

# Assignment 2 - Redesign a Notable Information Graphic: Nightingale's Coxcomb

## Why I Chose to Redesign Nightingale's Coxcomb

Florence Nightingale's coxcomb is a famous visualization for its impact in history and clarity in illustrating mortality data in the Crimean War. Its design compared mortality causes effectively and emphasizes preventable deaths due to poor sanitation.

I chose to redesign it because, with modern tools, we can make the clarity better and even add interactive elements, making it more accessible and informative.

## Areas for Improvement

While the Nightingale's coxcomb is informative enough, there are areas where we can improve. For example, in the original design, the 24 months are split into two rings, which can make the comparing months across rights challenging. Additionally, the lack of metrics make it difficult to identify exact death tolls of each category. Finally, as a image on paper, it lacks interaction with audiences.

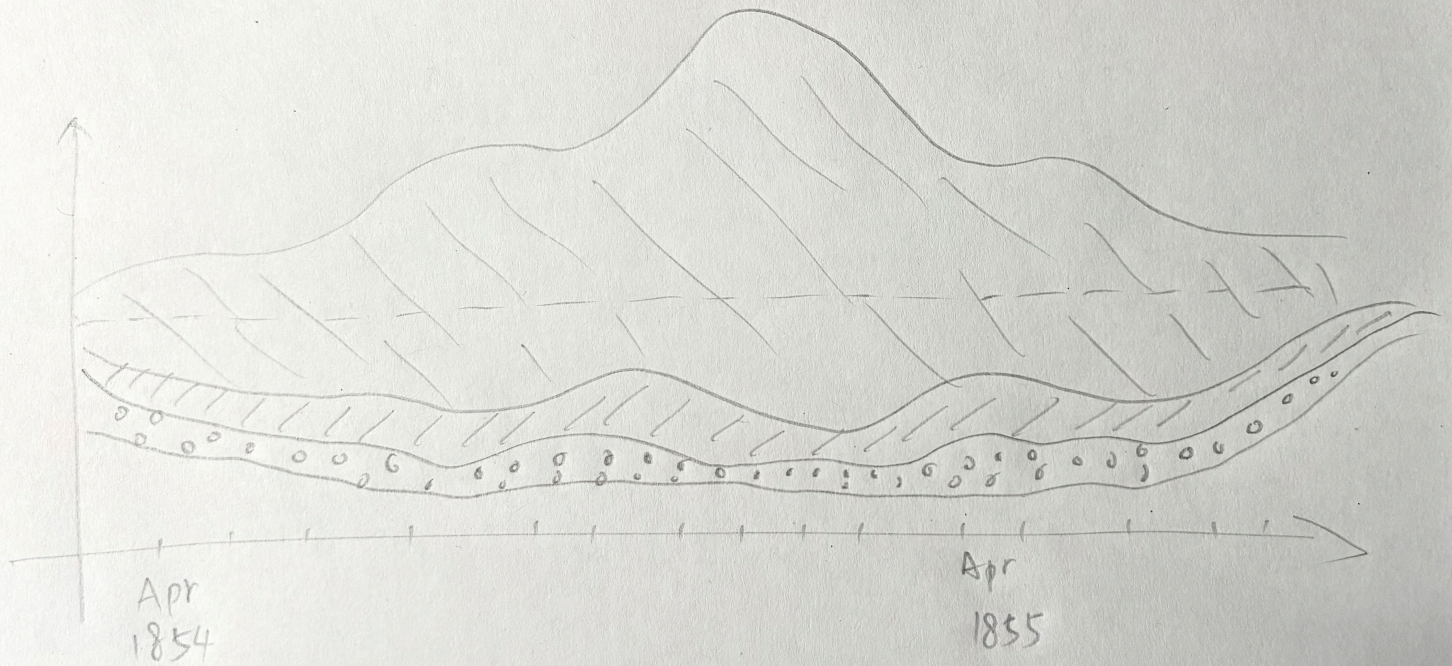
## Iteration 1

As seen below, my initial plan was to use a stacked bar chart. It is a natual choice as it aligns with the coxcomb's concept, just stretching both rings across a horizontal axis. Colors represented causes of death. But this design lacks storytelling depth and aesthetic appeal. To improve it, I connected categories by hand to better show the fluctuations. This led to the next iteration.

## Iteration 2

In this phase, I centered each category. It looks like mountains or clouds to show the trends.

From an online resource at <https://mode.com/blog/python-data-visualization-libraries>, I found that Altair's Steamgraph Altair perfectly met my requirements. It offers smooth transitions between categories, making the trends more visually compelling. This approach allows the audience to see increases and decreases clearly, with interactive displaying exact death tolls as they hover over each section. This enhances the engagement with audiences.



## Implementation and Limitations

The implementation uses the library of Altair to build the Streamgraph. The graph effectively shows how mortality rates declined after sanitary reforms were introduced in March 1855. The interactive features are also available. A vertical marker indicates the reform's timing.

However, technological limitations prevented me from adding more visual cues, such as a virus background for preventable diseases, or event-driven markers indicating specific reforms like improved ventilation.

## Conclusion

Nightingale's coxcomb is undoubtedly a landmark in data visualization, especially given its historical context. With the help of modern tool, we have the opportunities to redesign and enhance its visuals. Through the iteration design process, I learned to select appropriate visualizations for a clear storytelling, making data

more engaging and informative. This project has deepened my understanding of Python visualization, which will benefit my future work.

## Import Data from the Given CSV File

```
import pandas as pd
import altair as alt

# Ignore FutureWarnings which is not much relevant for this project
import warnings
warnings.simplefilter(action='ignore', category=FutureWarning)

# Load the data from the CSV file
data = pd.read_csv('data/nightingale.csv')

# Display the first few rows of the data
data.head()
```



	Unnamed: 0	Date	Month	Year	Army	Disease	Wounds	Other	Disease.rate	Wounds.rate	Other.rate
0	1	1854-04-01	Apr	1854	8571	1	0	5	1.4	0.0	0.0
1	2	1854-05-01	May	1854	23333	12	0	9	6.2	0.0	0.0
2	3	1854-06-01	Jun	1854	28333	11	0	6	4.7	0.0	0.0

## Streamgraph Visualization Implementation

```
# Convert the 'Date' column to datetime format and sort by date
data['Date'] = pd.to_datetime(data['Date'])
data = data.sort_values(by='Date')

# Melt the data to have a long format suitable for Altair
data_melted = data.melt(id_vars=['Date', 'Month', 'Year'], value_vars=['Disease', 'Wounds', 'Other'],
                        var_name='Cause', value_name='Deaths')

# Ensure the 'Cause' column is treated as a categorical variable
data_melted['Cause'] = data_melted['Cause'].astype('category')

# Create a streamgraph with Altair
streamgraph = alt.Chart(data_melted).mark_area().encode(
    x=alt.X('Date:T', title='Date', axis=alt.Axis(format='%b %Y')),
    y=alt.Y('Deaths:Q', stack='center', title='Number of Deaths'),
    color='Cause:N',
    tooltip=['Date:T', 'Cause:N', 'Deaths:Q']
).properties(
```



```
title='Monthly Mortality Rates by Cause Before and After Sanitary Reforms',
width=800 # Stretch the x-axis to clearly show each month
)

# Add a vertical line for the month when sanitary reforms were implemented (March 1855)
reform_line = alt.Chart(pd.DataFrame({
    'Date': [pd.to_datetime('1855-03-01')],
    'label': ['Sanitary Reforms Implemented']
})).mark_rule(color='red').encode(
    x='Date:T'
)

# Add text annotation for the sanitary reforms
reform_text = alt.Chart(pd.DataFrame({
    'Date': [pd.to_datetime('1855-03-01')],
    'label': ['Sanitary Reforms Implemented']
})).mark_text(align='left', dx=5, dy=-50, color='red').encode(
    x='Date:T',
    text='label'
)

# Combine the streamgraph, reform line, and reform text
final_chart = streamgraph + reform_line + reform_text

final_chart
```

