

CH05_Nightingales_Rose_Python_Supplement

September 3, 2026

1 Chapter 5 — Nightingale’s Rose (Python supplement)

Condensed notebook for the **Python** / **Plotly** portion of *Chapter 5: Nightingale’s Rose Data*.

Target graphics:

- Polar bar charts: area-from-center, stacked, and radius-from-center (**Figures 5.48–5.52**)
- Plotly Express polar bar chart (**Figure 5.53**)
- Rectangular charts: grouped/faceted/stacked bars, areas, and aggregated means (**Figures 5.54–5.60**)

Data: `Nightingale.csv` in Data For Condensed Notebooks.

Dependencies:

- `pandas`
- `plotly`
- `numpy`

1.1 Imports and display options

Import **pandas**, **plotly.express**, and **numpy**, and set Plotly output plus pandas options as in the chapter.

```
[1]: import pandas as pd
import plotly.express as px
import numpy as np
from IPython.display import Image
```

```
[2]: # Set output options.
import plotly.io as pio
pio.renderers.default = "pdf+jupyterlab+notebook"
```

```
[3]: # Set the maximum number of DataFrame rows to display
pd.options.display.max_rows = 8
```

1.2 Loading and organizing the data

Read `Nightingale.csv` from Data For Condensed Notebooks.

```
[4]: df = pd.read_csv('../Data For Condensed Notebooks/Nightingale.csv')
df
```

```
[4]:      Unnamed: 0      Date Month  Year  Army  Disease  Wounds  Other  \
0             1  1854-04-01  Apr  1854   8571         1         0         5
1             2  1854-05-01  May  1854  23333        12         0         9
2             3  1854-06-01  Jun  1854  28333        11         0         6
3             4  1854-07-01  Jul  1854  28722       359         0        23
..          ...      ...      ...      ...      ...      ...      ...
20           21  1855-12-01  Dec  1855  43217        91        18        28
21           22  1856-01-01  Jan  1856  44212        42         2        48
22           23  1856-02-01  Feb  1856  43485        24         0        19
23           24  1856-03-01  Mar  1856  46140        15         0        35
```

```
      Disease.rate  Wounds.rate  Other.rate
0             1.4           0.0           7.0
1             6.2           0.0           4.6
2             4.7           0.0           2.5
3          150.0           0.0           9.6
..          ...           ...           ...
20           25.3           5.0           7.8
21           11.4           0.5          13.0
22            6.6           0.0           5.2
23            3.9           0.0           9.1
```

[24 rows x 11 columns]

Drop the index column and the Date column; keep month, year, army counts, and mortality rates.

```
[5]: df.drop(columns = ['Unnamed: 0', 'Date'], inplace = True)
df
```

```
[5]:      Month  Year  Army  Disease  Wounds  Other  Disease.rate  Wounds.rate  \
0     Apr  1854   8571         1         0         5           1.4           0.0
1     May  1854  23333        12         0         9           6.2           0.0
2     Jun  1854  28333        11         0         6           4.7           0.0
3     Jul  1854  28722       359         0        23          150.0           0.0
..    ...    ...    ...      ...      ...      ...      ...
20   Dec  1855  43217        91        18        28          25.3           5.0
21   Jan  1856  44212        42         2        48          11.4           0.5
22   Feb  1856  43485        24         0        19           6.6           0.0
23   Mar  1856  46140        15         0        35           3.9           0.0
```

```
      Other.rate
0           7.0
1           4.6
2           2.5
3           9.6
```

```

..      ...
20      7.8
21     13.0
22      5.2
23      9.1

```

[24 rows x 9 columns]

Keep only columns needed for plotting and rename the `.rate` fields to readable cause labels (**Figure 5.9** in the chapter illustrates similar relabeling in Excel).

```

[6]: rates = df.drop(columns = ['Disease', 'Wounds', 'Other'])
rates.rename(columns = {'Disease.rate': 'Preventable diseases', 'Wounds.rate': 'Wounds & injuries', 'Other.rate': 'All other causes'}, inplace = True)
rates

```

```

[6]:   Month  Year  Army  Preventable diseases  Wounds & injuries \
0   Apr  1854   8571                1.4                0.0
1   May  1854  23333                6.2                0.0
2   Jun  1854  28333                4.7                0.0
3   Jul  1854  28722             150.0                0.0
..  ...  ...  ...
20  Dec  1855  43217                25.3                5.0
21  Jan  1856  44212                11.4                0.5
22  Feb  1856  43485                 6.6                0.0
23  Mar  1856  46140                 3.9                0.0

```

```

All other causes
0                7.0
1                4.6
2                2.5
3                9.6
..              ...
20               7.8
21             13.0
22               5.2
23               9.1

```

[24 rows x 6 columns]

Three scaling choices: (1) radius equals the death rate; (2) sector **area** equals the rate (Nightingale's approach); (3) non-overlapping stacked polar rectangles (polar histogram). The chapter walks through the geometry; we store resulting radii in a multi-index DataFrame `r`.

Build empty `r` with rows aligned to `rates` (see chapter text on `range(len(rates))`).

```

[7]: columns = pd.MultiIndex.from_product([('Stacked Histogram', 'Radius from Center', 'Area from Center'),
                                          ('r_wi', 'r_aoc', 'r_pd')])

```

```
r = pd.DataFrame(index=range(len(rates)), columns=columns)
r
```

```
[7]:      Stacked Histogram      Radius from Center      \
      r_wi r_aoc r_pd      r_wi r_aoc r_pd
0      NaN  NaN  NaN      NaN  NaN  NaN
1      NaN  NaN  NaN      NaN  NaN  NaN
2      NaN  NaN  NaN      NaN  NaN  NaN
3      NaN  NaN  NaN      NaN  NaN  NaN
..      ...  ...  ...      ...  ...  ...
20     NaN  NaN  NaN      NaN  NaN  NaN
21     NaN  NaN  NaN      NaN  NaN  NaN
22     NaN  NaN  NaN      NaN  NaN  NaN
23     NaN  NaN  NaN      NaN  NaN  NaN

      Area from Center
      r_wi r_aoc r_pd
0      NaN  NaN  NaN
1      NaN  NaN  NaN
2      NaN  NaN  NaN
3      NaN  NaN  NaN
..      ...  ...  ...
20     NaN  NaN  NaN
21     NaN  NaN  NaN
22     NaN  NaN  NaN
23     NaN  NaN  NaN

[24 rows x 9 columns]
```

Radius from center: use the reported rates directly as radial extent.

```
[8]: # Radius from the center version
r.loc[:, ('Radius from Center', 'r_wi')] = rates['Wounds & injuries']
r.loc[:, ('Radius from Center', 'r_aoc')] = rates['All other causes']
r.loc[:, ('Radius from Center', 'r_pd')] = rates['Preventable diseases']
```

Shorthand m_0, m_1, m_2 for the three cause-specific rates (wounds, other, disease).

```
[9]: # Set shorthand notation to simplify formulas.
m0 = rates['Wounds & injuries']
m1 = rates['All other causes']
m2 = rates['Preventable diseases']
```

Area from center: outer radius $r = \sqrt{m \cdot 12 / \pi}$ so sector area matches m for a 30° slice.

```
[10]: # Area of slice from the center version
r.loc[:, ('Area from Center', 'r_wi')] = np.sqrt(m0 * (12 / np.pi))
r.loc[:, ('Area from Center', 'r_aoc')] = np.sqrt(m1 * (12 / np.pi))
r.loc[:, ('Area from Center', 'r_pd')] = np.sqrt(m2 * (12 / np.pi))
```

Stacked histogram: polar rectangles stacked by cumulative area (chapter formulas).

```
[11]: # Stacked histogram version
r.loc[:, ('Stacked Histogram', 'r_wi')] = np.sqrt(m0 * (12 / np.pi))
r.loc[:, ('Stacked Histogram', 'r_aoc')] = np.sqrt((m1 + m0) * (12 / np.pi))
r.loc[:, ('Stacked Histogram', 'r_pd')] = np.sqrt((m2 + m1 + m0) * (12 / np.pi))
```

Inspect r: stacked radii should increase from r_wi to r_aoc to r_pd each month.

```
[12]: r
```

```
[12]:      Stacked Histogram      Radius from Center      \
      r_wi      r_aoc      r_pd      r_wi r_aoc      r_pd
0          0.0  5.170883  5.664418          0.0  7.0      1.4
1          0.0  4.191743  6.422847          0.0  4.6      6.2
2          0.0  3.090194  5.244232          0.0  2.5      4.7
3          0.0  6.055518 24.690628          0.0  9.6     150.0
..          ...          ...          ...          ...
20         4.370194   6.99231  12.063635          5.0   7.8     25.3
21         1.381977   7.180961   9.752487          0.5  13.0     11.4
22          0.0  4.456741   6.713619          0.0   5.2      6.6
23          0.0  5.895714   7.046726          0.0   9.1      3.9

      Area from Center
      r_wi      r_aoc      r_pd
0          0.0  5.170883  2.312489
1          0.0  4.191743  4.866442
2          0.0  3.090194  4.23706
3          0.0  6.055518 23.936537
..          ...          ...          ...
20         4.370194   5.45837   9.830508
21         1.381977   7.046726   6.598848
22          0.0  4.456741   5.02097
23          0.0  5.895714   3.859651
```

[24 rows x 9 columns]

Add Month-Year labels (coerce Year to string for concatenation).

```
[13]: r['Month-Year'] = rates['Month'] + " " + rates['Year'].astype(str)
r
```

```
[13]:      Stacked Histogram      Radius from Center      \
      r_wi      r_aoc      r_pd      r_wi r_aoc      r_pd
0          0.0  5.170883  5.664418          0.0  7.0      1.4
1          0.0  4.191743  6.422847          0.0  4.6      6.2
2          0.0  3.090194  5.244232          0.0  2.5      4.7
3          0.0  6.055518 24.690628          0.0  9.6     150.0
..          ...          ...          ...          ...
20         4.370194   6.99231  12.063635          5.0   7.8     25.3
21         1.381977   7.180961   9.752487          0.5  13.0     11.4
22          0.0  4.456741   6.713619          0.0   5.2      6.6
23          0.0  5.895714   7.046726          0.0   9.1      3.9
```

20	4.370194	6.99231	12.063635	5.0	7.8	25.3
21	1.381977	7.180961	9.752487	0.5	13.0	11.4
22	0.0	4.456741	6.713619	0.0	5.2	6.6
23	0.0	5.895714	7.046726	0.0	9.1	3.9

	Area from Center			Month-Year
	r_wi	r_aoc	r_pd	
0	0.0	5.170883	2.312489	Apr 1854
1	0.0	4.191743	4.866442	May 1854
2	0.0	3.090194	4.23706	Jun 1854
3	0.0	6.055518	23.936537	Jul 1854
..	...	...	...	...
20	4.370194	5.45837	9.830508	Dec 1855
21	1.381977	7.046726	6.598848	Jan 1856
22	0.0	4.456741	5.02097	Feb 1856
23	0.0	5.895714	3.859651	Mar 1856

[24 rows x 10 columns]

Set Month-Year as the index for polar tick labels.

```
[14]: r.set_index('Month-Year', inplace=True)
r
```

```
[14]:
```

	Stacked Histogram			Radius from Center		\
	r_wi	r_aoc	r_pd	r_wi	r_aoc	
Month-Year						
Apr 1854	0.0	5.170883	5.664418	0.0	7.0	
May 1854	0.0	4.191743	6.422847	0.0	4.6	
Jun 1854	0.0	3.090194	5.244232	0.0	2.5	
Jul 1854	0.0	6.055518	24.690628	0.0	9.6	
...	...	...	...	...	...	
Dec 1855	4.370194	6.99231	12.063635	5.0	7.8	
Jan 1856	1.381977	7.180961	9.752487	0.5	13.0	
Feb 1856	0.0	4.456741	6.713619	0.0	5.2	
Mar 1856	0.0	5.895714	7.046726	0.0	9.1	

	Area from Center			
	r_pd	r_wi	r_aoc	r_pd
Month-Year				
Apr 1854	1.4	0.0	5.170883	2.312489
May 1854	6.2	0.0	4.191743	4.866442
Jun 1854	4.7	0.0	3.090194	4.23706
Jul 1854	150.0	0.0	6.055518	23.936537
...	...	...	...	...
Dec 1855	25.3	4.370194	5.45837	9.830508
Jan 1856	11.4	1.381977	7.046726	6.598848
Feb 1856	6.6	0.0	4.456741	5.02097

Mar 1856 3.9 0.0 5.895714 3.859651

[24 rows x 9 columns]

1.3 Graphics with Plotly Graph Objects

MakeBarPolarGraphic builds one polar bar figure for a **timeframe** (before = first 12 months, after = second 12) and a **version** (Area from Center, Radius from Center, or Stacked Histogram). Example outputs correspond to **Figures 5.48–5.51** in the chapter.

```
[15]: # Get access to graphics objects.
import plotly.graph_objects as go

def MakeBarPolarGraphic(timeframe='before', version='Area from Center'):
    # Choose the right set of months based on the 'timeframe' argument.
    if timeframe == 'before':
        r_data = r[:12]
    elif timeframe == 'after':
        r_data = r[12:]
    else:
        print('Bad timeframe.')
        return 0

    # Pick out the appropriate r values based on the 'version' argument.
    r_vals = r_data[version]

    # Center the polar rectangles on the angular axis tick marks.
    offset = -0.5

    fig = go.Figure()

    fig.add_trace(go.Barpolar(
        r=r_vals['r_pd'],
        name='Preventable Diseases',
        marker_color='lightblue',
        legendgroup='pd'
    ))
    fig.add_trace(go.Barpolar(
        r=r_vals['r_aoc'],
        name='All other causes',
        marker_color='darkgray',
        legendgroup='aoc',
    ))
    fig.add_trace(go.Barpolar(
        r=r_vals['r_wi'],
        name='Wounds & injuries',
```

```

        marker_color='pink',
        legendgroup='wi'
    ))

    fig.update_traces(text=list(r_data.index.values))
    fig.update_traces(theta=list(r_data.index.values))
    fig.update_traces(offset=offset)
    fig.update_traces(base=0)

    if version != 'Stacked Histogram':
        fig.update_traces(opacity=0.8)
        fig.update_layout(barmode='overlay')
        fig.update_traces(marker_pattern_shape='x', selector=({'name': 'All_
↪other causes'}))

    fig.update_layout(
        legend_font_size=14,
        polar_angularaxis_rotation=170,
        polar_angularaxis_direction='clockwise',
        polar_radialaxis_visible=False,
        polar_bargap=0,
        height=600
    )

    return fig

```

Figure 5.48 — Before sanitation, area-from-center version.

```
[16]: MakeBarPolarGraphic(timeframe='before', version='Area from Center')
```

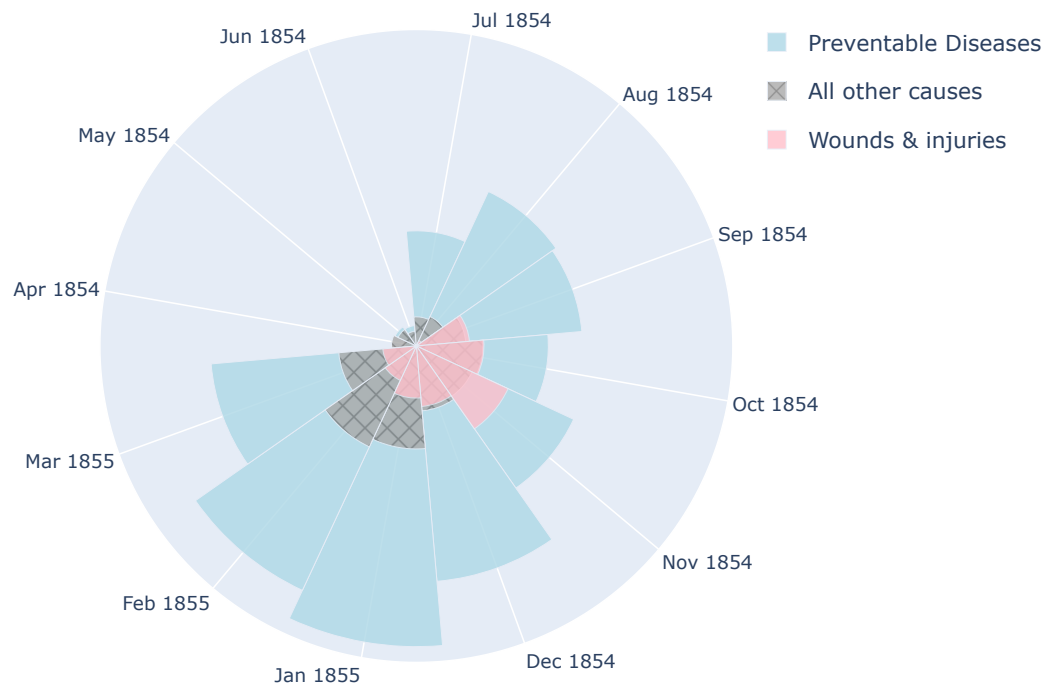


Figure 5.49 — After sanitation, area-from-center version.

```
[17]: MakeBarPolarGraphic(timeframe='after', version='Area from Center')
```

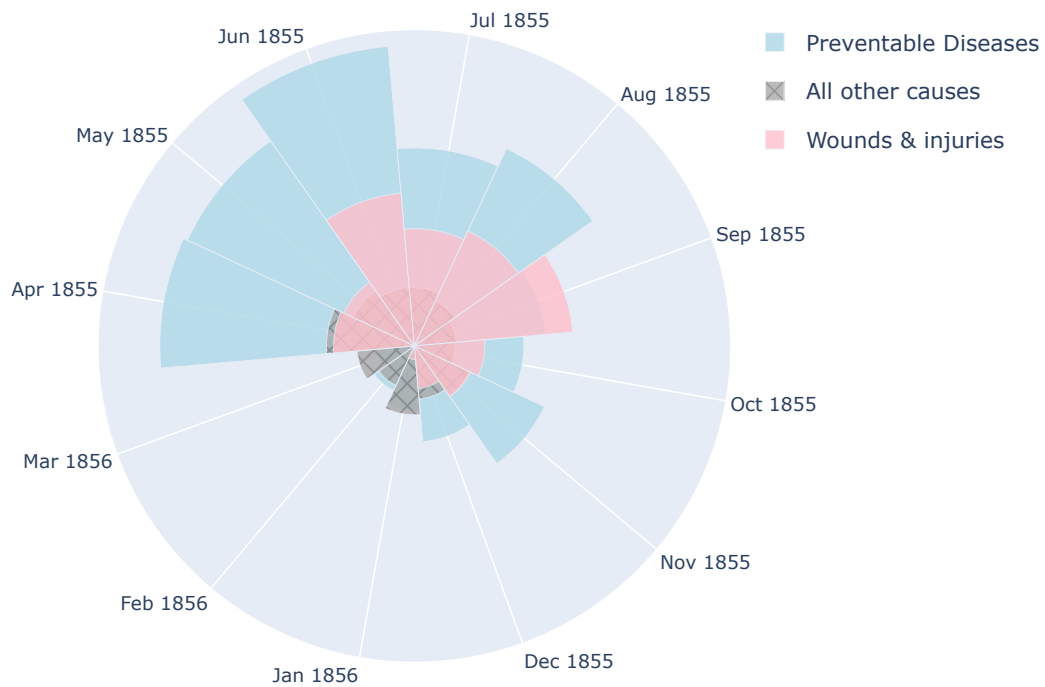


Figure 5.50 — Before sanitation, stacked polar histogram.

```
[18]: MakeBarPolarGraphic(timeframe='before', version='Stacked Histogram')
```

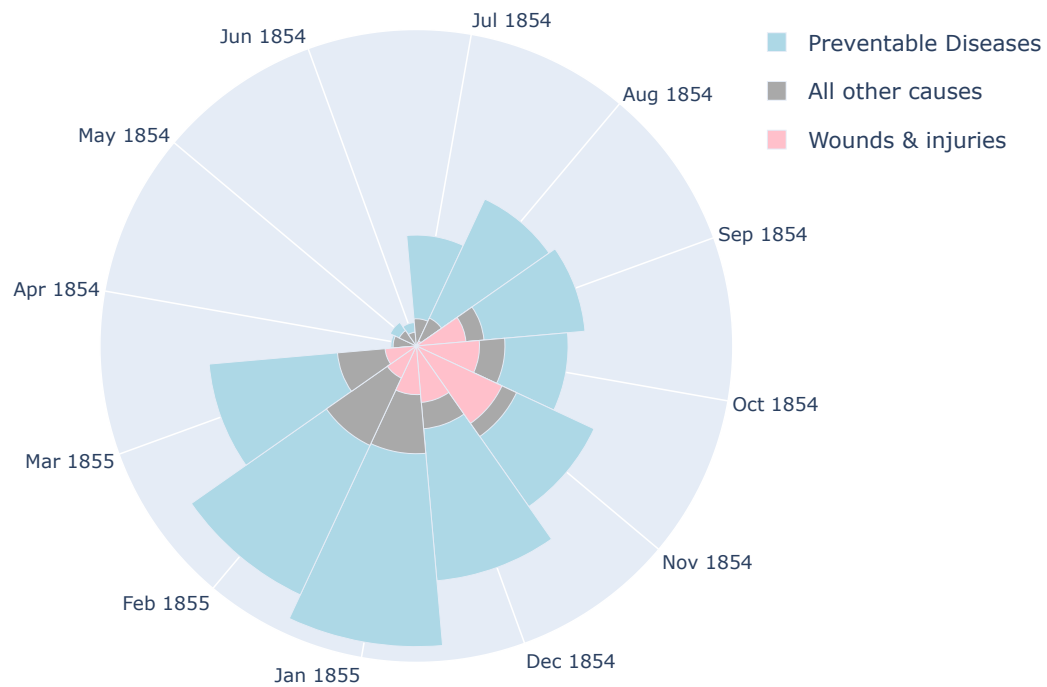
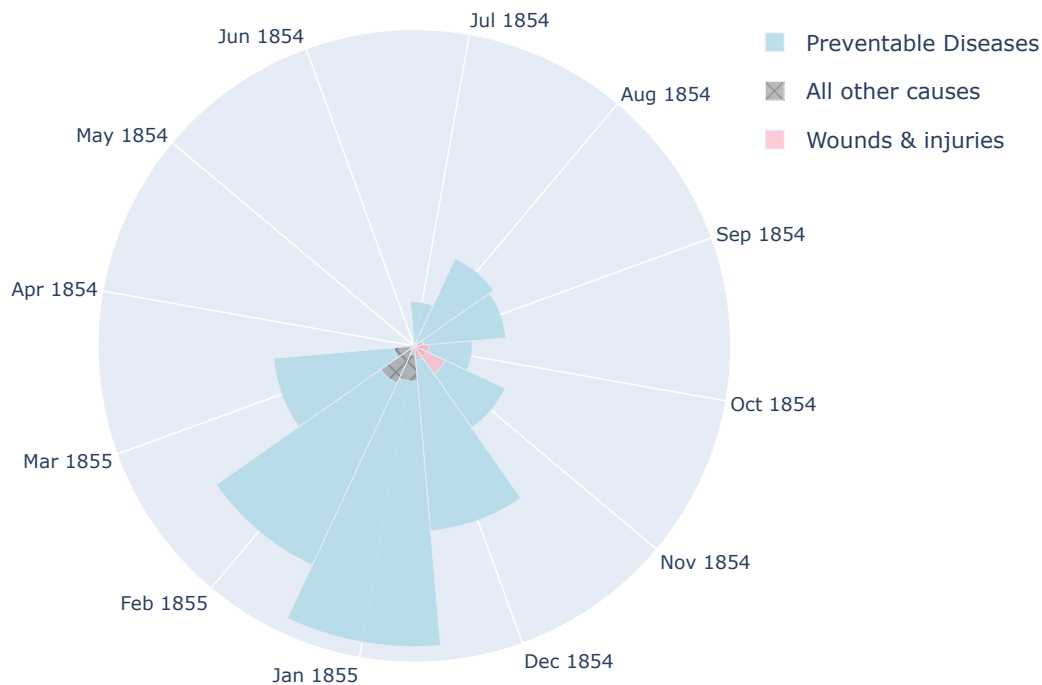


Figure 5.51 — Radius-from-center (misleading area perception; chapter discusses why).

[19]: `MakeBarPolarGraphic(timeframe='before', version='Radius from Center')`



Area does not scale linearly with radius, so radius-from-center distorts comparative areas.

For the side-by-side before/after comparison (**Figure 5.52**), the chapter uses the **stacked histogram** version and a shared radial range so both roses are comparable.

```
[20]: # Import the command needed to create a figure with subplots.
from plotly.subplots import make_subplots

fig_before = MakeBarPolarGraphic(timeframe='before', version='Stacked_
    ↪Histogram')
fig_after = MakeBarPolarGraphic(timeframe='after', version='Stacked Histogram')

fig = make_subplots(rows=1, cols=2,
                    specs=[[{"type": "polar"}, {"type": "polar"}]],
                    subplot_titles=('April 1854 to March 1855', 'April 1855 to_
    ↪March 1856'))

fig_before.update_traces(showlegend=False)
```

```

for trace in fig_before.data:
    fig.add_trace(trace, row=1, col=1)

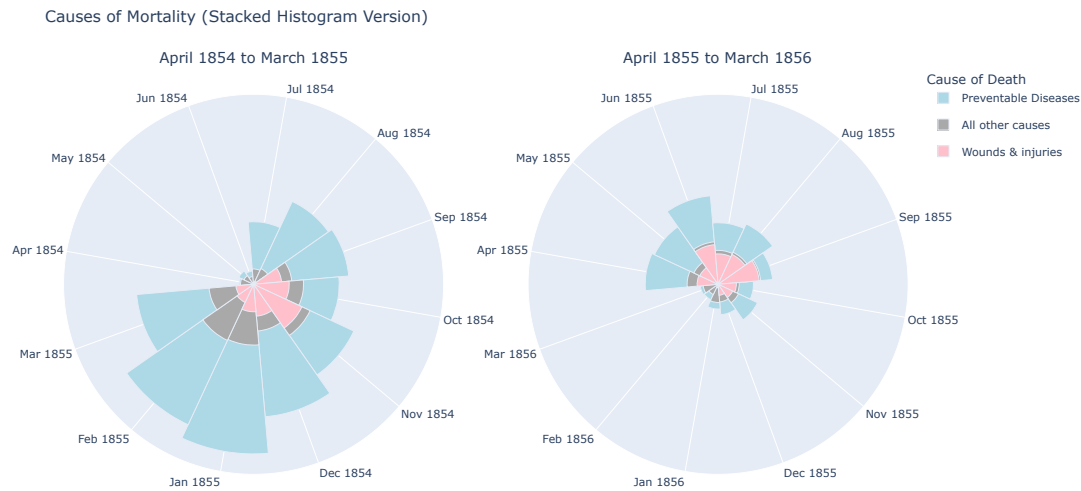
for trace in fig_after.data:
    fig.add_trace(trace, row=1, col=2)

fig.update_layout(height=650,
                  width=1200,
                  title_text="Causes of Mortality (Stacked Histogram Version)")
fig.layout['legend']['title'] = 'Cause of Death'

fig.update_layout(
    polar=dict(
        radialaxis_visible=False,
        angularaxis_direction='clockwise',
        angularaxis_rotation=170,
        radialaxis_range=[0, 75],
        bargap=0
    ),
    polar2=dict(
        radialaxis_visible=False,
        angularaxis_direction='clockwise',
        angularaxis_rotation=170,
        radialaxis_range=[0, 75],
        bargap=0
    )
)

fig

```



Formatting notes (see chapter): trace order matters for overlays; `polar` / `polar2` subplot layouts must be set after adding traces; `radialaxis_range=[0, 75]` keeps scales aligned across panels.

1.4 Plotly Express polar bar

Long-form data via `melt`, then `px.bar_polar` (**Figure 5.53**). `base=np.zeros(...)` keeps sectors starting at the origin while using overlay mode.

```
[21]: r_before = r[:12]['Area from Center'].reset_index()
      r_before
```

```
[21]:
```

	Month-Year	r_wi	r_aoc	r_pd
0	Apr 1854	0.0	5.170883	2.312489
1	May 1854	0.0	4.191743	4.866442
2	Jun 1854	0.0	3.090194	4.23706
3	Jul 1854	0.0	6.055518	23.936537
..	...	...	...	...
8	Dec 1854	12.620708	13.54055	49.113667
9	Jan 1855	10.828913	21.409489	62.504466
10	Feb 1855	7.89059	23.133149	56.061257
11	Mar 1855	6.99231	16.187424	42.832358

[12 rows x 4 columns]

```
[22]: r_before_long = r_before.melt(value_vars=['r_wi', 'r_aoc', 'r_pd'],
                                   id_vars='Month-Year',
                                   value_name='r',
                                   var_name='Cause of Death')
```

```
r_before_long
```

```
[22]:
```

	Month-Year	Cause of Death	r
0	Apr 1854	r_wi	0.0
1	May 1854	r_wi	0.0
2	Jun 1854	r_wi	0.0
3	Jul 1854	r_wi	0.0
..	...	...	...
32	Dec 1854	r_pd	49.113667
33	Jan 1855	r_pd	62.504466
34	Feb 1855	r_pd	56.061257
35	Mar 1855	r_pd	42.832358

```
[36 rows x 3 columns]
```

```
[23]: r_before_long['Cause of Death'] = r_before_long['Cause of Death'].replace({
    'r_pd': 'Preventable diseases',
    'r_wi': 'Wounds & injuries',
    'r_aoc': 'All other causes'})
r_before_long
```

```
[23]:
```

	Month-Year	Cause of Death	r
0	Apr 1854	Wounds & injuries	0.0
1	May 1854	Wounds & injuries	0.0
2	Jun 1854	Wounds & injuries	0.0
3	Jul 1854	Wounds & injuries	0.0
..	...	...	...
32	Dec 1854	Preventable diseases	49.113667
33	Jan 1855	Preventable diseases	62.504466
34	Feb 1855	Preventable diseases	56.061257
35	Mar 1855	Preventable diseases	42.832358

```
[36 rows x 3 columns]
```

```
[24]: category_orders = {'Cause of Death': ['Preventable diseases',
    'All other causes',
    'Wounds & injuries']}

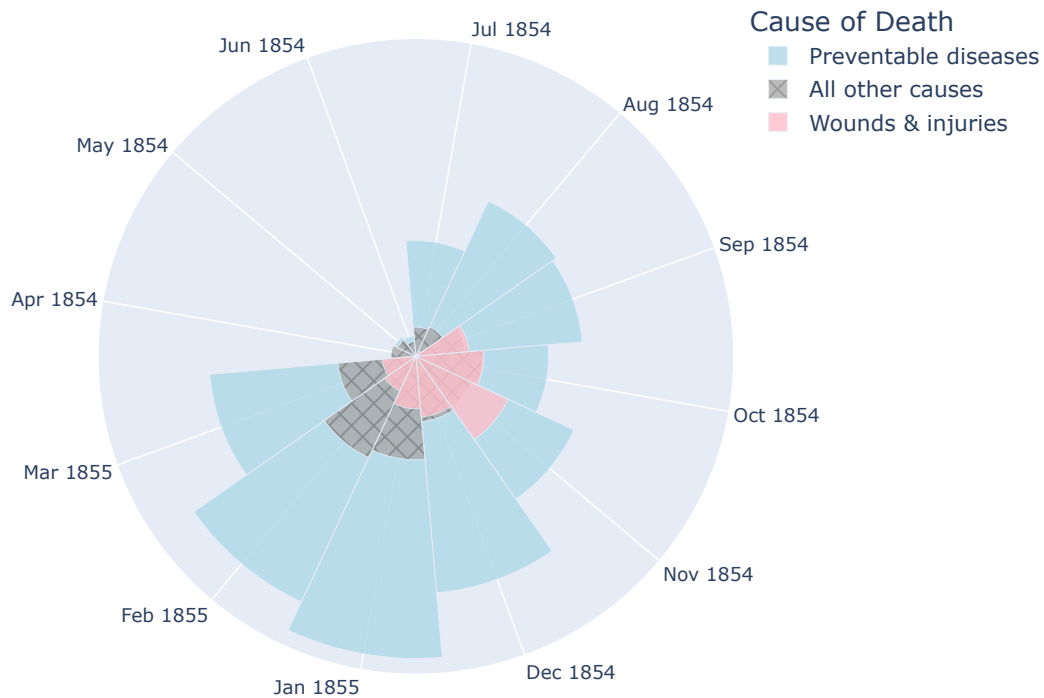
fig = px.bar_polar(r_before_long, r='r', theta='Month-Year',
    color='Cause of Death',
    color_discrete_sequence=['lightblue', 'darkgray', 'pink'],
    pattern_shape='Cause of Death',
    pattern_shape_sequence=['', 'x', ''],
    category_orders=category_orders,
    barmode='overlay',
    base=np.zeros(len(r_before_long)))
```

```
fig.update_traces(opacity=0.8)

fig.update_layout(
    title='April 1854 to March 1855',
    legend_font_size=14,
    polar_angularaxis_rotation=170,
    polar_angularaxis_direction='clockwise',
    polar_radialaxis_visible=False,
    polar_bargap=0,
    height=600
)

fig.show()
```

April 1854 to March 1855



1.5 Rectangular charts (Plotly Express)

Easier to read than polar wedges for many readers (**Figures 5.54–5.60**): grouped bars, facets, stacks, areas, a single timeline with reference line, and aggregated means.

```
[25]: rates.head(10)
```

```
[25]:
```

	Month	Year	Army	Preventable diseases	Wounds & injuries	\
0	Apr	1854	8571	1.4	0.0	
1	May	1854	23333	6.2	0.0	
2	Jun	1854	28333	4.7	0.0	
3	Jul	1854	28722	150.0	0.0	
..	...	...	...	...	...	
6	Oct	1854	30643	197.0	51.7	
7	Nov	1854	29736	340.6	115.8	
8	Dec	1854	32779	631.5	41.7	
9	Jan	1855	32393	1022.8	30.7	

All other causes

0	7.0
1	4.6
2	2.5
3	9.6
..	...
6	50.1
7	42.8
8	48.0
9	120.0

[10 rows x 6 columns]

Tag each row as before or after sanitation (12 months each).

```
[26]: rates['Timeframe'] = 12 * ['before'] + 12 * ['after']
rates.head(10)
```

```
[26]:
```

	Month	Year	Army	Preventable diseases	Wounds & injuries	\
0	Apr	1854	8571	1.4	0.0	
1	May	1854	23333	6.2	0.0	
2	Jun	1854	28333	4.7	0.0	
3	Jul	1854	28722	150.0	0.0	
..	...	...	...	...	...	
6	Oct	1854	30643	197.0	51.7	
7	Nov	1854	29736	340.6	115.8	
8	Dec	1854	32779	631.5	41.7	
9	Jan	1855	32393	1022.8	30.7	

All other causes Timeframe

0	7.0	before
1	4.6	before
2	2.5	before
3	9.6	before
..	...	...

6	50.1	before
7	42.8	before
8	48.0	before
9	120.0	before

[10 rows x 7 columns]

Reuse `r`'s Month-Year index for unique month labels across both years.

```
[27]: rates['Month-Year'] = r.index
rates.head(10)
```

```
[27]:
```

	Month	Year	Army	Preventable diseases	Wounds & injuries \
0	Apr	1854	8571	1.4	0.0
1	May	1854	23333	6.2	0.0
2	Jun	1854	28333	4.7	0.0
3	Jul	1854	28722	150.0	0.0
..	...	...	...	...	...
6	Oct	1854	30643	197.0	51.7
7	Nov	1854	29736	340.6	115.8
8	Dec	1854	32779	631.5	41.7
9	Jan	1855	32393	1022.8	30.7

	All other causes	Timeframe	Month-Year
0	7.0	before	Apr 1854
1	4.6	before	May 1854
2	2.5	before	Jun 1854
3	9.6	before	Jul 1854
..	...	...	...
6	50.1	before	Oct 1854
7	42.8	before	Nov 1854
8	48.0	before	Dec 1854
9	120.0	before	Jan 1855

[10 rows x 8 columns]

Melt to long form for `px.bar` / `px.area`.

```
[28]: rates_long = rates.melt(id_vars=['Timeframe', 'Month', 'Year', 'Army'],
                             value_vars=['Preventable diseases', 'Wounds &
                             injuries', 'All other causes'],
                             var_name='Cause of Death',
                             value_name='Death Rate')
rates_long
```

```
[28]:
```

	Timeframe	Month	Year	Army	Month-Year	Cause of Death	Death Rate
0	before	Apr	1854	8571	Apr 1854	Preventable diseases	1.4

1	before	May	1854	23333	May 1854	Preventable diseases	6.2
2	before	Jun	1854	28333	Jun 1854	Preventable diseases	4.7
3	before	Jul	1854	28722	Jul 1854	Preventable diseases	150.0
..	...	...	...	...	...	...	...
68	after	Dec	1855	43217	Dec 1855	All other causes	7.8
69	after	Jan	1856	44212	Jan 1856	All other causes	13.0
70	after	Feb	1856	43485	Feb 1856	All other causes	5.2
71	after	Mar	1856	46140	Mar 1856	All other causes	9.1

[72 rows x 7 columns]

Figure 5.54 — Grouped bars with pattern by timeframe.

```
[29]: fig = px.bar(rates_long,
                  x='Month',
                  y='Death Rate',
                  barmode='group',
                  color='Cause of Death',
                  pattern_shape='Timeframe',
                  height=500,
                  width=1200)

fig
```

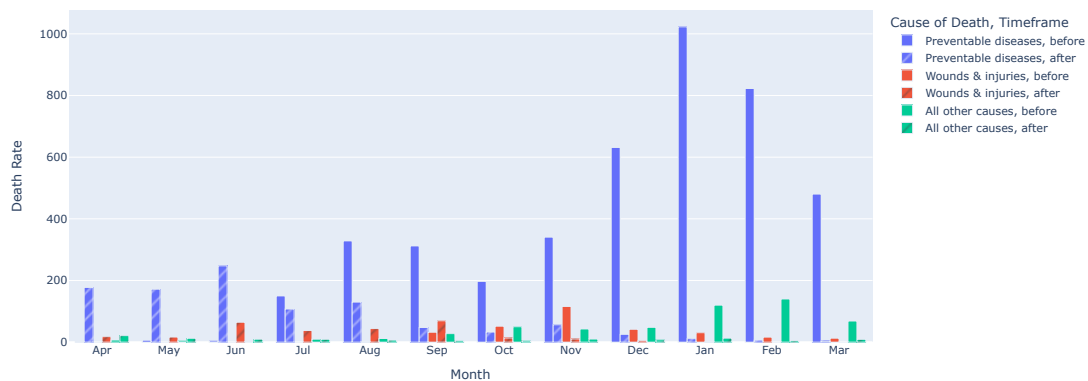


Figure 5.55 — Facet columns by timeframe.

```
[30]: fig = px.bar(rates_long,
                  x='Month',
                  y='Death Rate',
                  barmode='group',
                  color='Cause of Death',
                  facet_col='Timeframe',
                  labels={'Month': ''},
```

```
fig
```

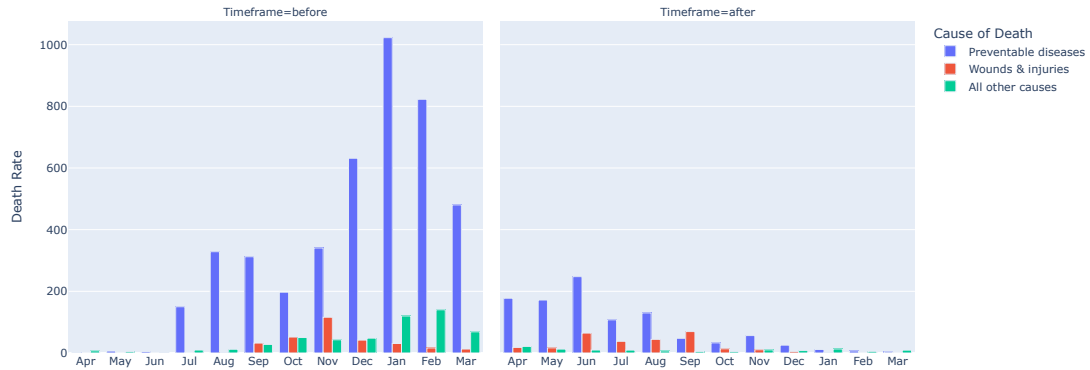


Figure 5.56 — Facet by cause; color encodes timeframe.

```
[31]: fig = px.bar(rates_long,
                  x='Month',
                  y='Death Rate',
                  barmode='group',
                  color='Timeframe',
                  color_discrete_map={'before': 'red', 'after': 'blue'},
                  facet_col='Cause of Death',
                  labels={'Month': ''},
                  height=500,
                  width=1200)

fig
```

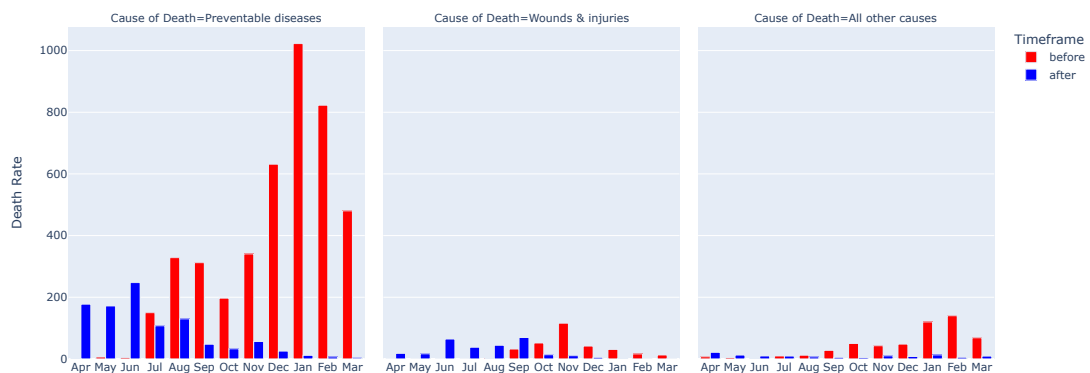


Figure 5.57 — Stacked bars by timeframe.

```
[32]: fig = px.bar(rates_long,
                  x='Month',
                  y='Death Rate',
                  barmode='stack',
                  color='Cause of Death',
                  facet_col='Timeframe',
                  labels={'Month': ''},
                  height=500,
                  width=1200)

fig
```

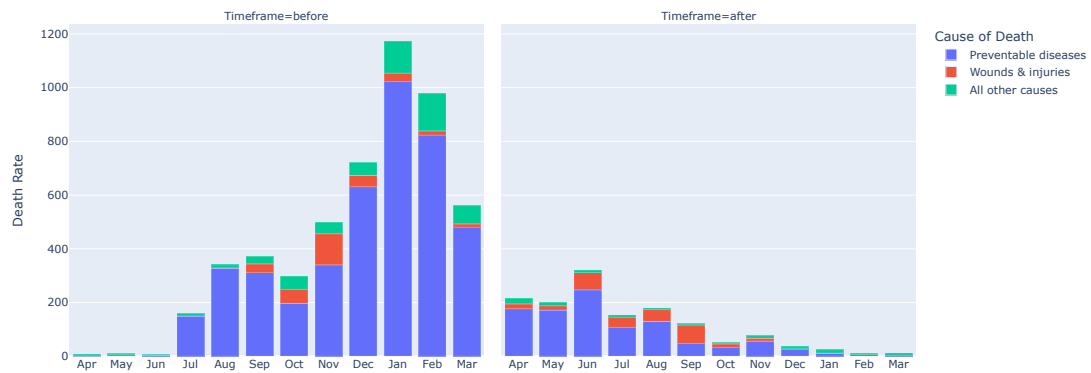


Figure 5.58 — Faceted stacked area charts.

```
[33]: fig = px.area(rates_long,
                   x='Month',
                   y='Death Rate',
                   color='Cause of Death',
                   facet_col='Timeframe',
                   facet_col_spacing=0.05,
                   labels={'Month': ''},
                   height=500,
                   width=1200)

fig
```

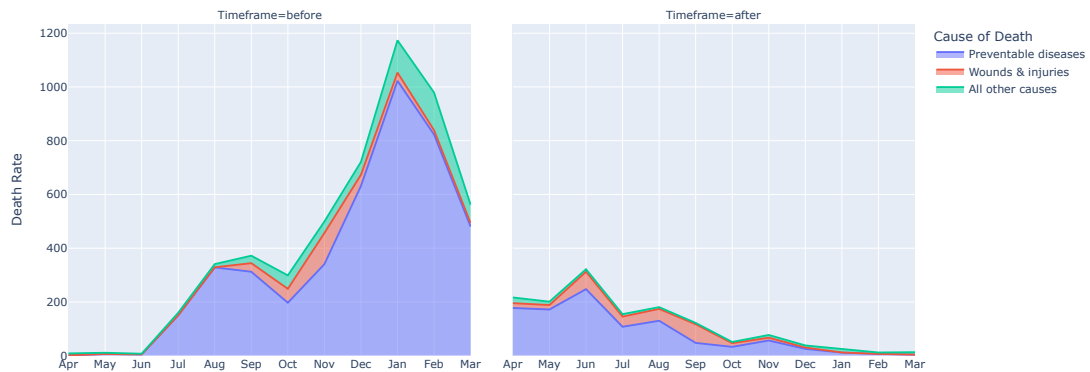


Figure 5.59 — One continuous timeline with vertical rule and annotations (cf. Tableau section).

```
[34]: fig = px.area(rates_long,
                  x='Month-Year',
                  y='Death Rate',
                  color='Cause of Death',
                  labels={'Month': ''},
                  height=500,
                  width=1200)

fig.add_vline(x=12, line_width=3, line_dash='dash', line_color='black')

fig.add_annotation(x=4, y=900,
                  text='Before Sanitation Measures',
                  font=dict(family='Courier New, monospace', size=16,
                  ↪color='#ffffff'),
                  bordercolor='#c7c7c7',
                  borderwidth=2,
                  borderpad=4,
                  bgcolor='lightslategray',
                  showarrow=False,
                  opacity=0.8)

fig.add_annotation(x=19.15, y=900,
                  text='After Sanitation Measures',
                  font=dict(family='Courier New, monospace', size=16,
                  ↪color='#ffffff'),
                  bordercolor='#c7c7c7',
                  borderwidth=2,
                  borderpad=4,
                  bgcolor='lightslategray',
```

```
showarrow=False,
opacity=0.8)
```

```
fig
```

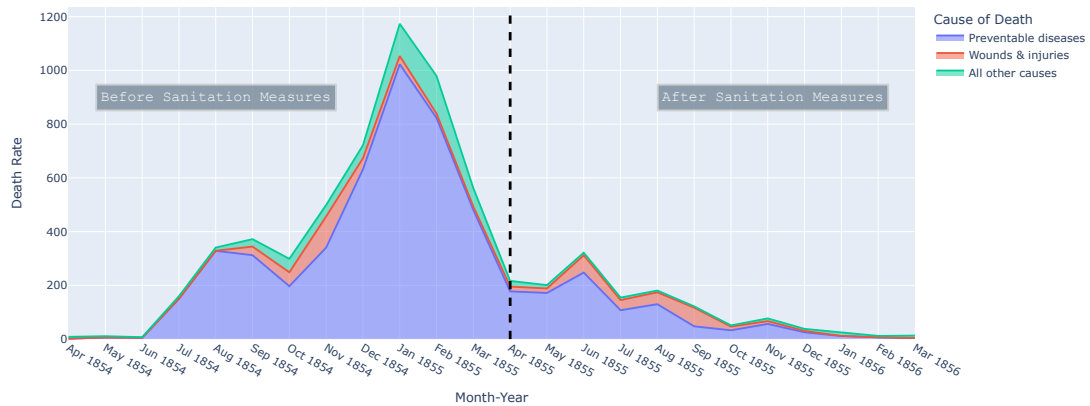


Figure 5.60 — Mean death rate by timeframe and cause (groupby + faceted bars).

```
[35]: grouped = rates_long.groupby(['Timeframe', 'Cause of Death']).
      ↪mean(numeric_only=True)
      grouped
```

```
[35]:
```

		Year	Army	Death Rate
Timeframe	Cause of Death			
after	All other causes	1855.25	41946.583333	9.508333
	Preventable diseases	1855.25	41946.583333	84.850000
	Wounds & injuries	1855.25	41946.583333	23.316667
before	All other causes	1854.25	28006.000000	44.408333
	Preventable diseases	1854.25	28006.000000	358.166667
	Wounds & injuries	1854.25	28006.000000	25.125000

```
[36]: grouped.reset_index(inplace=True)
      grouped
```

```
[36]:
```

	Timeframe	Cause of Death	Year	Army	Death Rate
0	after	All other causes	1855.25	41946.583333	9.508333
1	after	Preventable diseases	1855.25	41946.583333	84.850000
2	after	Wounds & injuries	1855.25	41946.583333	23.316667
3	before	All other causes	1854.25	28006.000000	44.408333
4	before	Preventable diseases	1854.25	28006.000000	358.166667
5	before	Wounds & injuries	1854.25	28006.000000	25.125000

```
[37]: fig = px.bar(grouped,
                    x='Timeframe',
                    y='Death Rate',
                    color='Timeframe',
                    category_orders={'Timeframe': ['before', 'after']},
                    color_discrete_map={'before': 'darkorange', 'after': 'blue'},
                    facet_col='Cause of Death',
                    labels={'Timeframe': ''},
                    height=500,
                    width=1200)

fig.update_xaxes(ticklabelposition='inside bottom')

fig
```

