

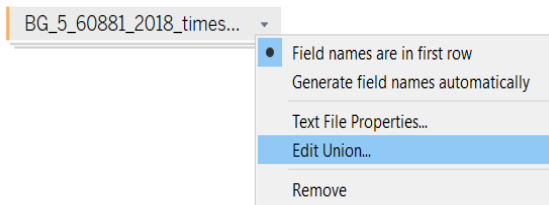
Pre-Processing and Import

Dataset 1/4 : AIR QUALITY DATA (EEA Folder)

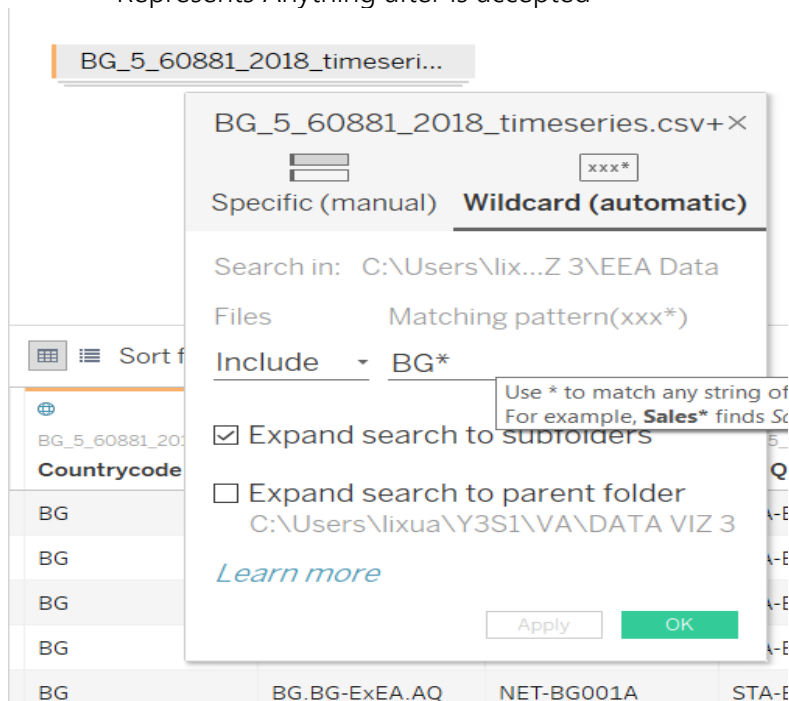
Contains numerous files *BG_5_{ ID of Sampling point }_{ Year }.csv*

1. Load any CSV first
2. Select edit union to add additional fields

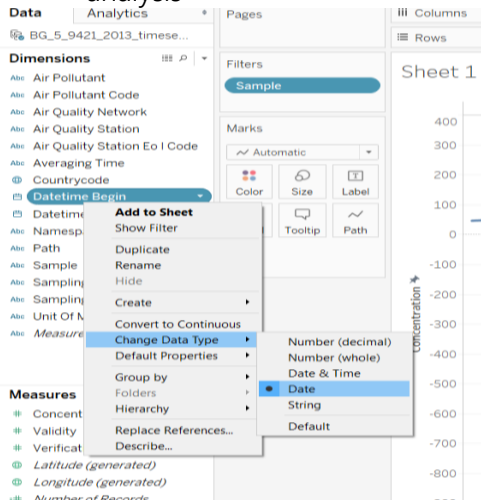
⊖ BG_5_9421_2013_timeseries.csv+ (M



3. Use a wildcard to add every file in the folder that starts with BG*
* Represents Anything after is accepted



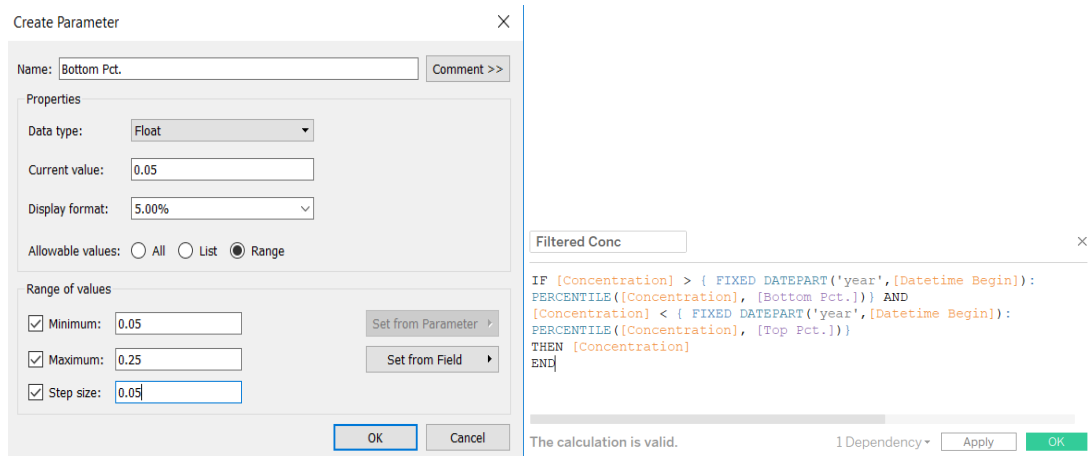
- Change DateTime Begin and DateTime End to Date, since we would want a daily analysis



- The default aggregation of all data within the day is SUM. Remember to always use average



- Judging by the green lines, there are outliers that we may consider excluding. Create a lower percentile parameter. From dimensions, data pane, click the drop-down arrow in the upper right corner and select Create Parameter. Right-click Bottom Pct. in the Parameter pane of the Data window and select Show Parameter Control.



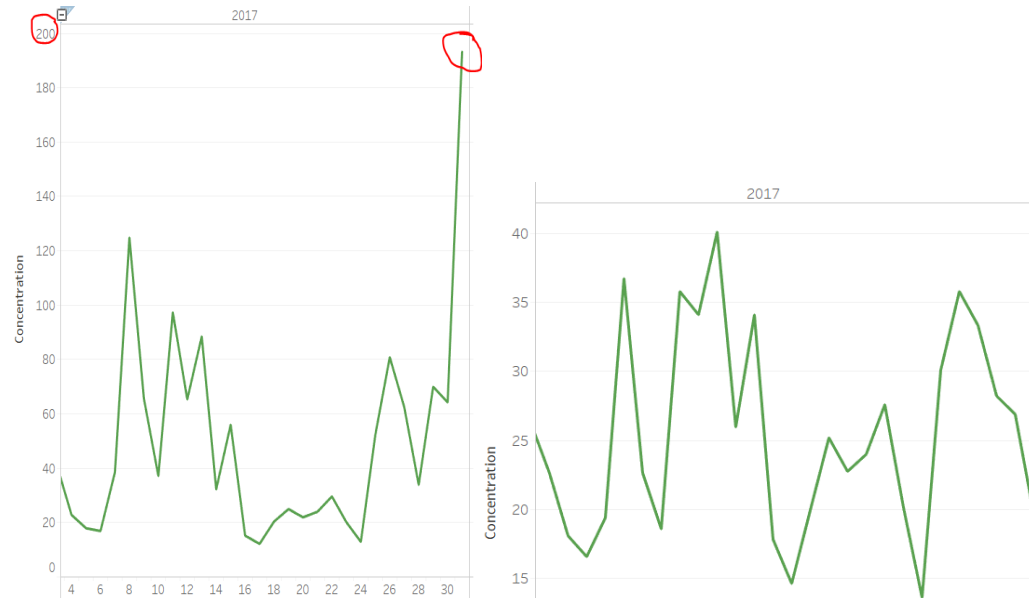
Repeat the above steps to create another parameter Top Pct. with Minimum to 0.75, Maximum to 1.0 and Step size to 0.05.

7. Drag the calculated field Filtered Conc created above to the Filters shelf.

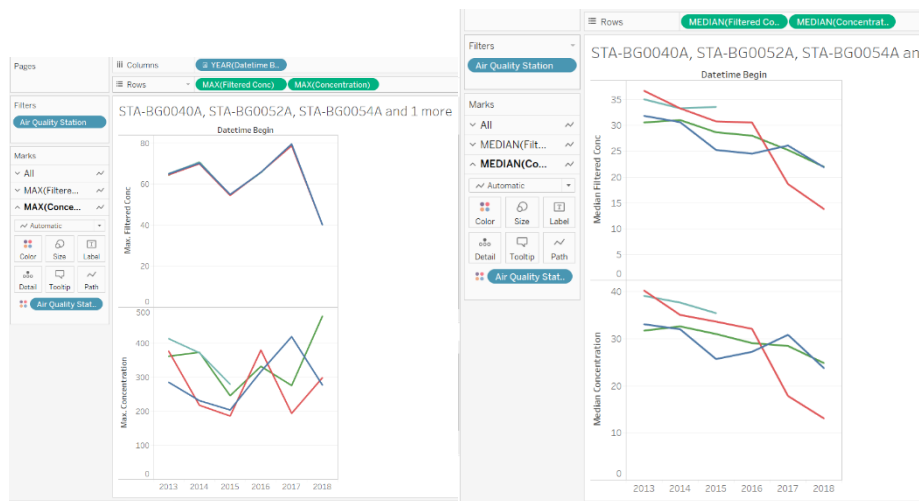
>In the Filter's dialog box

>All values >Next

> Special > Non-null values > OK.

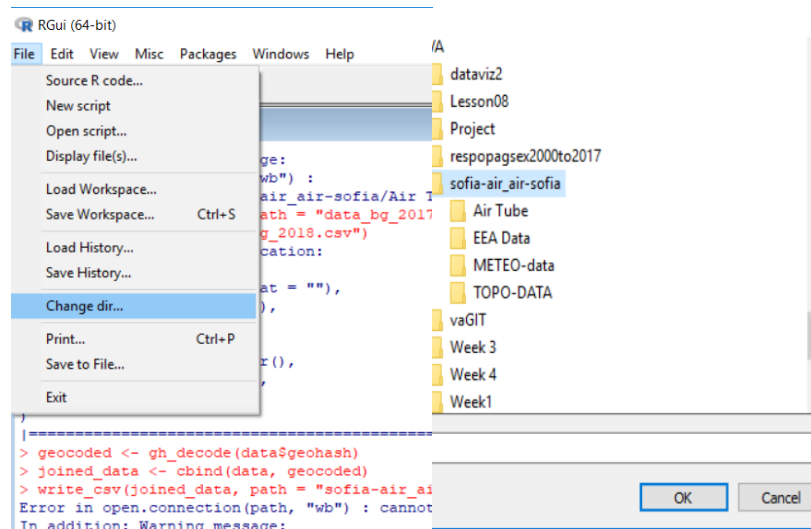


Before after shots shown!



Geohash is used to represent topology (gridded) but Tableau cannot interpret it.
How to we convert Geohash to GeoCode?

Download r, change directory to the unzipped Sofia's data file: *sofia-air_air-sofia*



Follow the following codes (Given by Prof)

Installing and Launching r packages
`packages = c('devtools', 'tidyverse')`

```
for(p in packages){
  if(!require(p, character.only = T)){
    install.packages(p)
  }
  library(p, character.only = T)
}
```

Next, use the code chunk below to install geohash package from github source.

```
devtools::install_github("ironholds/geohash")
library(geohash)
```

Importing Data

```
data <- read_csv("sofia-air_air-sofia/Air Tube/data_bg_2017.csv")
```

Geocoding

```
geocoded <- gh_decode(data$geohash)
```

Performing Relation Join

```
joined_data <- cbind(data, geocoded)
```

Saving output file

```
write_csv(joined_data, path = "sofia-air_air-sofia/Air Tube/data_bg_2017_geocoded.csv")
```

I will want to merge citizen science and EEA data together. From both data sources, I'll need the Lat, Long, PM10 Conc.

Ensure that EEA has Lat and Long.

Use Tableau, Inner join (Air Quality Stn) to merge with metadata.csv as provided. Then extract the output as a CSV.

Use Python to change the headers in both EEA merged csv and CitizenScience.

```
import pandas as pd

1 # combine both Dataset

df2017 = pd.read_csv("data_bg_2017_geocoded.csv")
df2018 = pd.read_csv("data_bg_2018_geocoded.csv")
df = pd.concat([df2017, df2018], ignore_index=True)

1 # Print current headers

names = df.columns.values.tolist()
print(names)

1 # Rename headers

df.rename(columns={'time':'Time','P1':'PM10 Conc','lat':'Latitude','lng':'Longitude'},inplace=True)

1 # Output specific columns only

header = ["Time", "PM10 Conc", "Latitude", "Longitude","P2","temperature", "humidity", "pressure"]
df.to_csv("outputCitizen.csv", columns = header)

EEA = pd.read_csv("EEA With Lat Long Meta data.csv")

EEANames = EEA.columns.values.tolist()
EEA.fillna(0)
EEA.reindex()

EEA.rename(columns={'Datetime Begin':'Time','Concentration':'PM10 Conc','Latitude':'Latitude','Longitude':'Longitude'},inplace=True)
EEANames = EEA.columns.values.tolist()
print(EEANames)
<
>

header = ( "Time", "PM10 Conc", "Latitude", "Longitude")
df.to_csv("outputEEA2.csv", columns = header)

print(pd.read_csv("outputEEA2.csv"))
```

Use a union to import again
Use a wildcard to add every file in the folder that starts with *

Merge columns if they are not recognised as the same thing.