Visualização da Informação

Escola de Matemática Aplicada - Fundação Getúlio Vargas

Mestrado em Modelagem Matemática

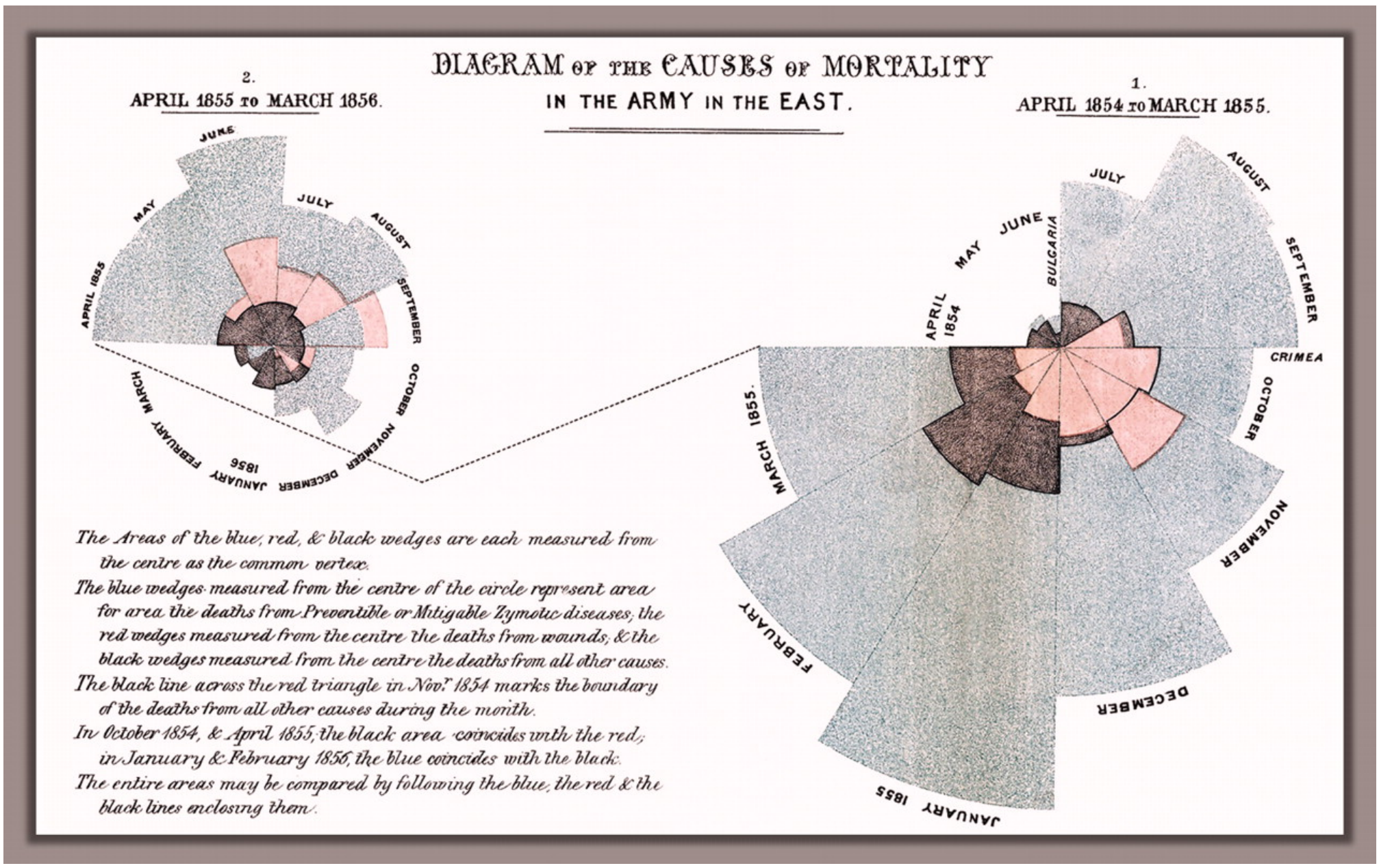
Aluno: Gianlucca Devigili
Github do projeto: https://github.com/GDevigili/information-visualization-homeworks

Trabalho 2: Análise e reprodução de uma visualização reconhecida ou relevante historicamente

- Parte1: Encontrar os dados (compartilhar referências de dados no slack)
- Parte2: Fazer uma análise de qual seria a função pretendida com a visualização proposta.
- Parte3: Fazer uma reprodução da visualização escolhida utilizando uma ferramenta computacional atual (de preferência a mesma escolhida por vocês no trabalho 1)
- Parte4: Propor alguma modificação (fundamentando conceitualmente) na visualização proposta. Exemplo: Incluir anotação, incluir interatividade, modificar título ou legenda, adicionar informação, etc.

In [1]:
```
import pandas as pd
import altair as alt
import plotly.express as px
from plotly.subplots import make_subplots

import vega_datasets

from altair_saver import save
```

A visualização escolhida foi a de **Florence Nightingale** referente às **causas de morte na guerra da Crimeia** (1853-1856)



(1) Aquisição dos dados

Para importar os dados, utilizei a biblioteca `vega_datasets` .
Inicialmente eu havia realizado um web-scrapping dos dados (como pode ser visto neste commit), porém ao plotar o gráfico percebi que os dados da fonte que eu peguei estavam errados e não reproduziam o gráfico de Nightingale, então preferi usar a biblioteca.

In [2]:
```
# Carrega os dados
df_crimea = vega_datasets.data.crimea()
df_crimea
```

Out[2]:

	date	wounds	other	disease
0	1854-04-01	0	110	110

	date	wounds	other	disease
1	1854-05-01	0	95	105
2	1854-06-01	0	40	95
3	1854-07-01	0	140	520
4	1854-08-01	20	150	800
5	1854-09-01	220	230	740
6	1854-10-01	305	310	600
7	1854-11-01	480	290	820
8	1854-12-01	295	310	1100
9	1855-01-01	230	460	1440
10	1855-02-01	180	520	1270
11	1855-03-01	155	350	935
12	1855-04-01	195	195	560
13	1855-05-01	180	155	550
14	1855-06-01	330	130	650
15	1855-07-01	260	130	430
16	1855-08-01	290	110	490
17	1855-09-01	355	100	290
18	1855-10-01	135	95	245
19	1855-11-01	100	140	325
20	1855-12-01	40	120	215
21	1856-01-01	0	160	160
22	1856-02-01	0	100	100
23	1856-03-01	0	125	90

Preparação dos dados

Para reproduzir o gráfico, precisamos dividi os dados em dois intervalos de tempo, sendo o primeiro indo de abril de 1854 até março de 1855 e o segundo de abril de 1855 até março de 1856.
Após isso, usei o método `pd.melt` para transformar o dataset de modo que ele tenha 3 colunas: `Date`, `Death` e `Cause`. Os dados então ficam da maneira apresentada abaixo:

In [3]:
```
causes = ['other', 'wounds', 'disease']

# Transforma o dataset completo em um dataset melted
df_melted = pd.melt(
        df_crimea,
        id_vars = ['date'],
        value_vars = causes,
        var_name = 'cause', value_name = 'deaths'
        )


# Formata a data no formato 'Apr 1854'
df_crimea['date'] = [date.strftime('%b %Y') for date in df_crimea['date']]

# Cria um dataset para cada período de 12 mesmos
df1 = pd.melt(
        df_crimea[:12],
        id_vars = ['date'],
        value_vars = causes,
        var_name = 'cause', value_name = 'deaths'
        )
df2 = pd.melt(
        df_crimea[12:],
        id_vars = ['date'],
        value_vars = causes,
        var_name = 'cause', value_name = 'deaths'
        )

# Remove o dia do dataset melted
df_melted['date'] = [date.strftime('%Y-%m') for date in df_melted['date']]

# Apresenta o dataset
df_melted.head()
```

Out[3]:

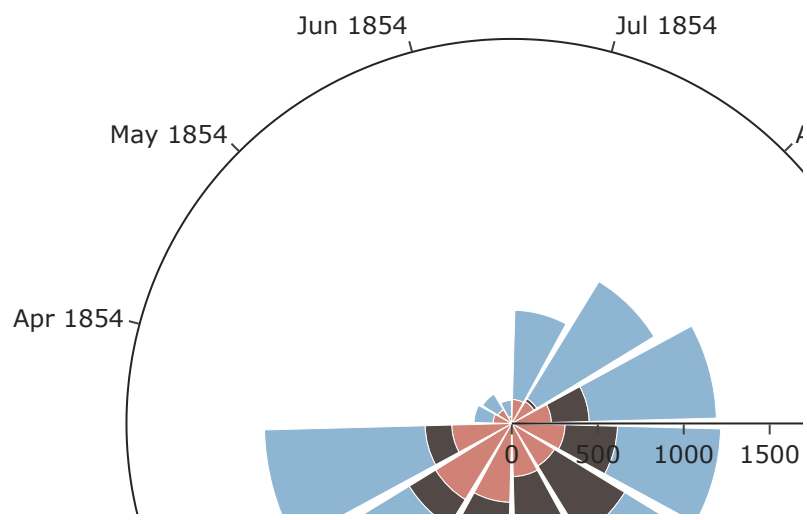
	date	cause	deaths
0	1854-04	other	110
1	1854-05	other	95
2	1854-06	other	40
3	1854-07	other	140
4	1854-08	other	150

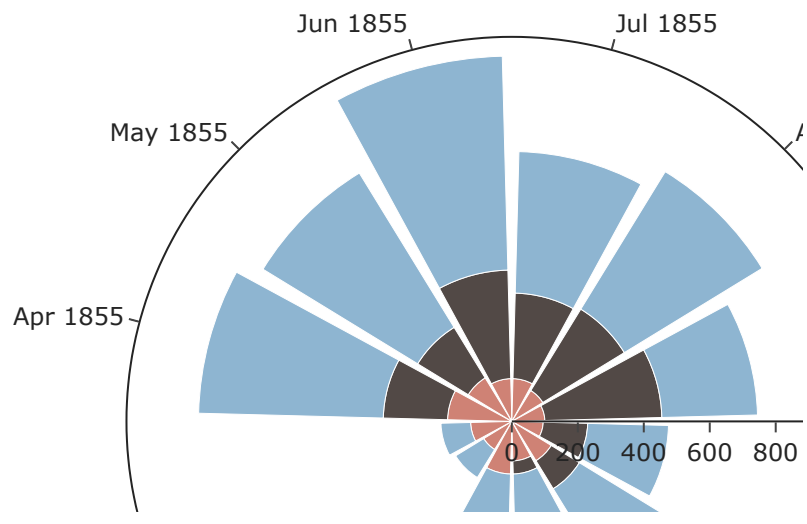
(2) Análise da Função da Visualização

Florence era uma enfermeira que atuou na Guerra da Criméia. A visualização que ela propôs tinah o intuito de evidenciar que a maior causa de morte entre seus pacientes era por doenças contraídas no campo de guerra, e não pelos ferimentos decorridos da mesma. A visualização teve grande contribuição para a aplicação de condições sanitárias melhores nos campos de batalha e hospitais, tendo uma melhora significativa no segundo ano.

(3) Reprodução da Visualização

In [4]:
```
colors = ['#CF8275', '#524946', '#8eb5d1']

fig1 = px.bar_polar(
    df1,
    r = 'deaths',
    color = 'cause',
    theta = 'date',
    start_angle = 165,
    color_discrete_sequence = colors,
    template = 'simple_white'
)

fig1.show()

fig2 = px.bar_polar(
    df2,
    r = 'deaths',
    color = 'cause',
    theta = 'date',
    start_angle = 165,
    color_discrete_sequence = colors,
    template = 'simple_white'
)

fig2.show()
```



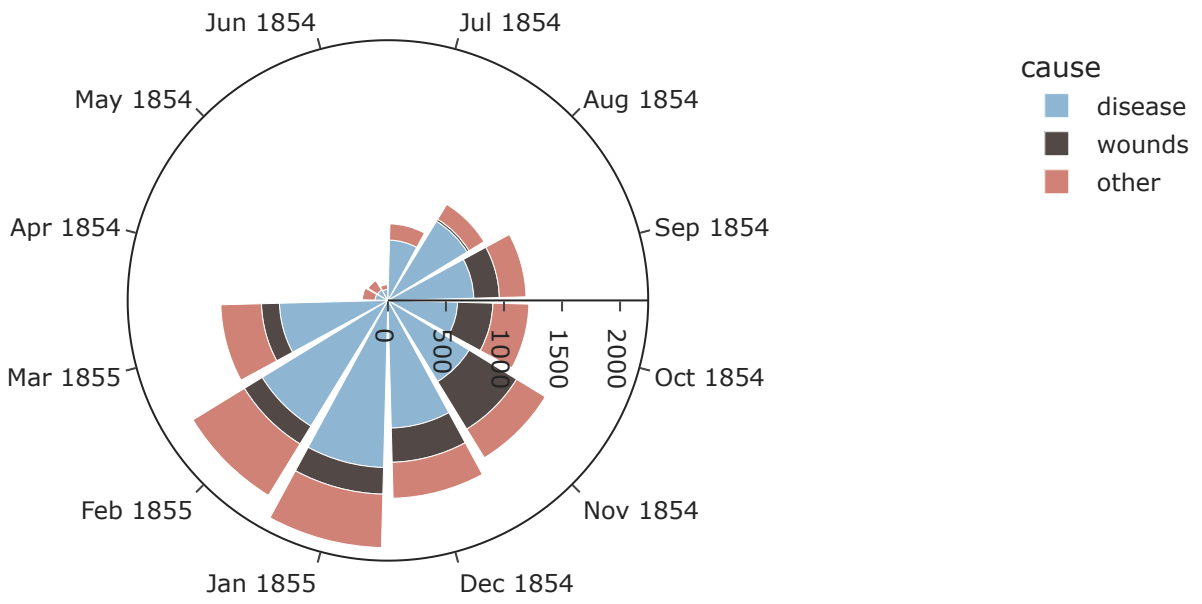


Optei por não reproduzir os gráficos tal quais a visualização original em alguns detalhes, como a inversão da ordem das causas em meses específicos, pois além de dificuldade adicional da programação, tais detalhes não contribuem tanto na compreensão da visualização em si.

(4) Modificações Propostas

O tipo de visualização escolhida por Nightingale, apesar de muito agradável visualmente, não consegue representar com precisão os dados, por exemplo, se invertermos a ordem dos dados o gráfico, parece que trocamos os valores numéricos dele:

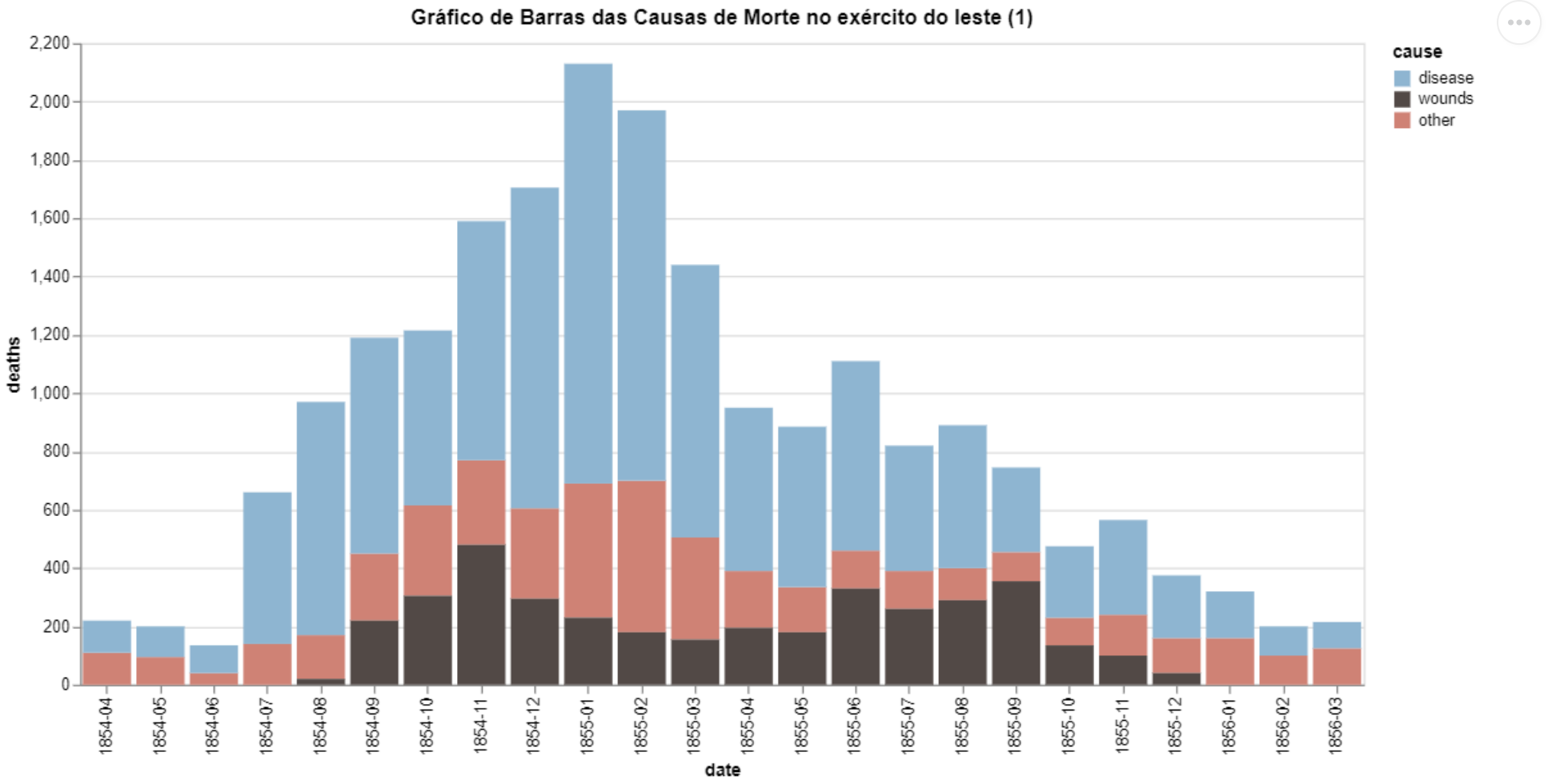
In [5]:
```
causes.reverse()
colors.reverse()

df_aux = pd.melt(
        df_crimea[:12],
        id_vars = ['date'],
        value_vars = causes,
        var_name = 'cause', value_name = 'deaths'
        )

fig1 = px.bar_polar(
    df_aux,
    r = 'deaths',
    color = 'cause',
    theta = 'date',
    start_angle = 165,
    color_discrete_sequence = colors,
    template = 'simple_white', width = 800, height = 400
)

fig1.show()
```



Além disso, os meses que tem valores pequenos, como abril, maio e junho de 1854, são quase que invisíveis no gráfico.

Uma modificação possível seria trocar o gráfico de barras em coordenadas polares por um gráfico de barras comum:

In [71]:
```
scale = alt.Scale(domain = ['disease', 'wounds', 'other'], range = colors)
color = alt.Color('cause', scale = scale)

bar1 = alt.Chart(df_melted).mark_bar().encode(
    x = 'date',
    y = 'deaths',
    color = color
).properties(width = 800, height = 400, title="Gráfico de Barras das Causas de Morte no exército do leste (1)")

bar1.save('bar1.html')
bar1.display()
```

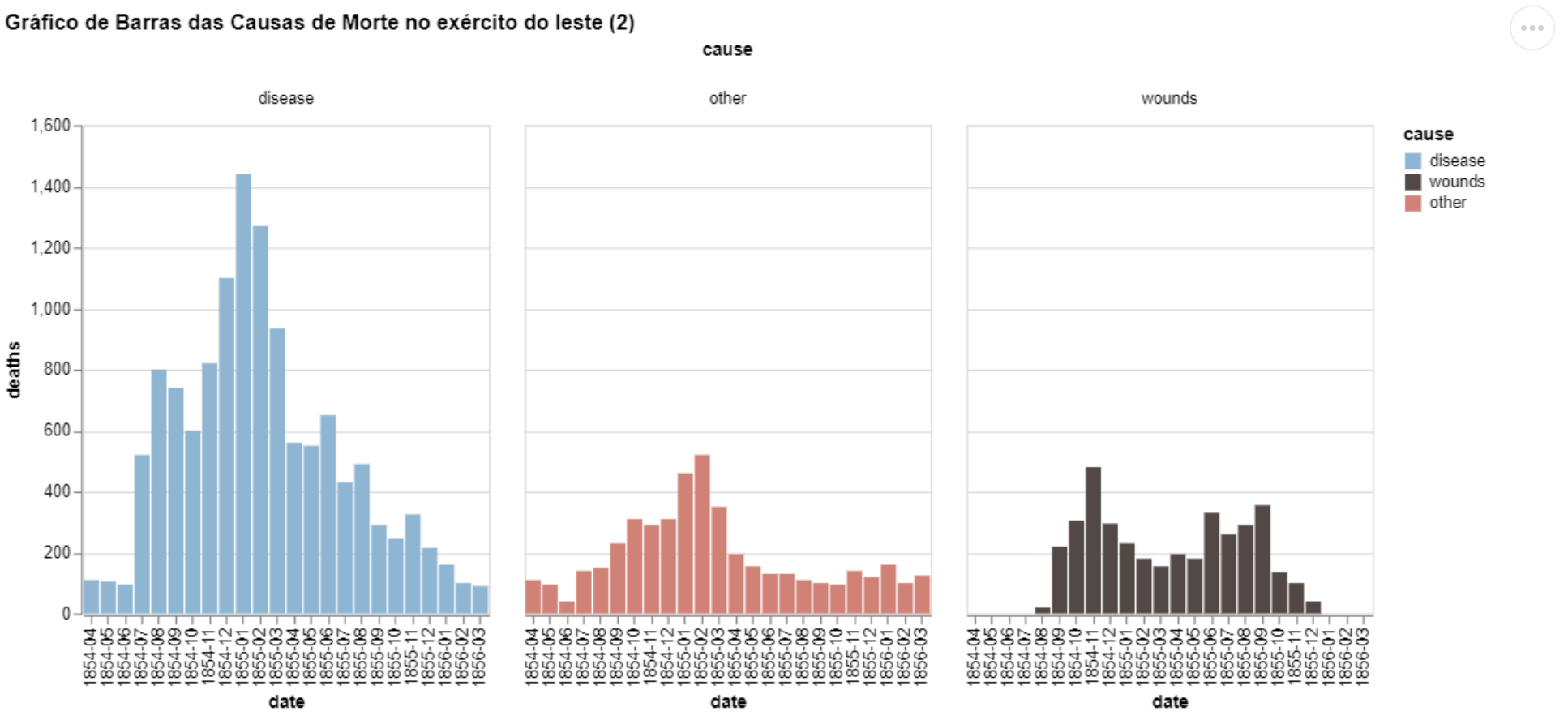


Com este gráfico a análise fica mais fácil, tornando possível perceber que alguns meses não tem mortes por ferimentos (o que em alguns casos não ficava claro na versão radial) e também é possível observar a redução no número de mortes ao longo do tempo. Além disso optei por unir os dois gráficos de modo a apresentar os dados como um todo, já que a separação por ano não faz tanto sentido agora que a compactação dos dados não é um problema como no bar plot radial.

Outra opção, ao invés do gráfico stacked, seria separar as barras por categoria para podermos observá-las separadamente:

In [72]:

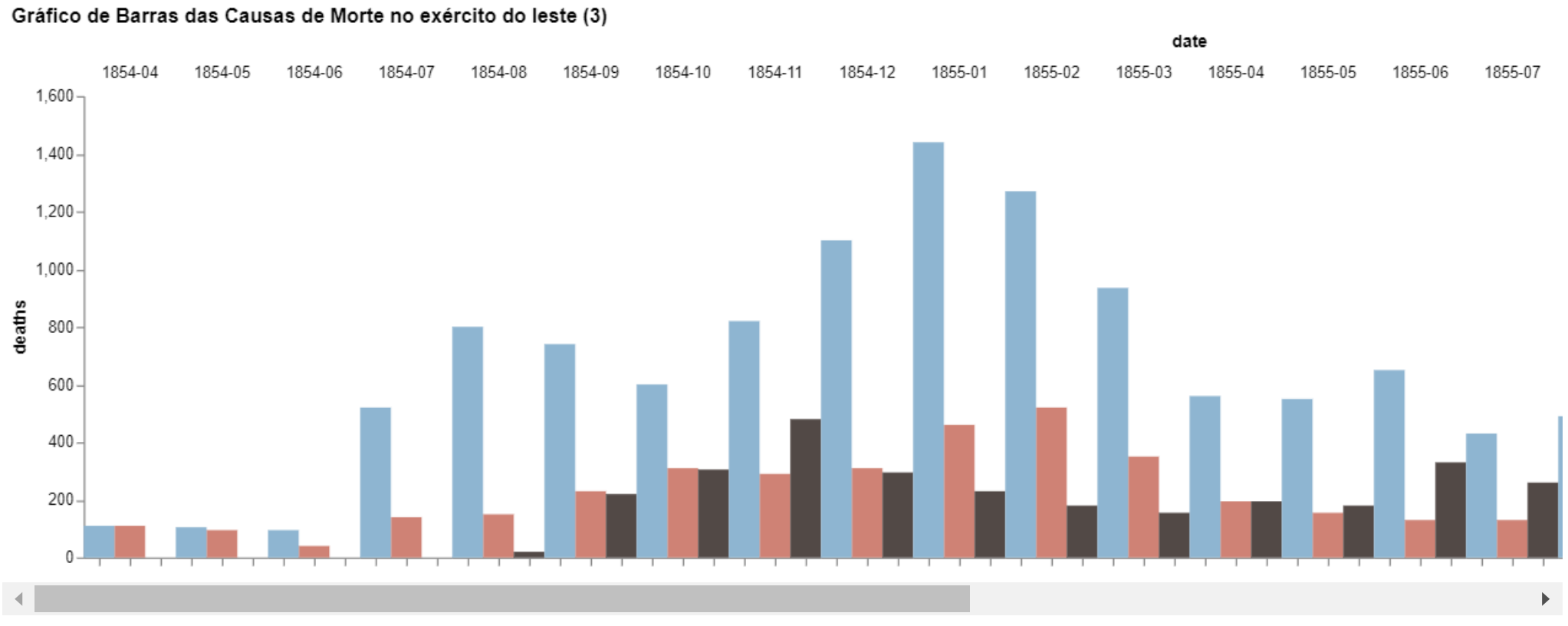
```
bar2 = alt.Chart(df_melted).mark_bar().encode(
    x = 'date',
    y = 'deaths',
    column = 'cause',
    color = color
).properties(width=250, title = "Gráfico de Barras das Causas de Morte no exército do leste (2)")

save(bar2, "bar2.html")

bar2.display()
```



In [74]:

```
bar3 = alt.Chart(df_melted).mark_bar(
    width=20
).encode(
    x = alt.X('cause', axis = alt.Axis(labels=False, title=None, grid=False)),
    y = 'deaths',
    #column = 'date',
    color = alt.Color('cause', scale = scale)
).facet(
    'date', spacing = 0
).configure_axis(
    grid=False
).configure_view(
    strokeWidth = 0
).properties(title = "Gráfico de Barras das Causas de Morte no exército do leste (3)")

save(bar3, "bar3.html")
```

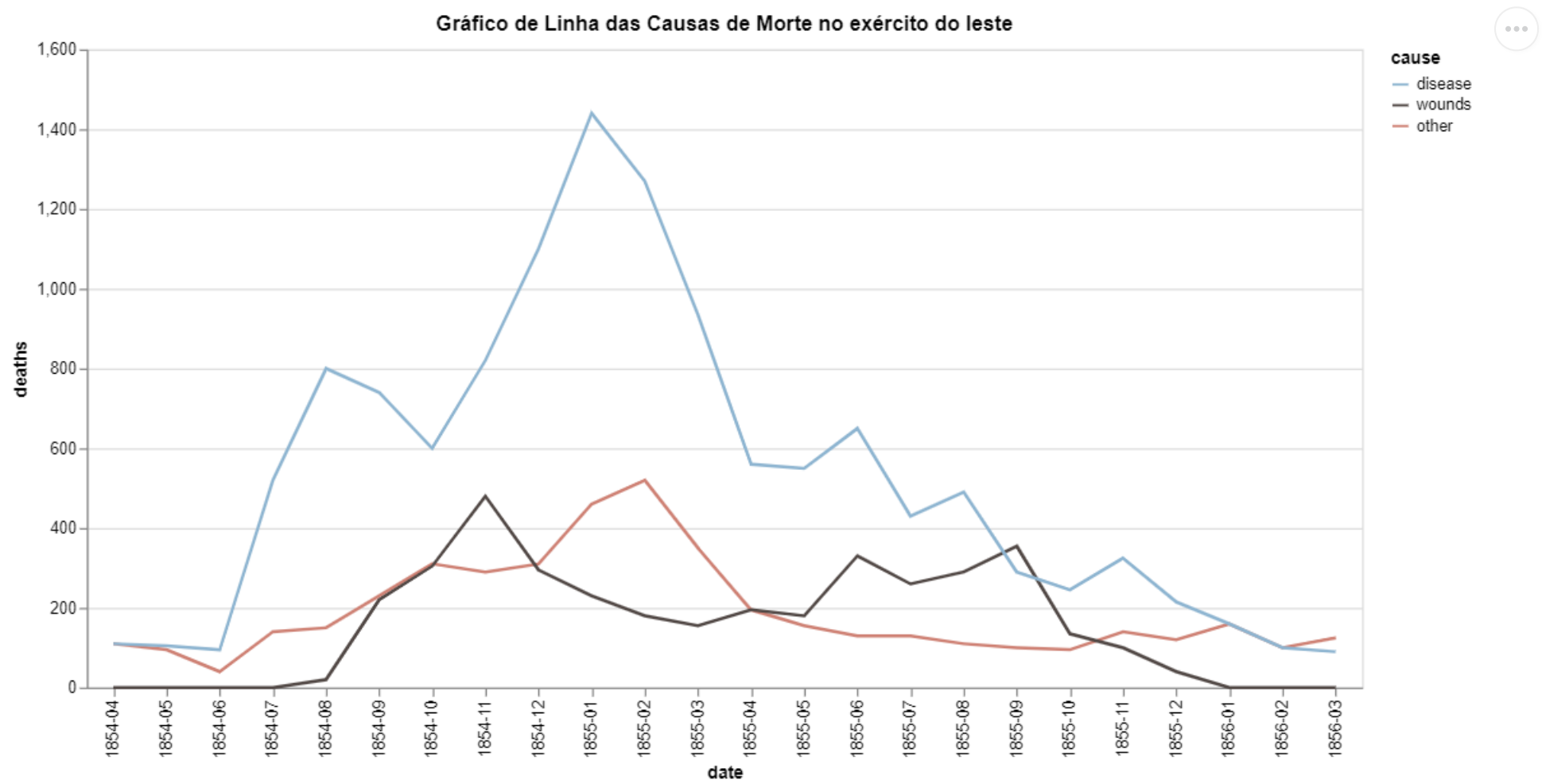
```
bar3.display()
```



Com as visualizações em barra, é possível entender melhor a diferença entre a quantidade de mortes por doença e a quantidade de mortes por ferimentos e outras causas, já que a distância vertical é uma métrica melhor para a compreensão de quantidade do que áreas e transformar as barras em um gráfico reto retira a ambiguidade que um gráfico radial ("redondo") pode gerar como exemplificado com a troca de ordem das causas de morte no primeiro gráfico da sessão 3.

Para observar a mudança ao longo do tempo na quantidade de mortes, um gráfico de linhas seria mais adequado:

In [9]:
```
line = alt.Chart(df_melted).mark_line().encode(
    x = 'date',
    y = 'deaths',
    color = color
).properties(width = 800, height = 400, title="Gráfico de Linha das Causas de Morte no exército do leste")

save(line, "line.html")

line.display()
```



Este último gráfico evidencia a diminuição dramática nas mortes por doença depois da aplicação de melhores condições sanitárias nos hospitais de guerra e no campo de batalhano segundo ano, o que o gráfico original de Florence não mostra tão bem já que ele está "normalizado", além de ter algumas inversões na ordem das categorias.

Caso alguma imagem não tenha ficado boa no pdf ou cortada, exportei todas para .png e .html e upei para o meu github: https://github.com/GDevigili/information-visualization-homeworks/tree/main/trabalho_2/png

In [ ]:
```
```