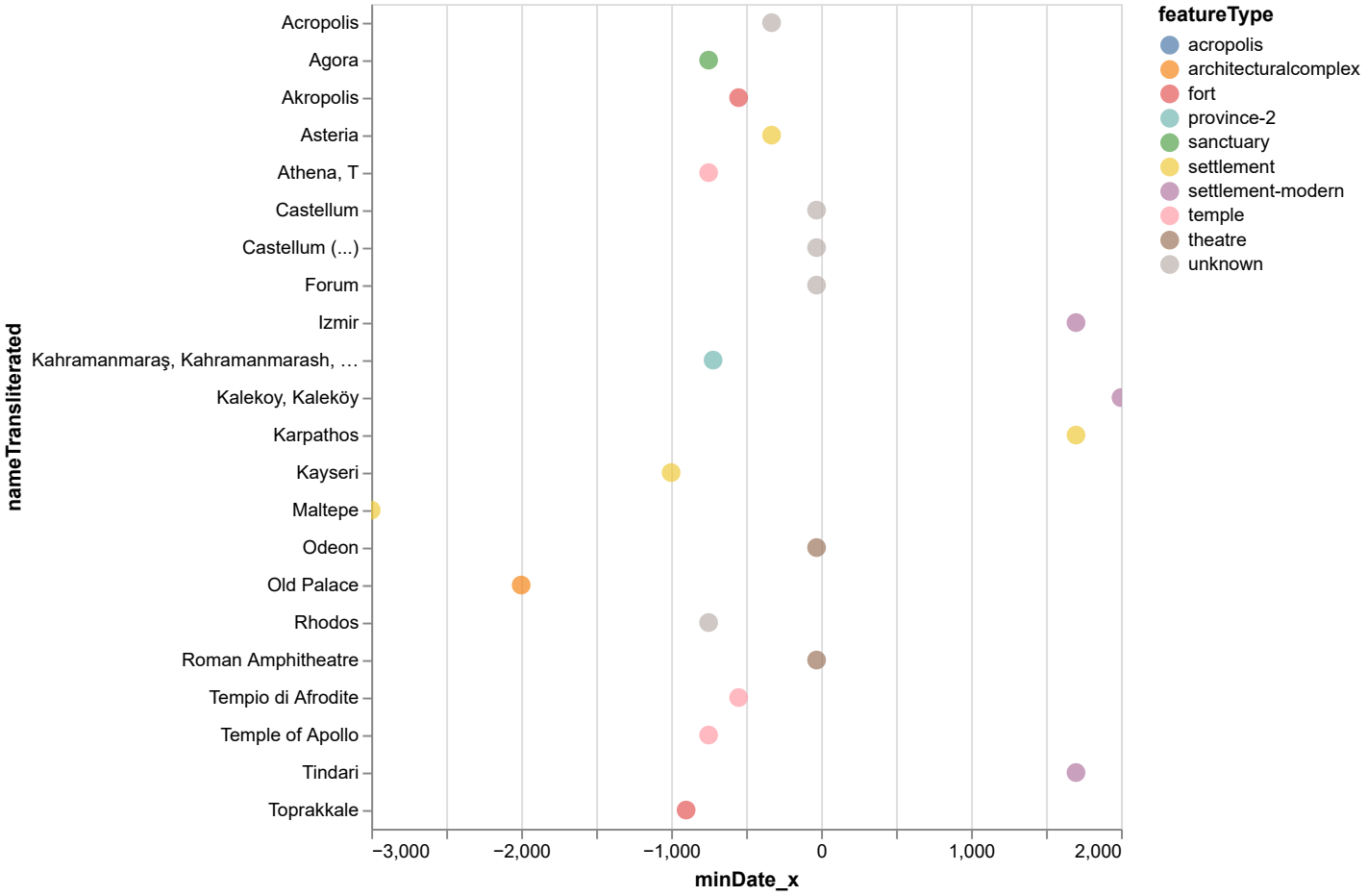
The biggest limitation (or maybe positive?)of this visualization might be the oversimplification of the graphs. There isn't much information at first glance but with some context and prior knowledge, these 3 graphs can provide much variable informations as mentioned in the aim section. An improvement would be to find more pattern like this and compare them.

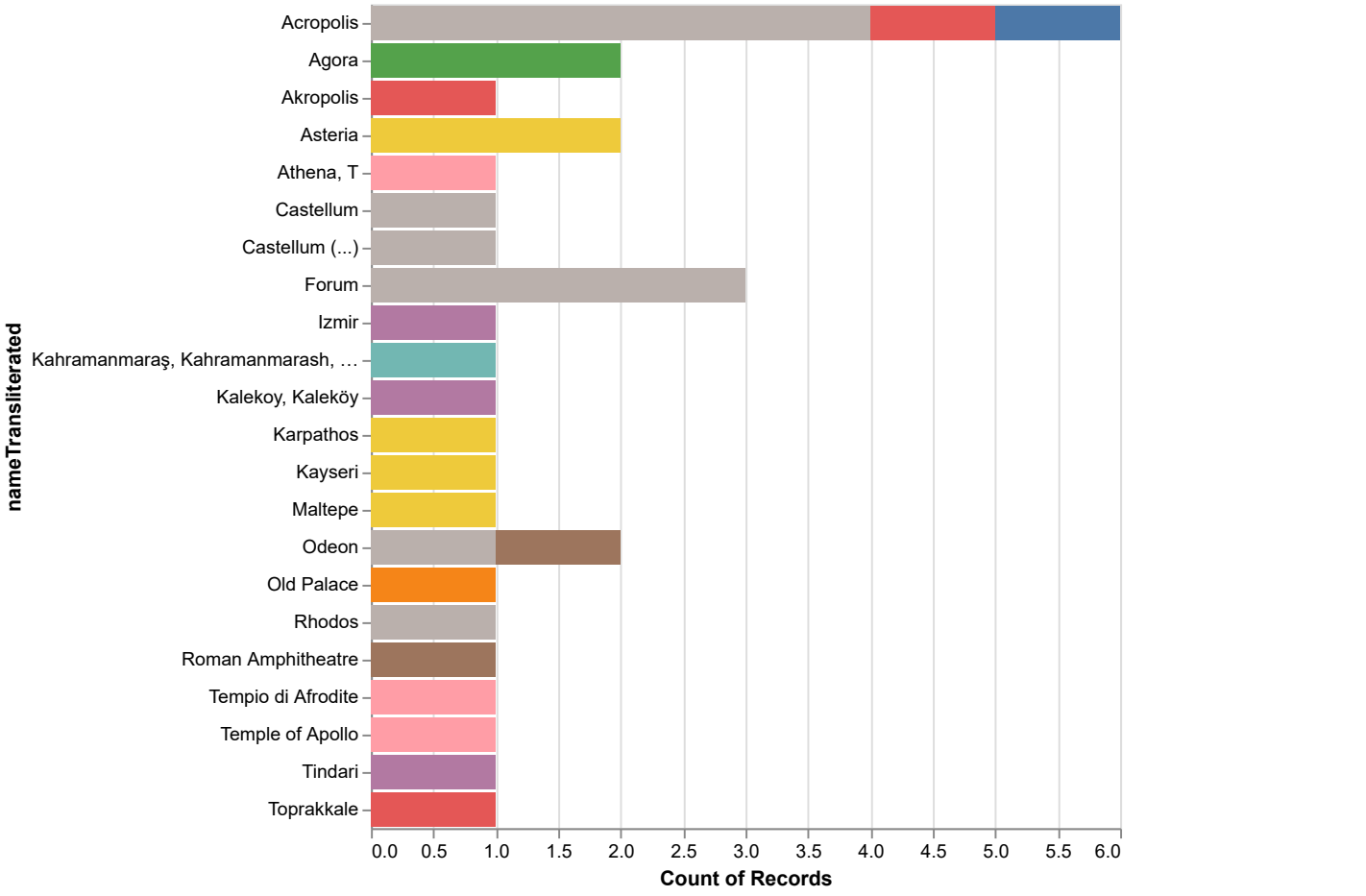
First feature type in each location

What can we learn from the visualization?

With this presentation, it is possible to check the first type of civilization that was found in each of the named locations. This helps us understand when each locatins was civilized and what type of civilization it was . Using selection on the second chart we can also compare two or more location to see which one was civilized before. further information on the number of feature types in each locations is also available to observe from the second chart.

```
new_locations = locations_[['id','featureType','minDate']]
new_names = names[['nameTransliterated','id','minDate']]
new = pd.merge(locations_,names,on = ['id'], how='inner')

data_vis4 = new.loc[new.groupby('nameTransliterated', sort=True)['minDate_x'].idxmin()]
multi = alt.selection_multi(fields=['nameTransliterated'])


dates = alt.Chart(data_vis4).mark_circle(size = 100).encode(
    y='nameTransliterated',
    x='minDate_x',
    color='featureType',
    tooltip = 'featureType'
).transform_filter(multi
).interactive()

counts = alt.Chart(new).mark_bar().encode(
    y='nameTransliterated',
    x='count()',
    color = alt.condition(multi, 'featureType' , alt.value('lightgrey')),
    tooltip = 'featureType'
).properties(
    selection=multi
)
dates & counts
```





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

Interactive scatter plot and bar chart with multi selection avbility

What are all visual mappings used?

Scatter Plot:

x positions

name of the location

y positions

date of the first found civilization

colour

featureType category

tooltip

featureType category

Bar Chart:

x positions

name of the location

length

number of records found in this location

colour

featureType category

tooltip

featureType category

Was there any special data preparation done?

• locations and names datasets were combined on column 'id'

• Data were aggregated by category to generate minimum date

What are the limitations of your design?

This visualization takes a big hit from the lack of connection between the two databases . only a handdful of rows in locations and names dataset can be merged and these locations dobn't have many records. howeever this visualization is still a very informative on these specific locations.