Visual Data Analytics CST4060

Coursework 2

MS Data Science (Full-time 2023-2024)

By: Madeeha Solanki - M00984984

Date of Submission: 31/12/2023

Submitted To: Dr. Sudhakar Camilos Fernando

Department of Computer Engineering & Informatics

Block 16, Knowledge Park, Dubai

Section 1: Analysis Questions

1. Describe trends and anomalies with respect to chemical contamination

i. Trends: changes over time and/or sensor site

ii. Anomalies: sudden change over time or one site significantly different from others.

2. Describe any data quality and uncertain issues, such as

i. missing data,

ii. change in collection frequency, and

iii. Unrealistic values (e.g. water temperature higher than 100 degrees)

Section 2: Solution

Data Preprocessing-

1. The column 'sample date' was dropped to retain the new column 'sample date' because the previous column was converted to a date type format to extract the month, day and the year.

```
#the sample date is in object data type to perform a meaningful analysis it will be converted to date type.
data['sample_date']=pd.to_datetime(data['sample date'])
#the original column has to be dropped since that is no longer useful to perform analysis
data.drop(["sample date"], axis = 1, inplace = True)
```

2. I have categorized the sample date into day, week, month, and year. Categorizing into week was done manually using the weeknum() function in excel for the records in sample date column whereas the day, month and the year were worked out using the pandas library in python.

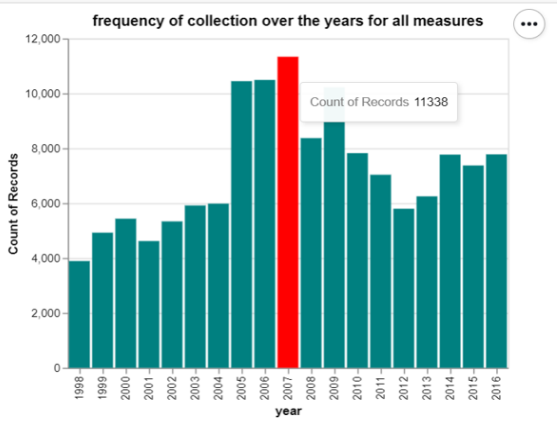
```
data['year'] = pd.DatetimeIndex(data['sample_date']).year
data['month'] = pd.DatetimeIndex(data['sample_date']).strftime('%b')
data['day'] = pd.DatetimeIndex(data['sample_date']).day
```

```
data.head() # to check whether the new fields have been updated or not
```

	id	value	location	measure	sample_date	year	month	day
0	2221	2.00	Boonsri	Water temperature	1998-01-11	1998	Jan	11
1	2223	9.10	Boonsri	Dissolved oxygen	1998-01-11	1998	Jan	11
2	2227	0.33	Boonsri	Ammonium	1998-01-11	1998	Jan	11
3	2228	0.01	Boonsri	Nitrites	1998-01-11	1998	Jan	11
4	2229	1.47	Boonsri	Nitrates	1998-01-11	1998	Jan	11

The quality of the data-

```
freq_chart=alt.Chart(data,title="frequency of collection over the years for all measures").mark_bar().encode(x='year:O'
,y='count(value)',tooltip=['count(value)'])

# Define the condition for color encoding
color_condition = alt.condition(
    alt.datum.year == 2007,
    alt.value('red'),  # if the condition is true that is for year 2007 the color will be red rest all will be teal
    alt.value('teal')
)
# Apply color encoding to the base chart
act_chart = freq_chart.encode(
    color=color_condition
)

# Display the chart
act_chart
```



Highlights: The graph shows that the lowest number of collections were in the year 1998 with a value of 3893.Whereas, the highest frequency in collection was in the year 2007 with a value of 11338.

The What: Year is the ordinal attribute here as there will be a specific ordering from the past to the latest.

The Why:

Aim: To find out the frequency in collection, to highlight the year with the highest collection of records and the year with the lowest collection of records.

Targets: Show the frequency and distribution of collections and highlighting the most noticeable areas.

The How: to find out the frequency of collection with ordinal type of data it is best to use a bar graph to show the distribution over a period of time in an optimal manner.

Mark: Line

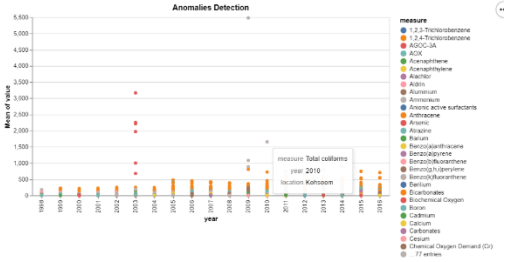
Channel:

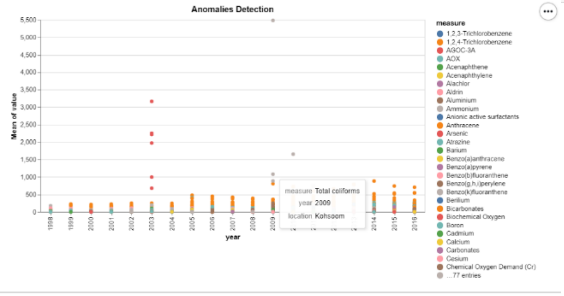
- Positions: x position: Ordinal Attribute, y position: Quantitative attribute
- Color: Conditional color encoding based on the year 2007, to highlight the highest frequency.
- Tooltip: Additional information displayed on hover like the count of records that is 'value'.

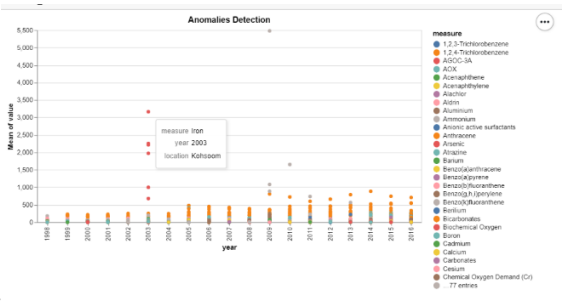
<u>Anomalies/ Outliers Detection</u>–

- **I have checked anomalies and the outliers for the measures one of the most significant outliers/ anomalies are detected for Iron and total coliforms. The measure of Iron was unusually high during 2003 in several different locations. However it was the highest in Kohsoom in 2003. Total coliforms were also high compared to other measures in the year 2009 and 2010 for the locations Achara and Kohsoom. If we take a look at their trends throughout the year they are not significant and constitute in one time rise which prove that these are anomalous.**

```
# Scatter plot to detect anomalies for a specific measure
scatter_anom = alt.Chart(data).mark_circle().encode(x='year:O',y='mean(value)',color='measure',
tooltip=['measure','year','location']).properties(title='Anomalies Detection',width=600)
scatter_anom
```







The What: Year is the ordinal attribute here as there will be a specific ordering from the past to the latest.

The Why:

Aim: to find out the outliers or the anomalies from the different measures and find out their location.

The How: to find out the outliers/ anomalies it is best to use scatter plot to easily identify the outliers.

Mark: point

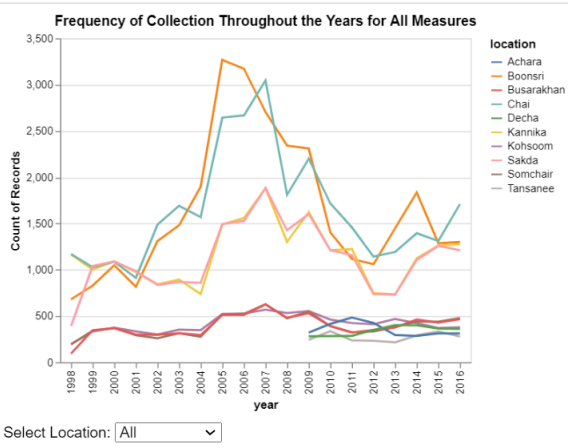
Channel:

- Positions: x position: Ordinal Attribute, y position: Quantitative attribute
- Color: to show the measures
- Tooltip: To gather the information such as location, year and the name of the measure.

2) missing data –

When we take a look at the different locations and their collection frequencies we can see that there are some locations where the collection frquency has no value in other words there was no sample collection in those areas until later years.Locations like Decha,Tansanee and Achara have no collection frequency before 2009, this indicates missing data for these locations.

```
locations = ['All'] + sorted(data['location'].unique())
location_dropdown = alt.binding_select(options=locations, name="Select Location: ")
location_select = alt.selection_single(fields=['location'], bind=location_dropdown, name='select_Location')
brush = alt.selection_interval(encodings=['x'], name='brush')

freq_chart = alt.Chart(data, title="Frequency of Collection Throughout the Years for All Measures").mark_line().encode(
    x='year:O',
    y="count(value)",
    color='location',
    tooltip=['count(value)','year']
).add_selection(location_select,brush)

# Use transform_filter to update the chart based on the selected location
filtered_chart = freq_chart.transform_filter(location_select).transform_filter(brush)

# Display the chart
filtered_chart
```



Highlights: Shows the missing values or inconsistency in the record collection in different location until later years.

The What: Year is the ordinal attribute here as there will be a specific ordering from the past to the latest showing the collection of records.

The Why:

Aim: To find out the frequency in collection for different locations and find out the missing data

Targets: show the inconsistency in the frequency of collection along with their locations.

The How: to find out the frequency of collection with ordinal type of data it is best to use a bar graph to show the distribution over a period of time in an optimal manner.

Mark: Line

Channel:

- Positions: x position: Ordinal Attribute, y position: Quantitative attribute
- Color: shows the different locations.
- Tooltip: Additional information displayed on hover like the count of records that is ‘value’, the year and the location.

3) **Change in collection frequency(over a period of time (day,month), for different measures):**

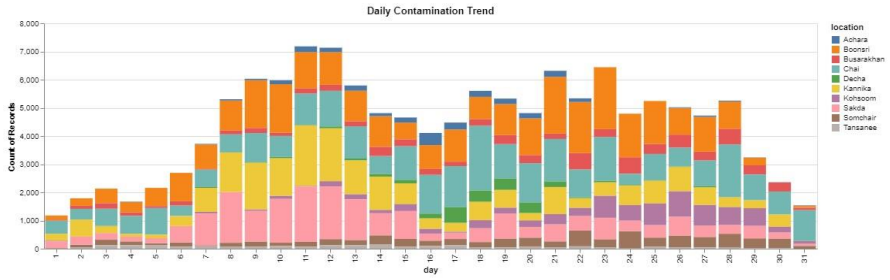
I have made graphs for everyday and monthly trends for checking the collection of records . The observations made from the graphs below is that overall everyday does not have the same amount of collection and for some locations the collection is missing. This is highly inconsistent.However,the monthly collections are almost close to each other. The highest collection was found in august and 11th day had the most collections where as the first day was the lowest.

Day :

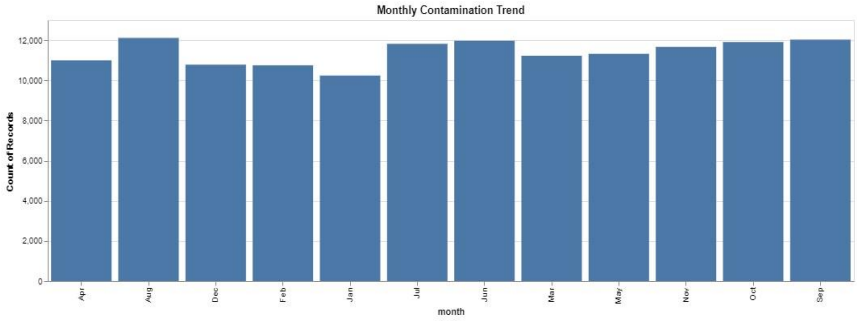
```
day_chart = alt.Chart(data).mark_bar().encode(
    x='day:O',
    y='count(value)',
    tooltip=['day:O', 'count(value)',],color='location'
).properties(
    title='Daily Contamination Trend',width=1000
)
day_chart
```

Month:

```
month_chart = alt.Chart(data).mark_bar().encode(
    x='month:O',
    y='count(value)',
    tooltip=['month:O', 'count(value)']
    #color='location'
).properties(
    title='Monthly Contamination Trend',
    width=1000
)
month_chart
```



Month:



Highlights: Shows the inconsistency in the record collection on the basis of days and months for different locations.

The What: Month and the day are the ordinal attribute here as there will be a specific ordering from the past to the latest showing the frequency in the collection of records.

The Why:

Aim: To find out the inconsistency in the frequency in collection for different locations and find out the missing data.

Targets: show the inconsistency in the frequency of collection along with their locations.

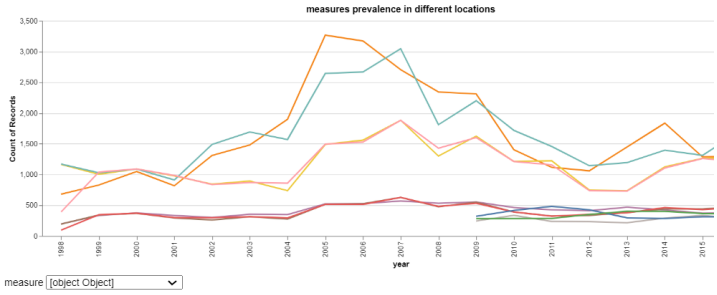
The How: to find out the inconsistency in the frequency of collection with ordinal type of data such as day and month it is best to use a bar graph to show the distribution over a period of time in an optimal manner.

Mark: Line

Channel:

- Positions: x position: Ordinal Attribute, y position: Quantitative attribute
- Color: shows the different locations.
- Tooltip: Additional information displayed on hover like the count of records that is 'value', the count of records for the specific location .

```
measure_dropdown = alt.binding_select(options=(data['measure']=='Ammonium','Nitrites','Nitrates',
'Anionic active surfactants','Chlorodinine','Magnesium','Methylosmoline','Total nitrogen','Water temperature','Ammonium'
                                               ,name="measure "))
measure_select = alt.selection_single(fields=['measure'], bind=measure_dropdown, name='select measure')

# Define the chart
day_chart = alt.Chart(data).transform_filter(
    measure_select
).mark_().encode(
    x='year:O',
    y='mean(value):Q',
    tooltip=['location', 'count(value)'],color='location'
).properties(
    title='measures prevalence in different locations',
    width=1000
).add_selection(
    measure_select
)

day_chart
```



Upon selecting a certain measure a zoom in version of that measure and the collection of that measures values will be displayed.

<u>iii. Unrealistic values (e.g. water temperature higher than 100 degrees)</u> -

I have plotted a heatmap against the month and the year for the measure 'Water temperature' . I have made observations where during certain month and certain year the values are suddenly changed where as the remaining months and the years it remains constant within a certain range eg. For April of 1998 the value recorded is 18.67. Where as, the remaining years for april the temperature was constant and there are more months with similar pattern.

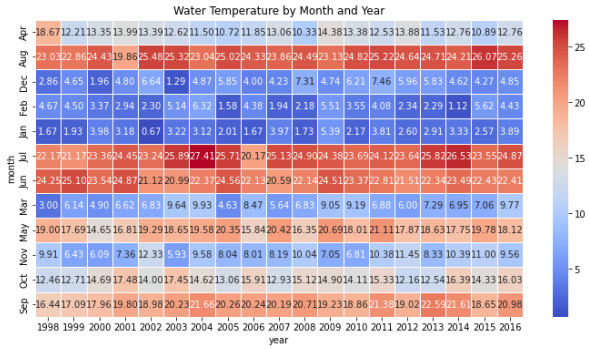
Highlights: Shows the unrealistic values for the water temperature along months and years.

The What: Month and the year are the ordinal attribute here as there will be a specific ordering from the past to the latest showing the unrealistic values ascending to descending.

The Why:

Aim: To find out which month and year shows unrealistic values for the water temperature. Targets: show the sudden temperature rise of water different from its regular pattern.

```
import seaborn as sns
import matplotlib.pyplot as plt

water_temp_data = data[data['measure'] == 'Water temperature']

grouped_data = water_temp_data.groupby(['location', 'month', 'year'])['value'].mean().reset_index()

plt.figure(figsize=(12, 6))
heatmap_data = grouped_data.pivot_table(index='month', columns='year', values='value', aggfunc='mean')
sns.heatmap(heatmap_data, cmap='coolwarm', annot=True, fmt=".2f", linewidths=.5)

plt.title('Water Temperature by Month and Year')
plt.show()


#5.*How is the water temperature in all locations by month and year?
```



The How: plotting an area chart will help to show the distribution of the recorded values along the month and the year making it easy to interpret the sudden change in temperature.

Mark: Area

Channel: Color Saturation