

## Chapter 4

# The Seaborn essentials for data visualization

# Objectives (part 1)

## Applied

1. Use the Seaborn methods to create these types of plots:
  - line plot
  - scatter plot
  - bar plot
  - box plot
  - histogram
  - KDE plot
  - ECDF plot
2. Use Seaborn parameters to create a plot with subplots.

## Objectives (part 2)

3. Use the methods of the Axes object to enhance a plot in these ways:
  - add a title and the labels for the x- and y-axis
  - set the ticks for a plot
  - set the x and y limits
  - add grid lines
  - annotate a plot
  - set the color palette
4. Use the methods of a Grid object to add a super title to a plot that has subplots.
5. Use the methods of the Grid or Axes object to save a plot to a file.

# Objectives (part 3)

## Knowledge

1. Describe two differences between Seaborn's general and specific methods for plotting.
2. Describe these Seaborn plot categories and identify one type of plot in each group:

Relational

Categorical

Distribution

## Relational plots

General method	Object returned	Plot type
<code>relplot(params)</code>	FacetGrid	Relational

Specific method	Object returned	Plot type
<code>scatterplot(params)</code>	Axes	Scatter plot
<code>lineplot(params)</code>	Axes	Line plot

# Categorical plots

General method	Object returned	Plot type
<code>catplot(params)</code>	FacetGrid	Categorical

Specific method	Object returned	Plot type
<code>barplot(params)</code>	Axes	Bar plot (or bar chart)
<code>boxplot(params)</code>	Axes	Box plot
<code>stripplot(params)</code>	Axes	Strip plot
<code>swarmplot(params)</code>	Axes	Swarm plot
<code>pointplot(params)</code>	Axes	Point plot
<code>boxenplot(params)</code>	Axes	Boxen plot
<code>violinplot(params)</code>	Axes	Violin plot
<code>countplot(params)</code>	Axes	Count plot

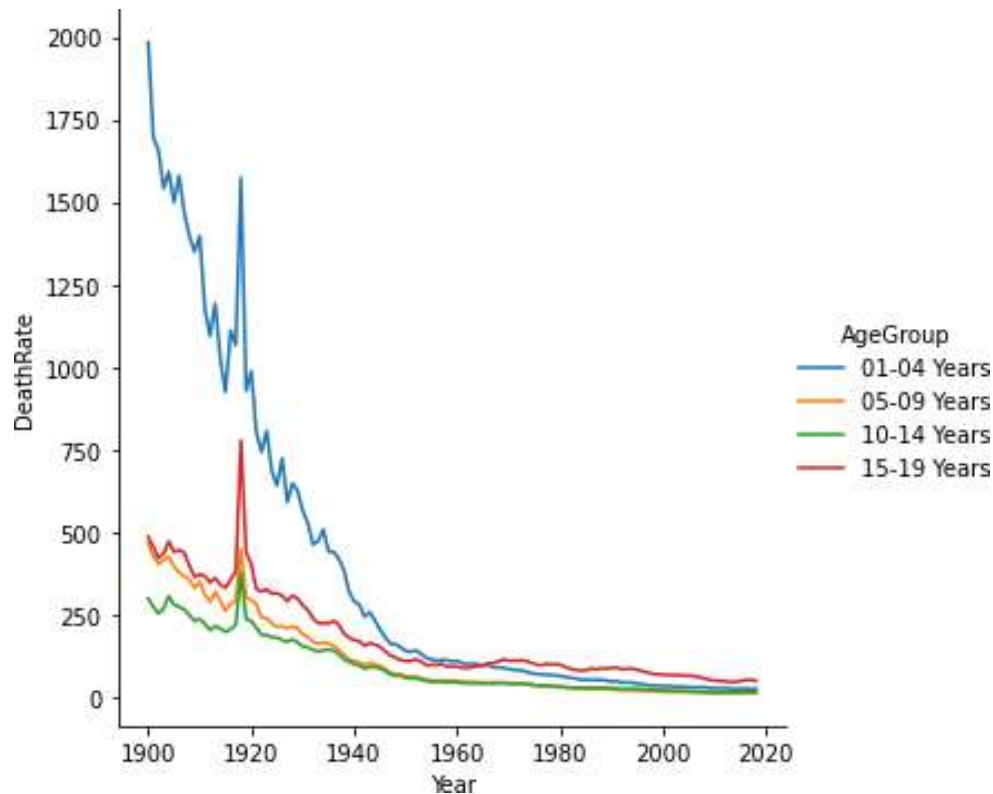
## Distribution plots

General method	Object returned	Plot type
<code>displot(params)</code>	FacetGrid	Distribution

Specific method	Object returned	Plot type
<code>histplot(params)</code>	Axes	Histogram
<code>kdeplot(params)</code>	Axes	Kernel density estimate
<code>ecdfplot(params)</code>	Axes	Empirical cumulative distribution function

# How to use the relplot() method to create a line plot

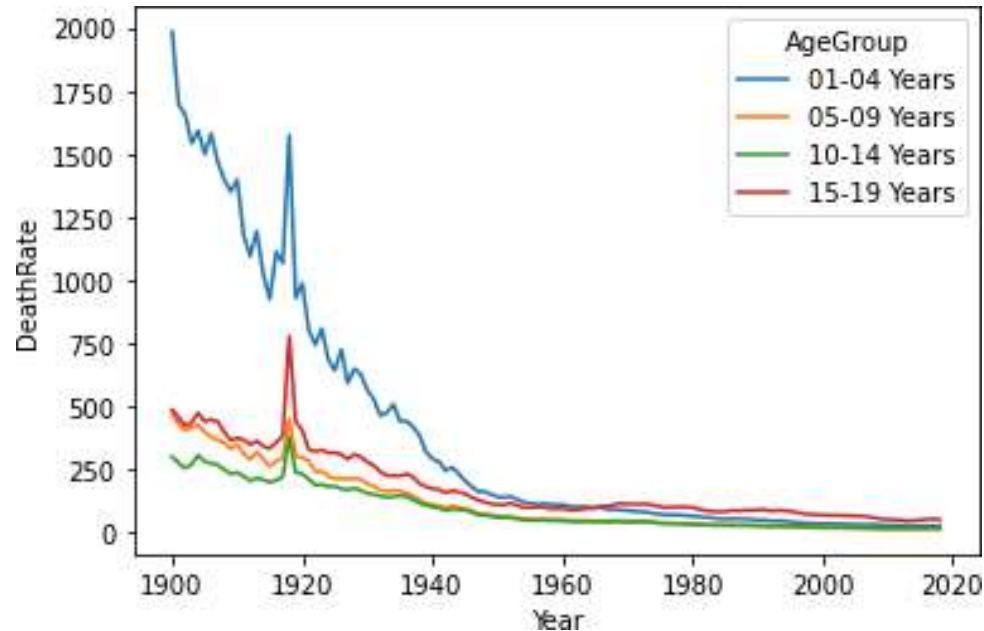
```
import seaborn as sns
sns.relplot(data=mortality_data, kind='line',
            x='Year', y='DeathRate', hue='AgeGroup')
```





# How to use the lineplot() method to create a line plot

```
sns.lineplot(data=mortality_data,  
             x='Year', y='DeathRate', hue='AgeGroup')
```



# The differences between the general and specific methods

- The general methods require the `kind` parameter; the specific methods don't.
- You can't use the specific methods to create plots that contain more than one subplot.
- The general methods return a `Grid` object that contains an `Axes` object for each subplot. The specific methods return an `Axes` object.

# The basic Seaborn parameters

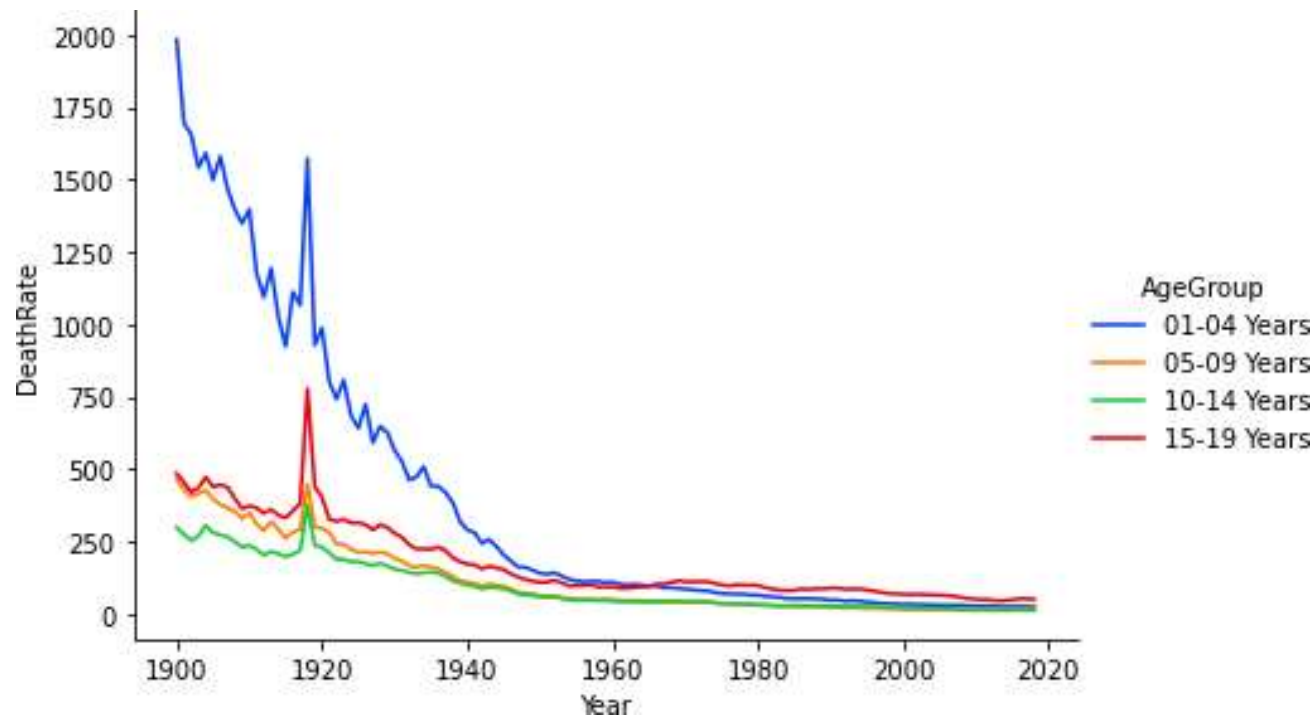
Parameter	Description
<b>data</b>	The DataFrame that provides the data.
<b>kind</b>	The type of plot.
<b>x, y</b>	The columns for the x- and y-axis.
<b>hue</b>	The column that determines the categories to plot using a different color for each category.
<b>palette</b>	The color palette for the plot. The default is None.

## The basic Seaborn parameters (continued)

Parameter	Description
<b>height</b>	The height of the plot in inches with a default of 5. Only for general plots.
<b>aspect</b>	The ratio of the width to the height so: width = aspect * height. Only for general plots.
<b>legend</b>	Four options that determine how the legend will be drawn: auto (the default), brief, full, and False. Only for general plots.

## A line plot that uses most of the parameters

```
import seaborn as sns
sns.relplot(data=mortality_data, kind='line',
            x='Year', y='DeathRate', hue='AgeGroup',
            palette='bright', height=4, aspect=1.5)
```



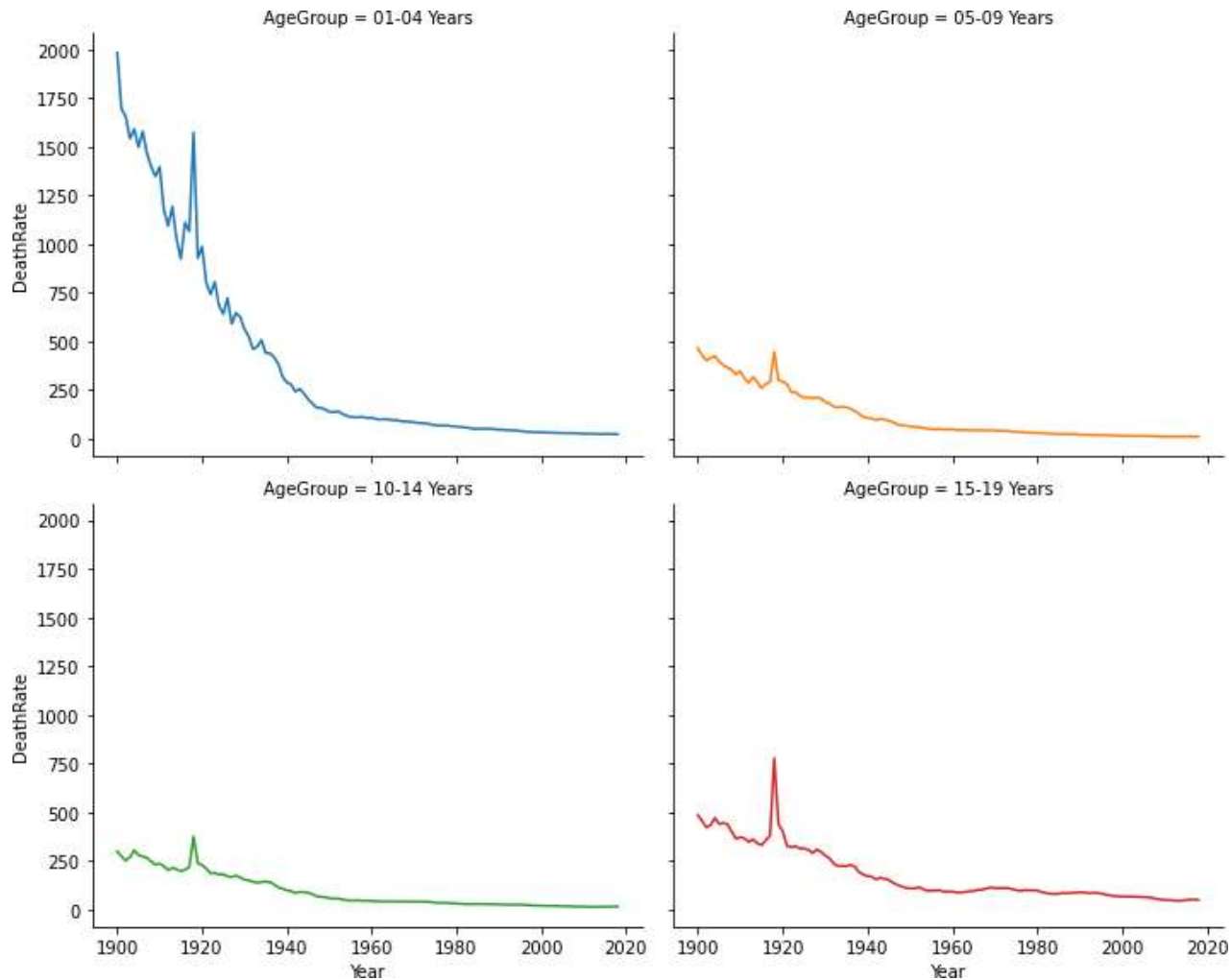
## The Seaborn parameters for subplots

Parameter	Description
<code>col</code>	The column that determines the subplots.
<code>col_wrap</code>	The number of subplots in each row.

## How to create a plot with four subplots

```
sns.relplot(data=mortality_data, kind='line',  
            x='Year', y='DeathRate', hue='AgeGroup',  
            height=4, aspect=1.25, col='AgeGroup',  
            col_wrap=2, legend=False)
```

# The plot with four subplots



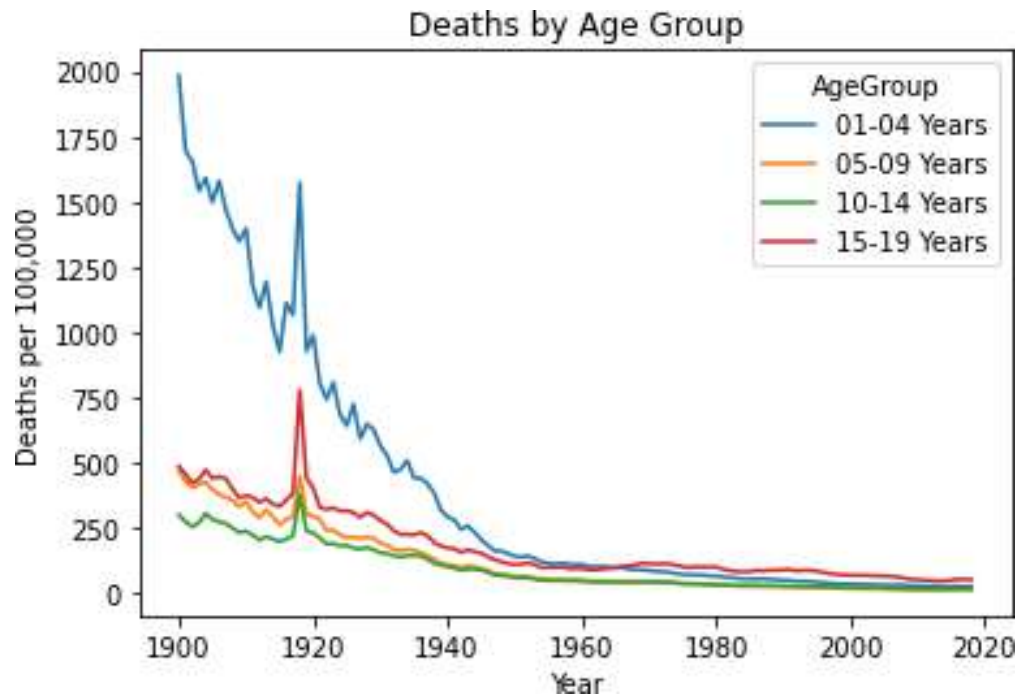
# The title and label parameters for the `set()` method of an Axes object

Parameter	Description
<code>title</code>	The title
<code>xlabel</code> , <code>ylabel</code>	The labels for the x- and y-axis



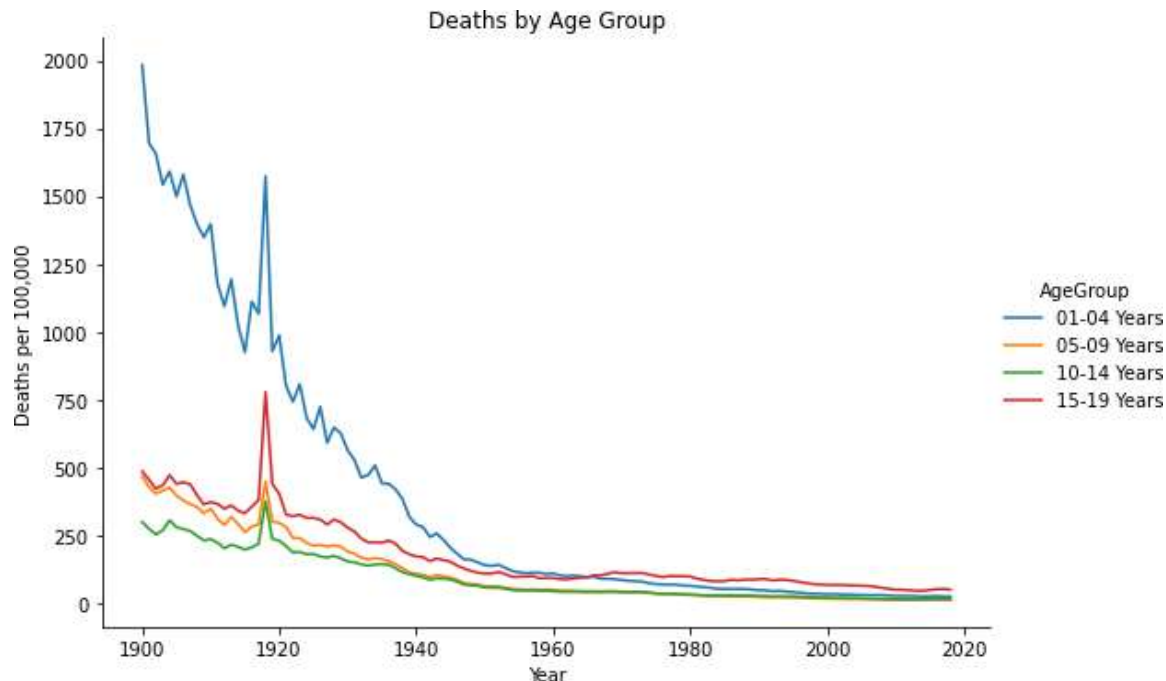
# How to use the set() method to enhance a specific plot

```
ax = sns.lineplot(data=mortality_data,  
                  x='Year', y='DeathRate', hue='AgeGroup')  
ax.set(title='Deaths by Age Group',  
       ylabel='Deaths per 100,000')
```



# How to use the set() method to enhance a general plot

```
g = sns.relplot(data=mortality_data, kind='line',  
                x='Year', y='DeathRate', hue='AgeGroup', aspect=1.5)  
for ax in g.axes.flat:  
    ax.set(title='Deaths by Age Group',  
           ylabel='Deaths per 100,000')
```



## The ticks and limit parameters for the `set()` method of an Axes object

Parameter	Description
<code>xticks, yticks</code>	The locations and values for the xticks and yticks.
<code>xlim, ylim</code>	Tuples that set the upper and lower limits of the x- and y-axis.

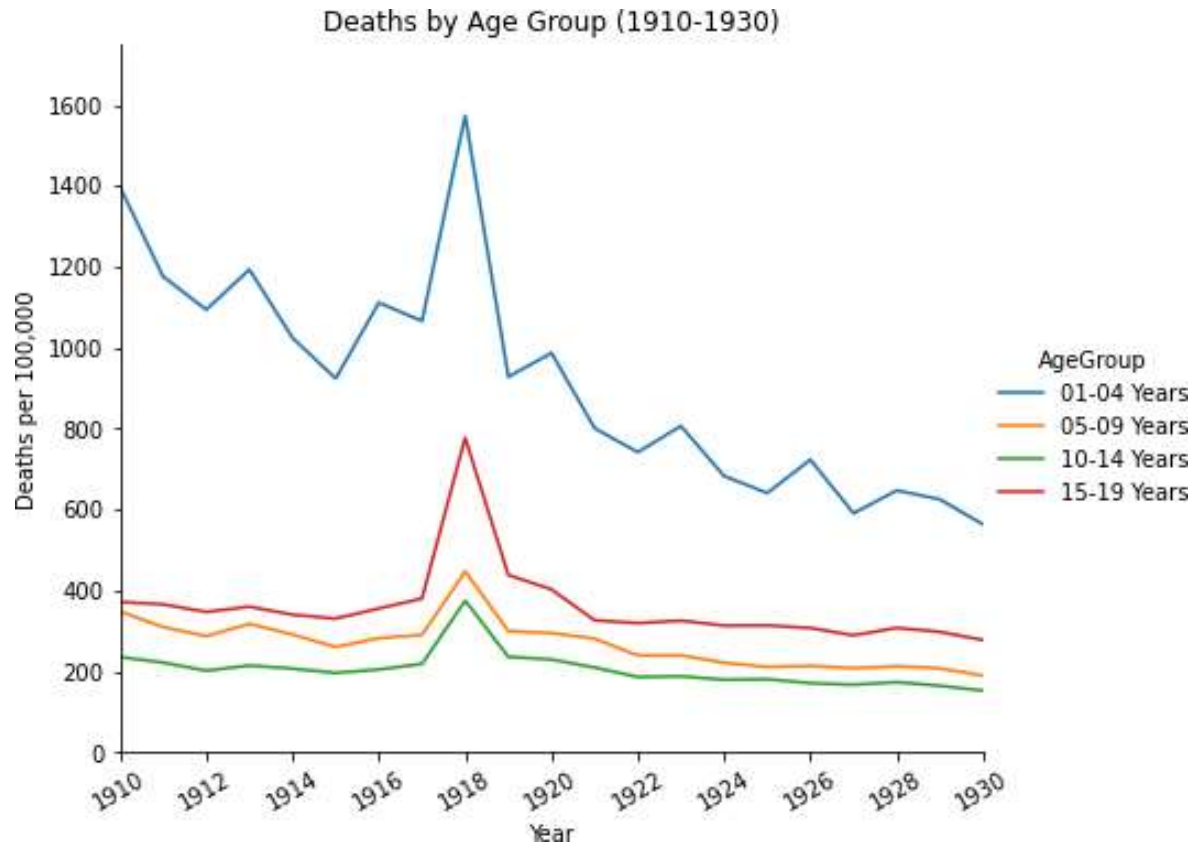
## The `tick_params()` method of an Axes object

Method	Description
<code>tick_params()</code>	Sets tick parameters like the label rotation for the xticks or yticks.

## How to use the `set()` and `tick_params()` methods to enhance a general plot

```
g = sns.relplot(data=mortality_data, kind='line',
                x='Year', y='DeathRate', hue='AgeGroup', aspect=1.25)
for ax in g.axes.flat:
    ax.set(title='Deaths by Age Group (1910-1930)',
           ylabel='Deaths per 100,000',
           xticks=[x for x in range(1910, 1931, 2)],
           xlim=(1910,1930), ylim=(0,1750)),
    ax.tick_params('x', labelrotation=30)
```

# The plot that uses `set()` and `tick_params()`



## How to enhance a specific plot in the same way

```
ax = sns.lineplot(data=mortality_data,  
                  x='Year', y='DeathRate', hue='AgeGroup')  
ax.set(title='Deaths by Age Group (1910-1930)',  
       ylabel='Deaths per 100,000',  
       xticks=[x for x in range(1910, 1931, 2)],  
       xlim=(1910,1930), ylim=(0,1750))  
ax.tick_params('x', labelrotation=30)
```

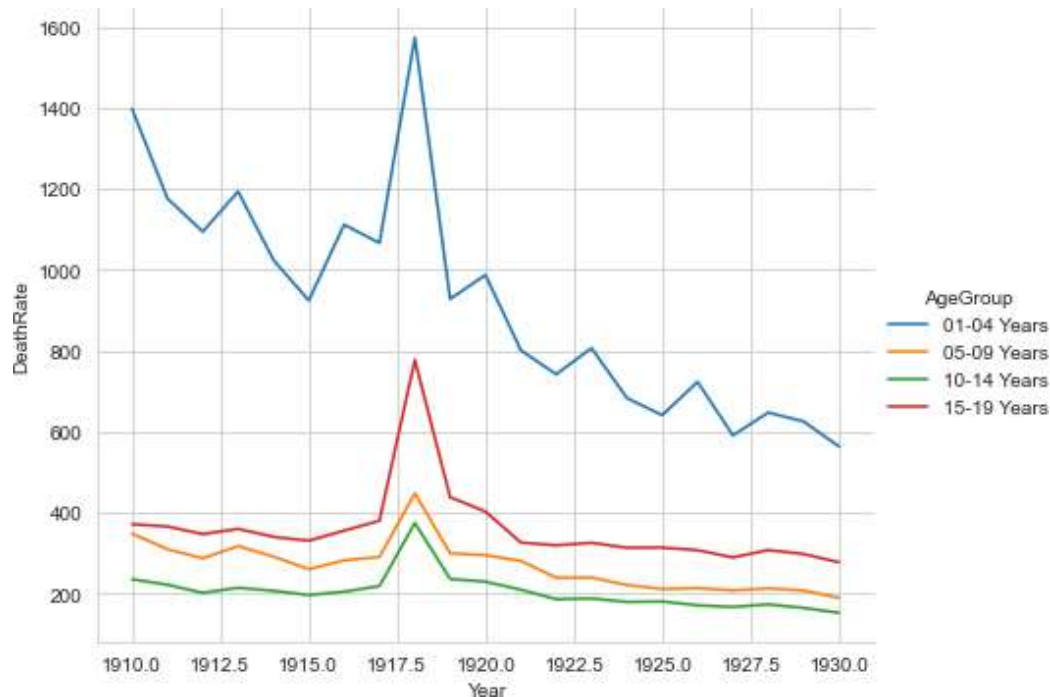
# The Seaborn set\_style() method

```
set_style(style)
```

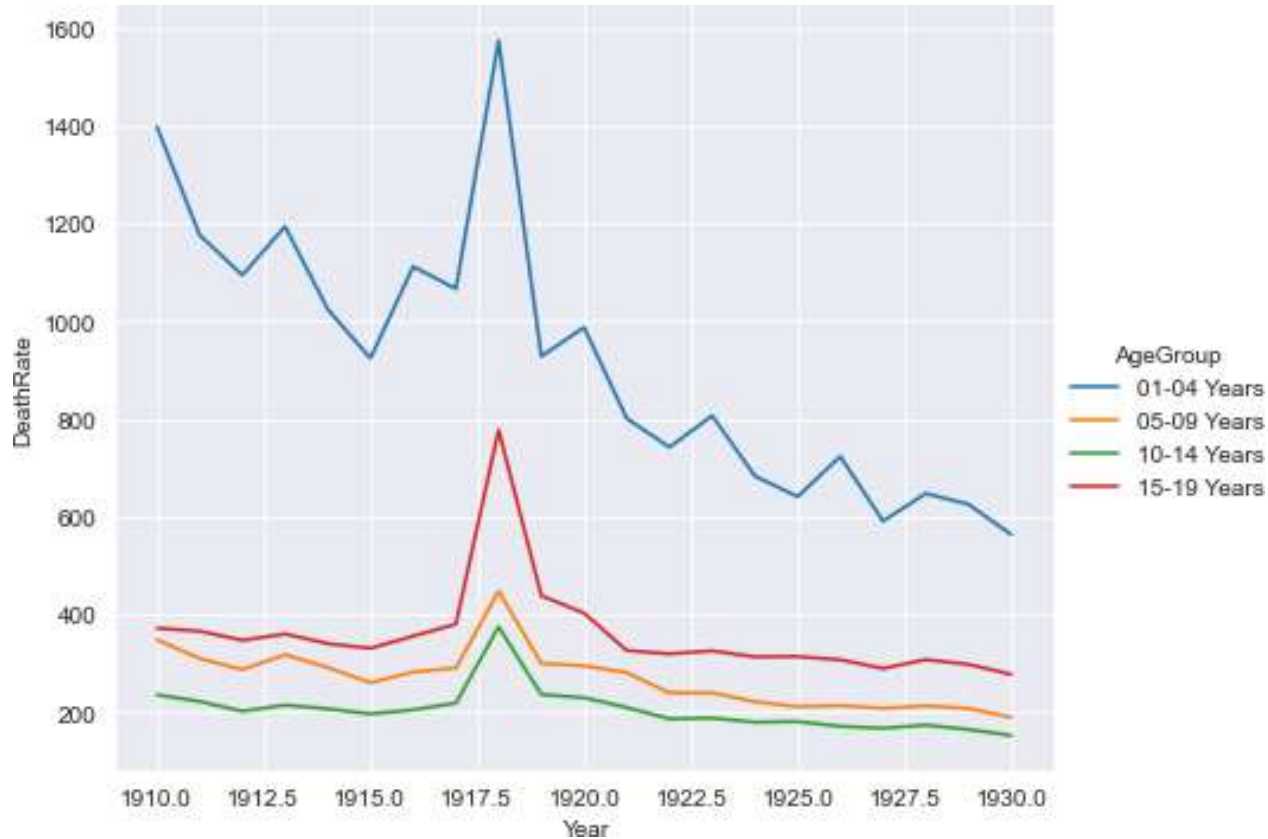
## How to set the background style

```
sns.set_style('whitegrid')
```

```
sns.relplot(data=mortality_data.query('Year >= 1910 and Year <= 1930'),  
            kind='line', x='Year', y='DeathRate', hue='AgeGroup', aspect=1.25)
```



## The same plot but with the darkgrid style





# How to restore the default background style

```
sns.set_style('ticks')
```

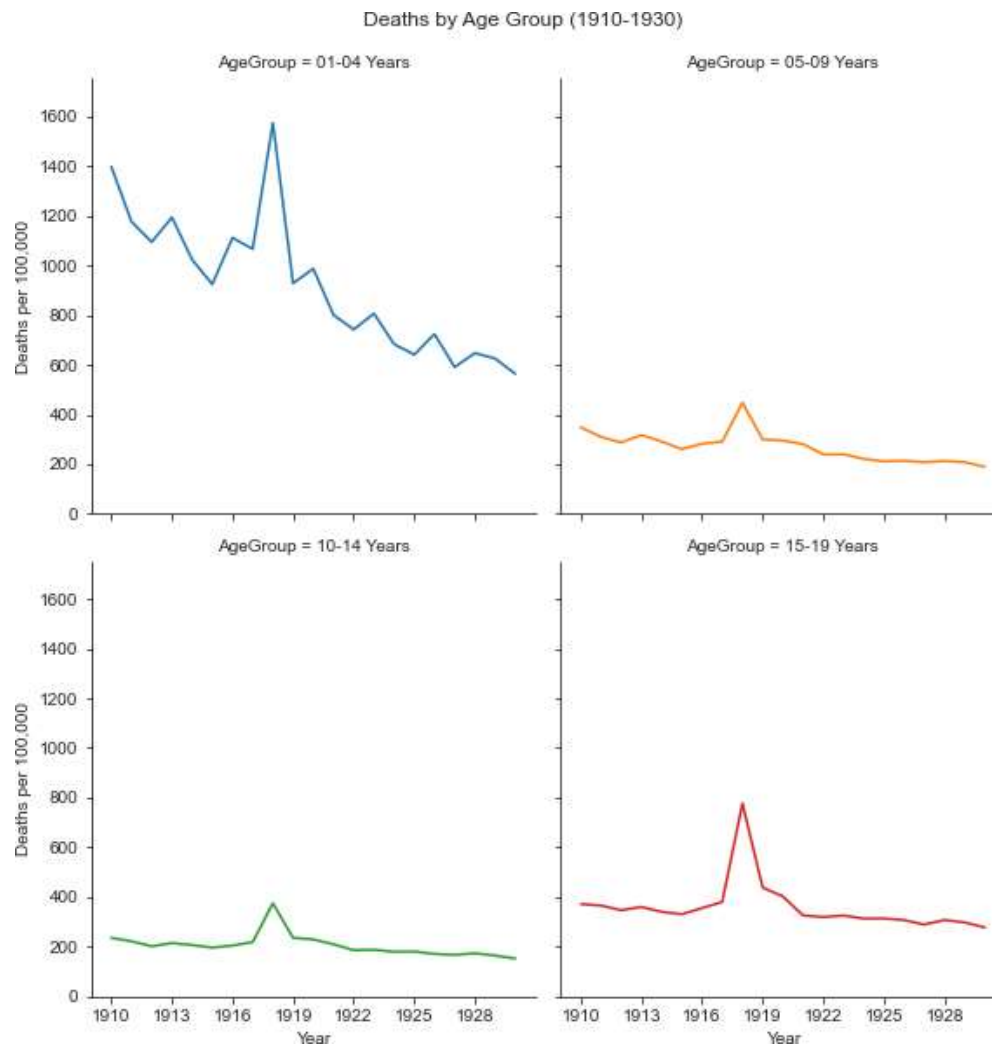
# Parameters of the `suptitle()` method of the `Figure` object

Parameter	Description
<code>t</code>	Sets the text for the title.
<code>y</code>	Sets the y location of the title in figure coordinates where 1 is the top of the figure and 0 is the bottom of the figure.

# How to use the `set()` and `suptitle()` methods with subplots

```
g = sns.relplot(  
    data=mortality_data.query('Year >= 1910 and Year <= 1930'),  
    kind='line', x='Year', y='DeathRate', hue='AgeGroup', legend=False,  
    col='AgeGroup', col_wrap=2, height=4)  
g.fig.suptitle('Deaths by Age Group (1910-1930)', y=1.025)  
for ax in g.axes.flat:  
    ax.set(ylabel='Deaths per 100,000',  
          xticks=[x for x in range(1910,1931,2)],  
          ylim=(0,1750))
```

# Subplots created using set() and supitle()



## A few of the supported file formats

File extension	File format
<b>.png</b>	PNG (Portable Network Graphics)
<b>.svg</b>	SVG (Scalable Vector Graphics)
<b>.pdf</b>	PDF (Portable Document Format)

## The `savefig()` method of a `FacetGrid` object or a `Figure` object

`savefig(filename)`

## The `get_figure()` method of the `Axes` object

`get_figure()`

## The `subplots_adjust()` method of the `Figure` object

`subplots_adjust()`

## How to save a general plot

```
g = sns.relplot(data=mortality_data, kind='line',  
                x='Year', y='DeathRate', hue='AgeGroup')  
  
# adjust the bottom to fix the y labels  
g.fig.subplots_adjust(bottom=0.35)  
  
g.savefig('lineChart.png')  
g.savefig('lineChart.svg')  
g.savefig('figures/lineChart.pdf')
```

## How to save a specific plot

```
ax = sns.lineplot(data=mortality_data,  
                  x='Year', y='DeathRate', hue='AgeGroup')  
ax.get_figure().savefig('lineChart.png')  
ax.get_figure().savefig('lineChart.svg')  
ax.get_figure().savefig('figures/lineChart.pdf')
```

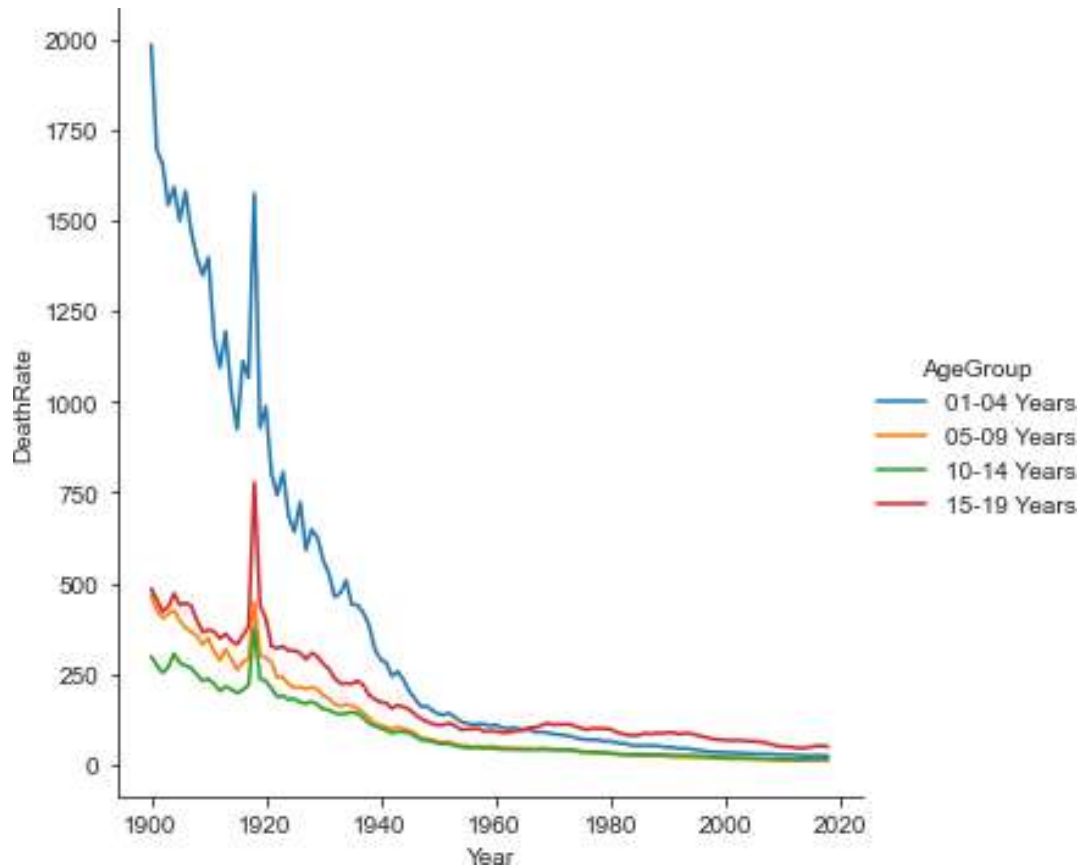


## The ci parameter for line plots

Parameter	Description
<b>ci</b>	The size of the confidence interval from 0 to 100. The default is 95. If None, the interval isn't shown. If 'sd', the interval is the standard deviation.

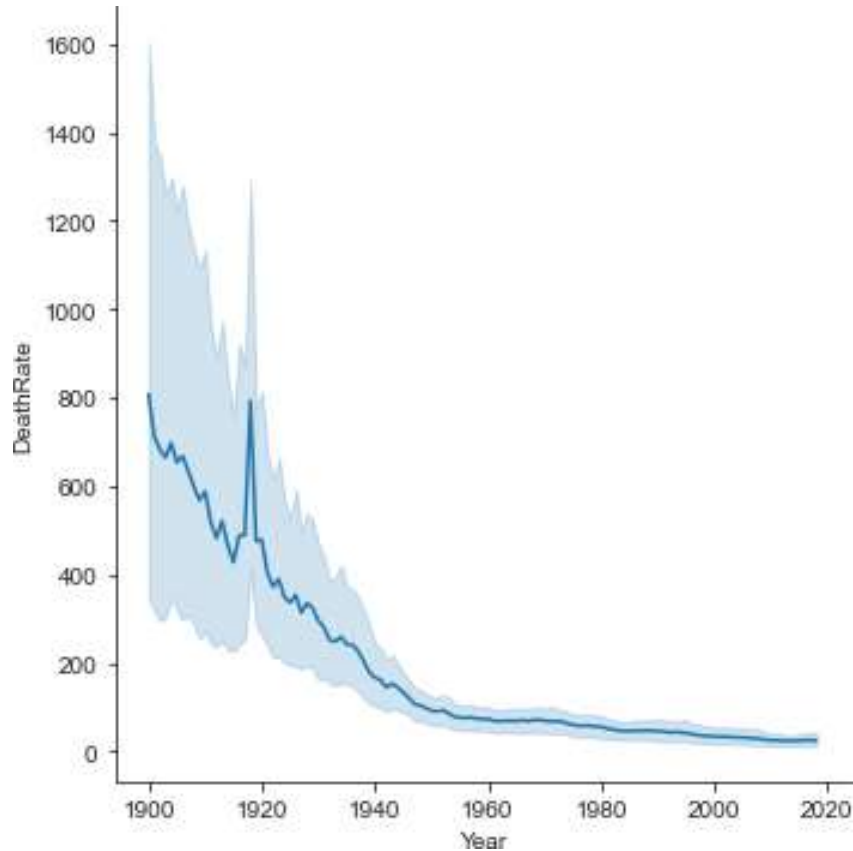
# How to create a line plot

```
sns.relplot(data=mortality_data, kind='line', x='Year',  
            y='DeathRate', hue='AgeGroup')
```



# How to create a line plot with a confidence interval

```
sns.relplot(data=mortality_data, kind='line',  
            x='Year', y='DeathRate')
```

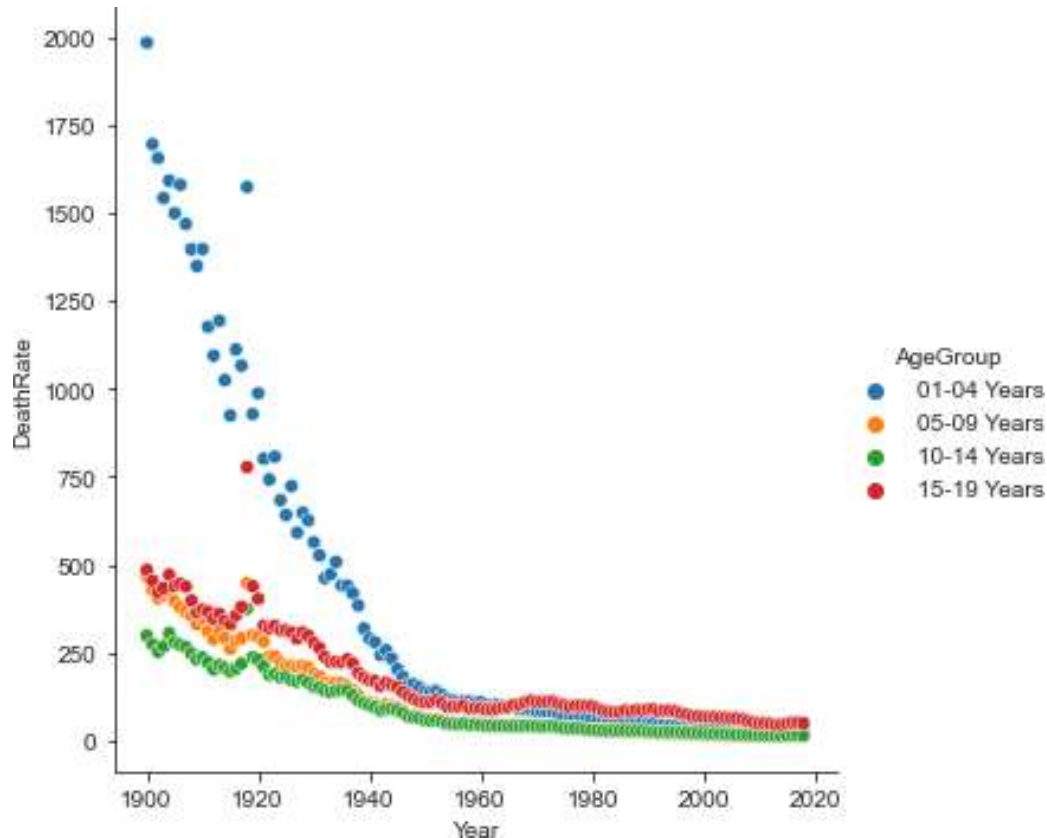


# The size and sizes parameters for a scatter plot

Parameter	Description
<b>size</b>	A numeric column that determines the size of the dots.
<b>sizes</b>	A tuple that sets the smallest and largest size of the dots.

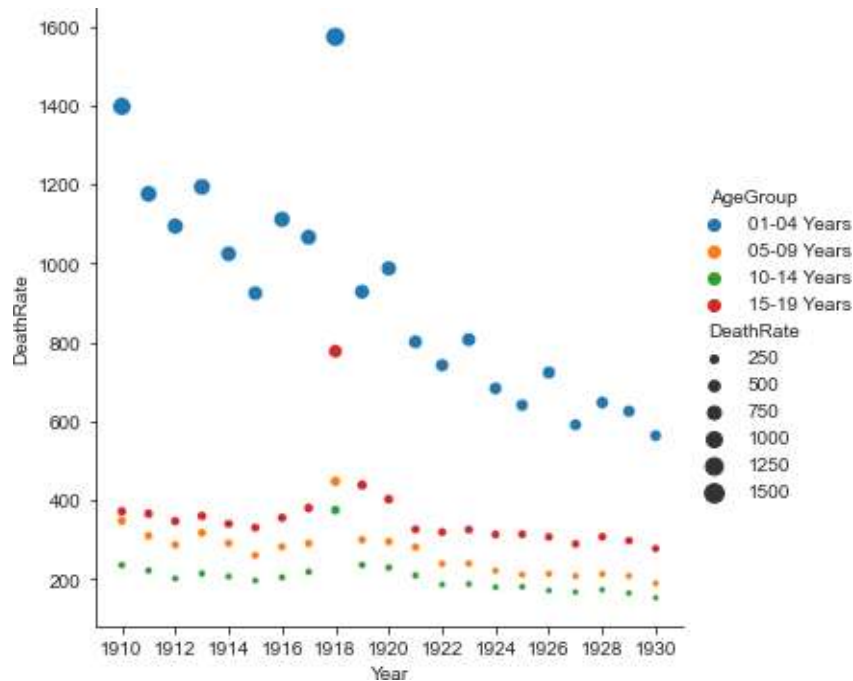
# How to create a scatter plot

```
sns.relplot(data=mortality_data,  
            kind='scatter', x='Year', y='DeathRate', hue='AgeGroup')
```



# How to create a scatter plot for a smaller date range

```
g = sns.relplot(  
    data=mortality_data.query('Year >= 1910 and Year <= 1930'),  
    kind='scatter', x='Year', y='DeathRate', hue='AgeGroup',  
    size='DeathRate', sizes=(10,100))  
for ax in g.axes.flat:  
    ax.set(xticks=[x for x in range(1910,1931,2)])
```

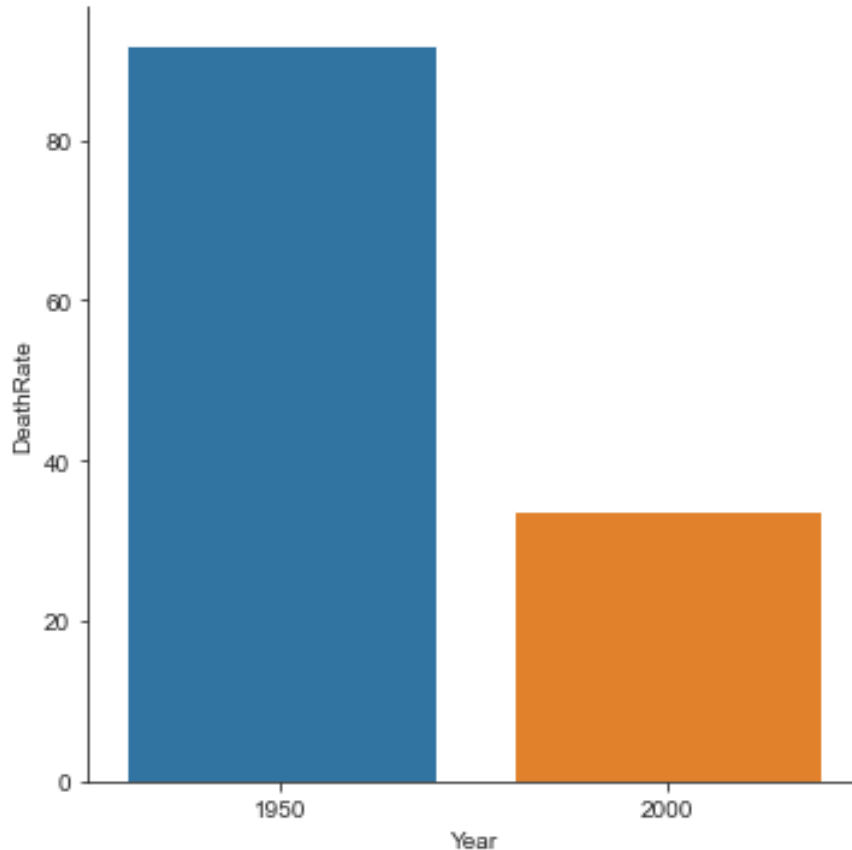


# The basic parameters for categorical plots

Parameter	Description
<b>kind</b>	The kind of plot: bar, box, strip, point, swarm, boxen, violin, or count.
<b>orient</b>	The orientation of the plot: v for vertical (the default) and h for horizontal. But this is inferred if only one dimension is numeric.
<b>ci</b>	The size of the confidence interval.

## How to create a vertical bar plot

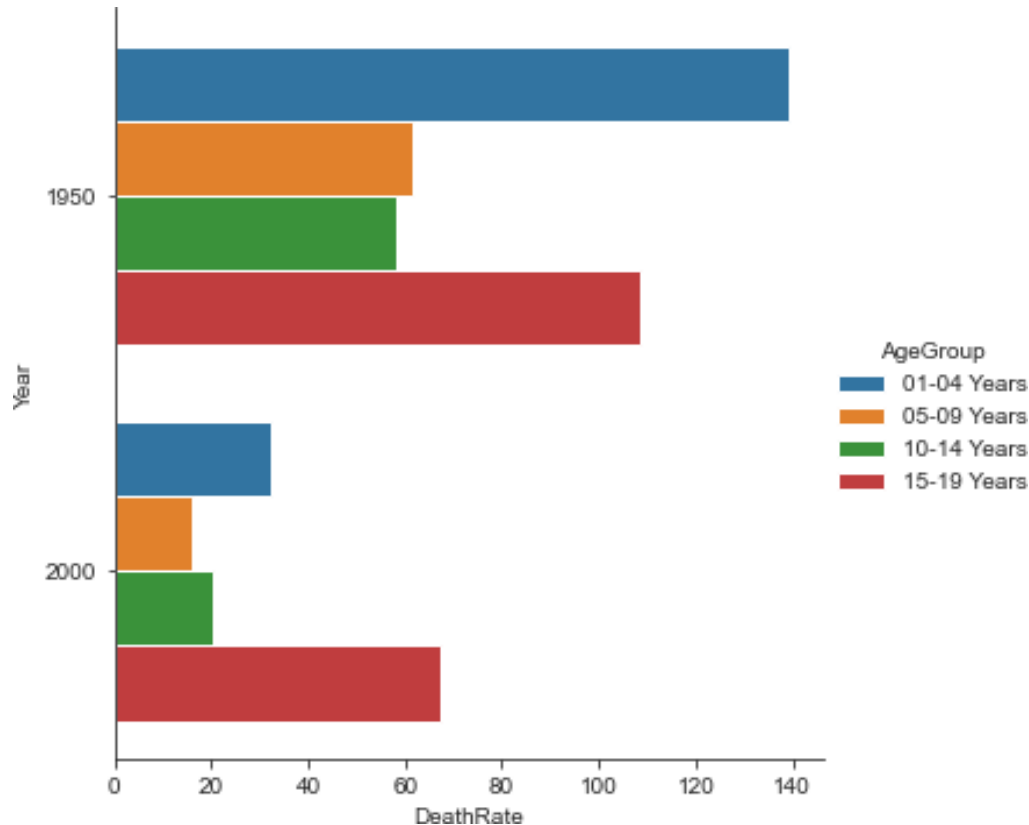
```
sns.catplot(data=mortality_data.query('Year in (1950,2000)'),  
            kind='bar', x='Year', y='DeathRate', ci=None)
```





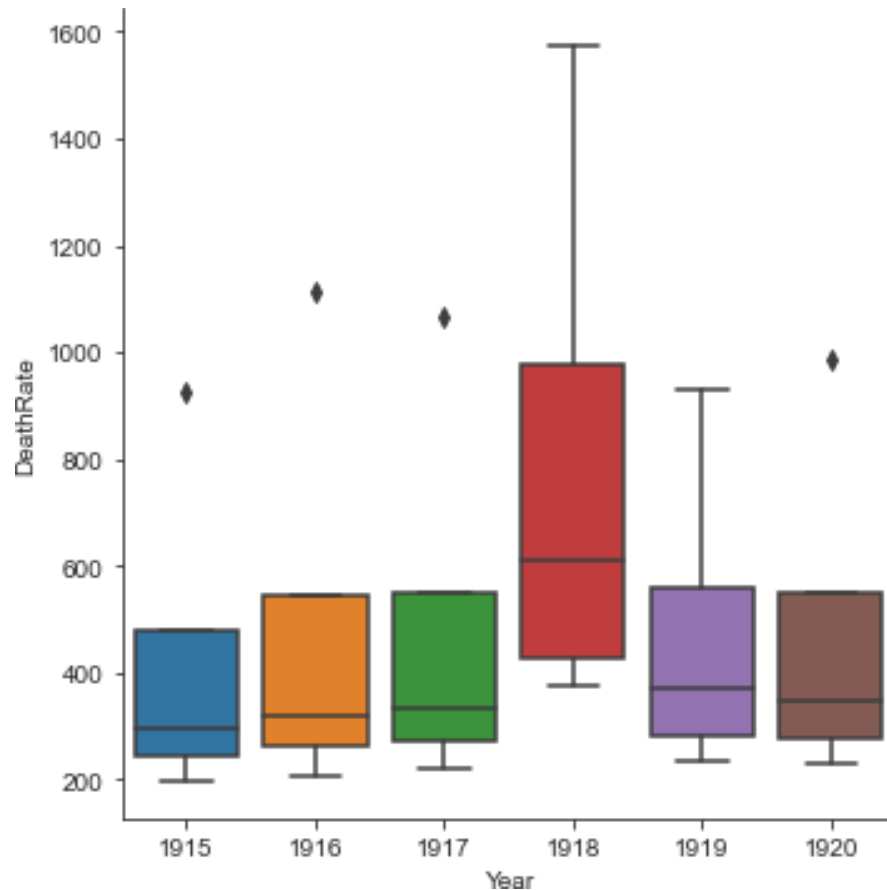
# How to create a horizontal bar plot

```
sns.catplot(data=mortality_data.query('Year in (1950,2000)'),  
            kind='bar', x='DeathRate', y='Year',  
            hue='AgeGroup', orient='h')
```



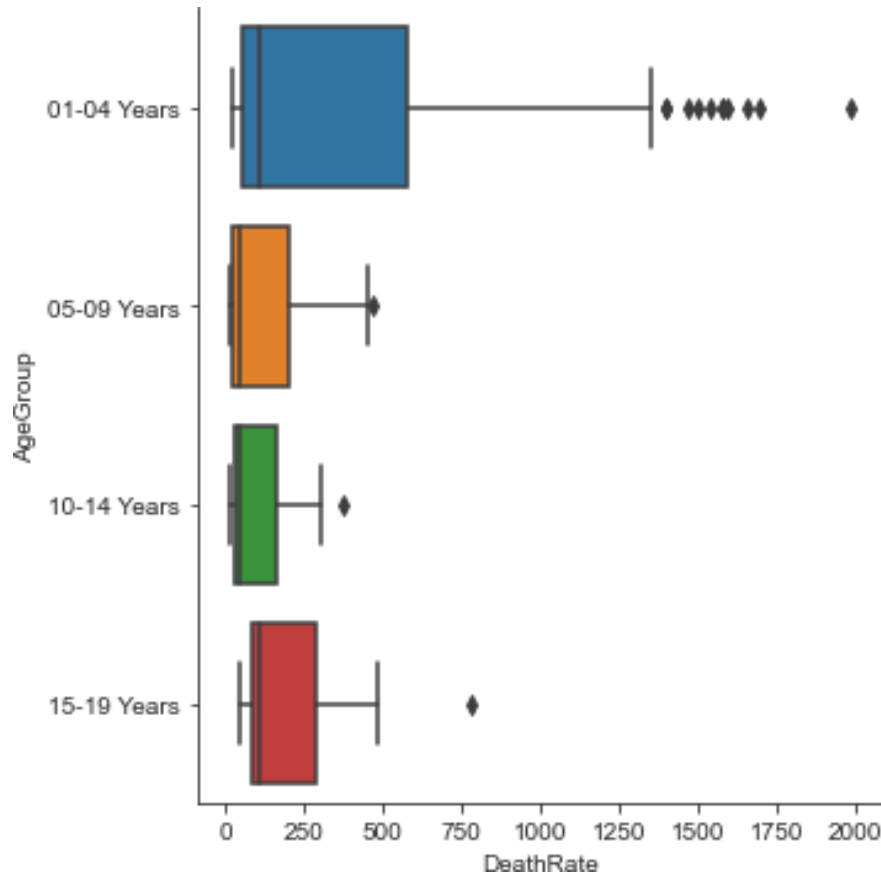
# How to create a box plot

```
sns.catplot(data=mortality_data.query('Year >= 1915 and Year <= 1920'),  
            kind='box', x='Year', y='DeathRate')
```



# How to create a horizontal box plot

```
sns.catplot(data=mortality_data, kind='box',  
            x='DeathRate', y='AgeGroup', orient='h')
```

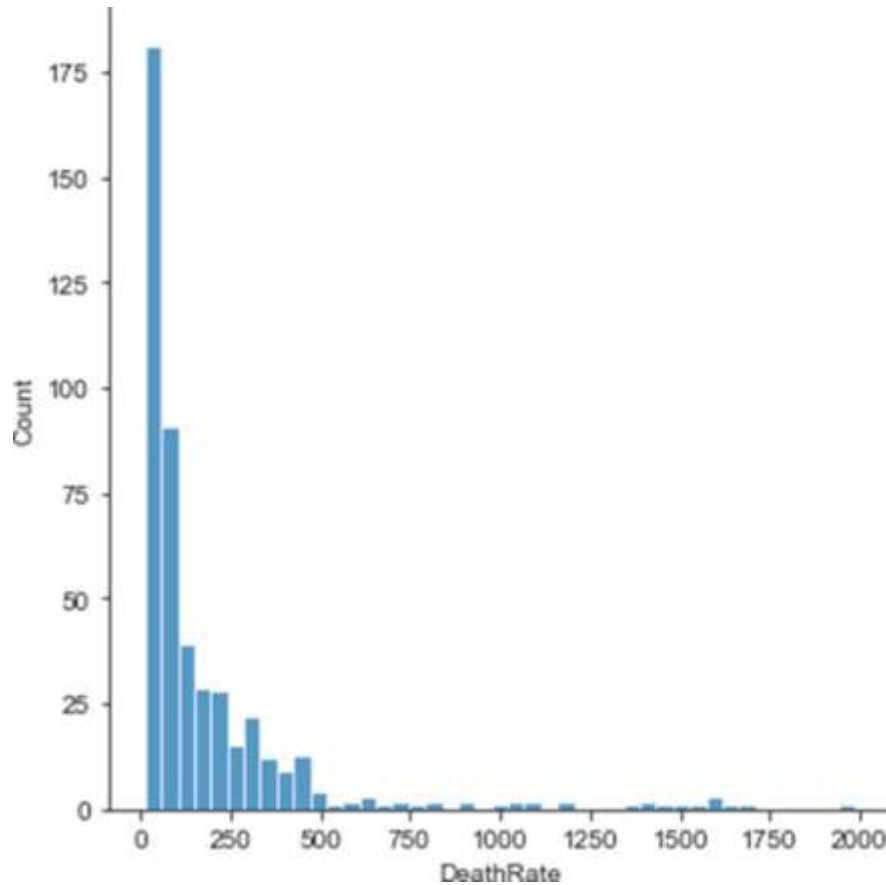


# The basic parameters for distribution plots

Parameter	Description
<b>kind</b>	Any of these: histplot, kdeplot, or ecdfplot.
<b>bins</b>	The number of bins on the x-axis. Only used with histplots.
<b>fill</b>	Fills the interior of the plot. Not used with ECDF plots.

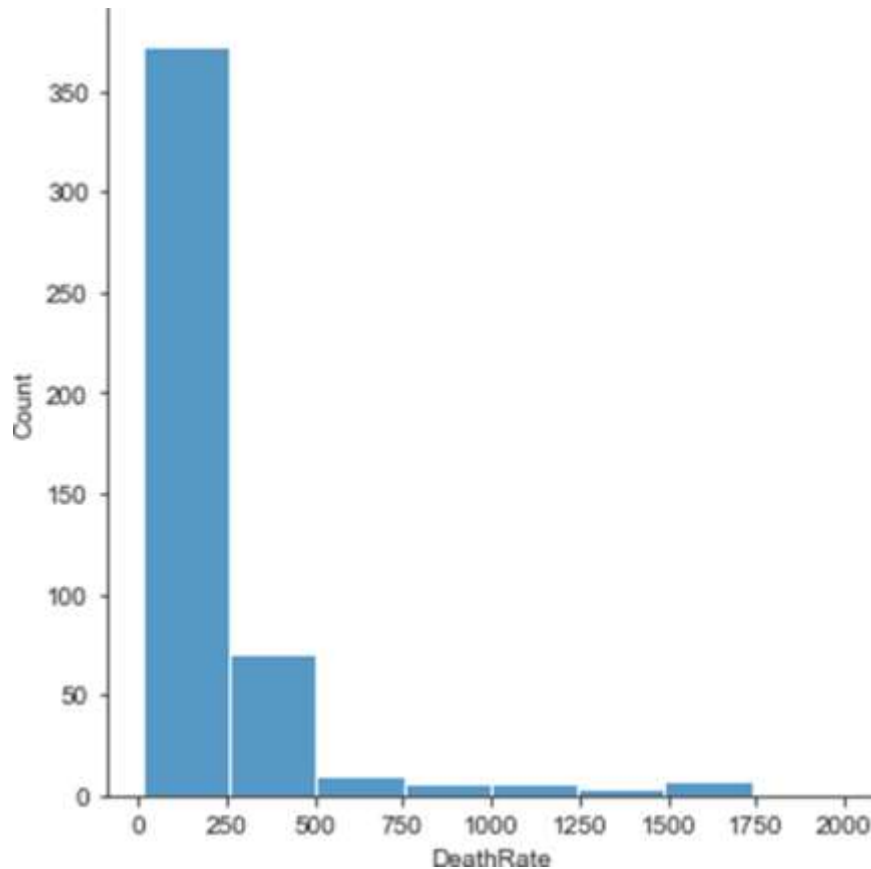
# How to create a histogram with the default settings

```
sns.displot(data=mortality_data, kind='hist', x='DeathRate')
```



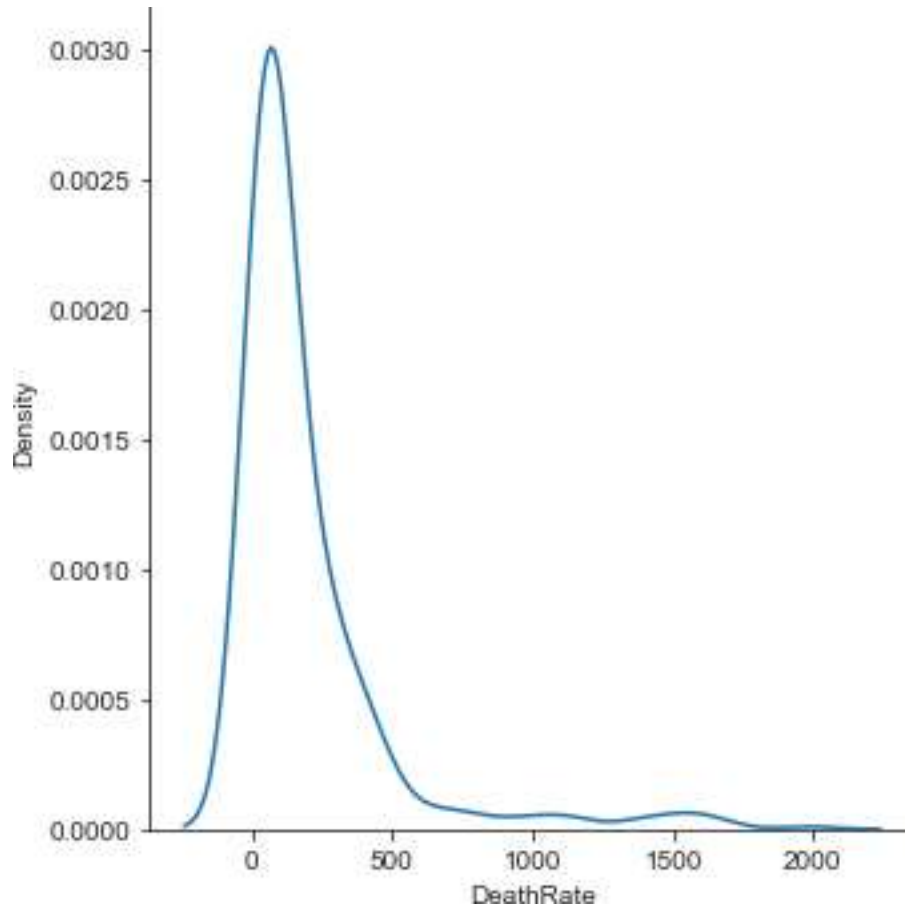
## How to create a histogram with 8 bins

```
sns.displot(data=mortality_data, kind='hist', x='DeathRate',  
            bins=8)
```



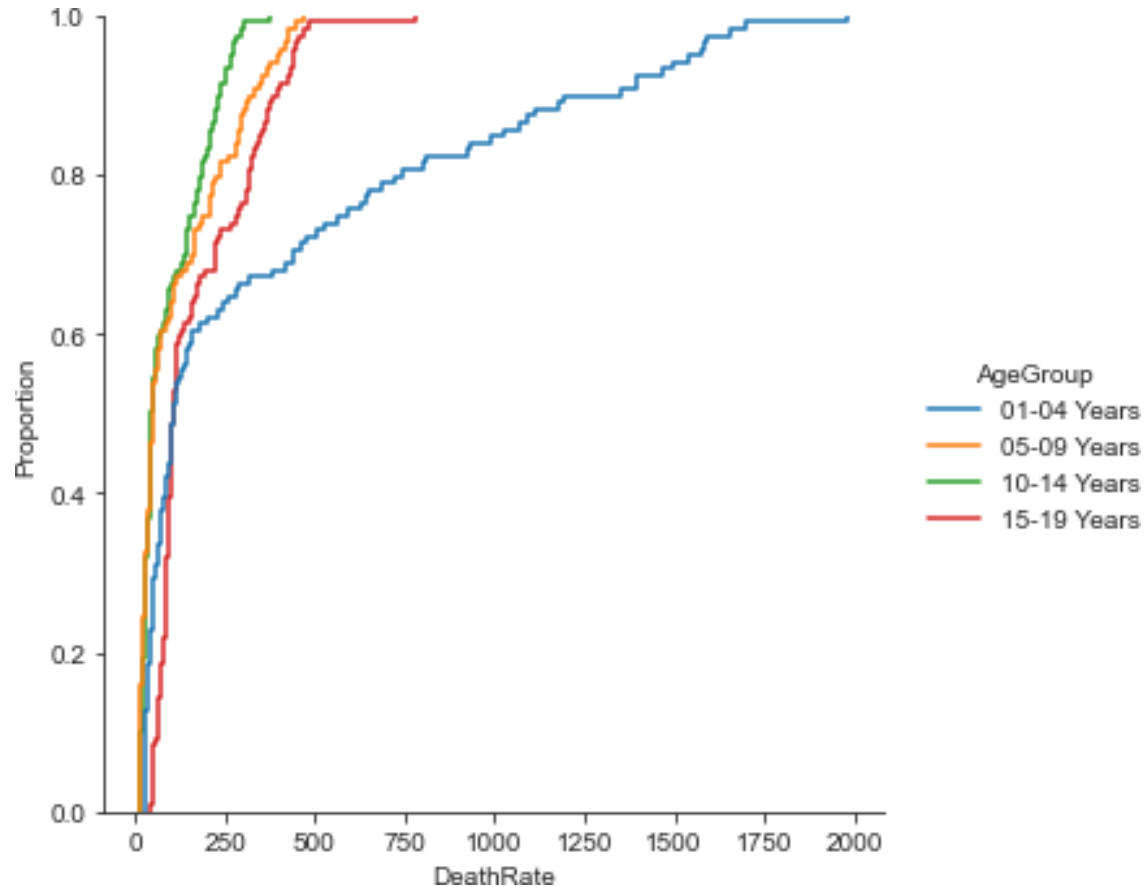
## How to create a KDE (or density) plot

```
sns.displot(data=mortality_data, kind='kde', x='DeathRate')
```



# How to create an ECDF plot

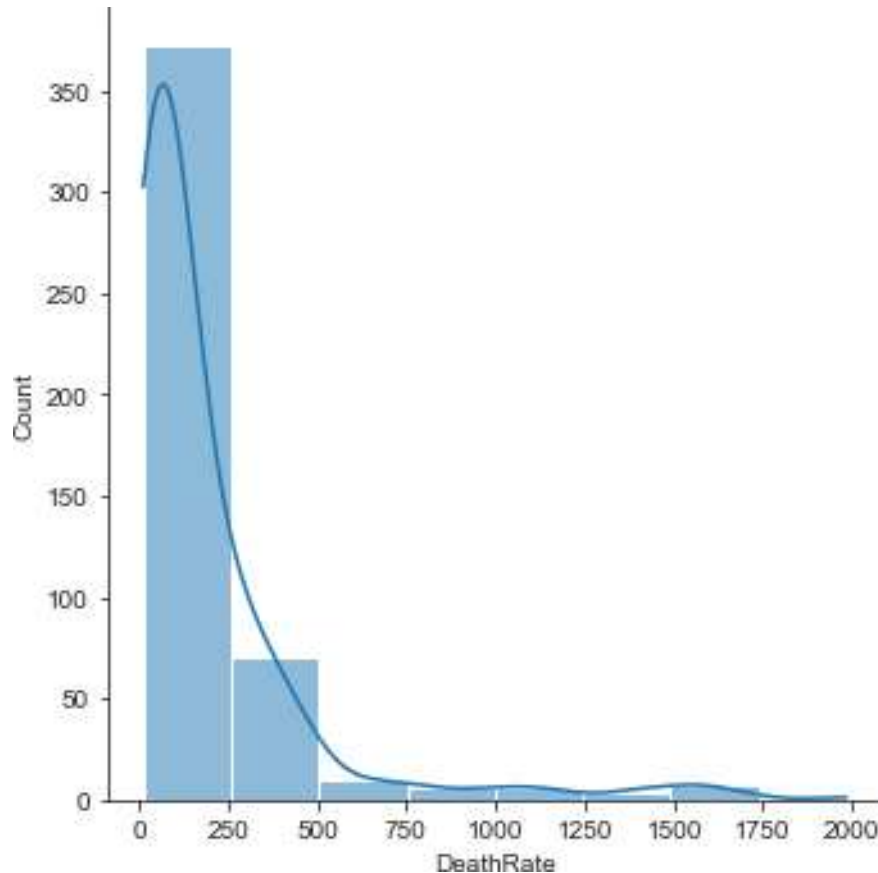
```
sns.displot(data=mortality_data, kind='ecdf', x='DeathRate',  
            hue='AgeGroup')
```





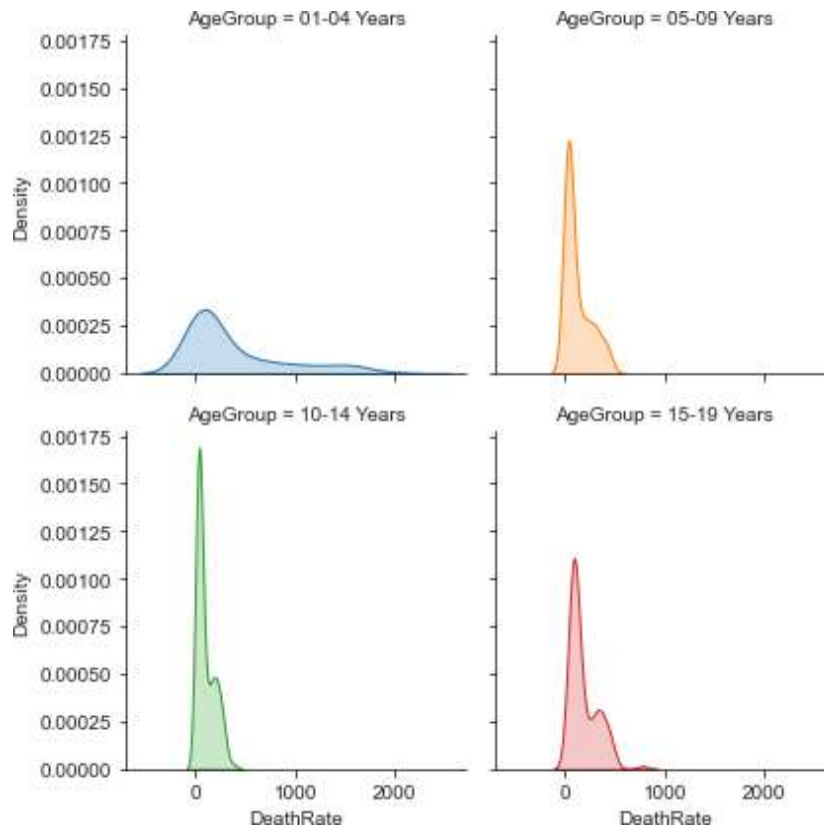
# How to combine a histogram with a KDE plot

```
sns.displot(data=mortality_data, kind='hist', x='DeathRate',  
            kde=True, bins=8)
```



# How to create a KDE plot with filled subplots

```
sns.displot(data=mortality_data, kind='kde', x='DeathRate',  
            hue='AgeGroup', fill=True, col='AgeGroup',  
            col_wrap=2, height=3, legend=False)
```



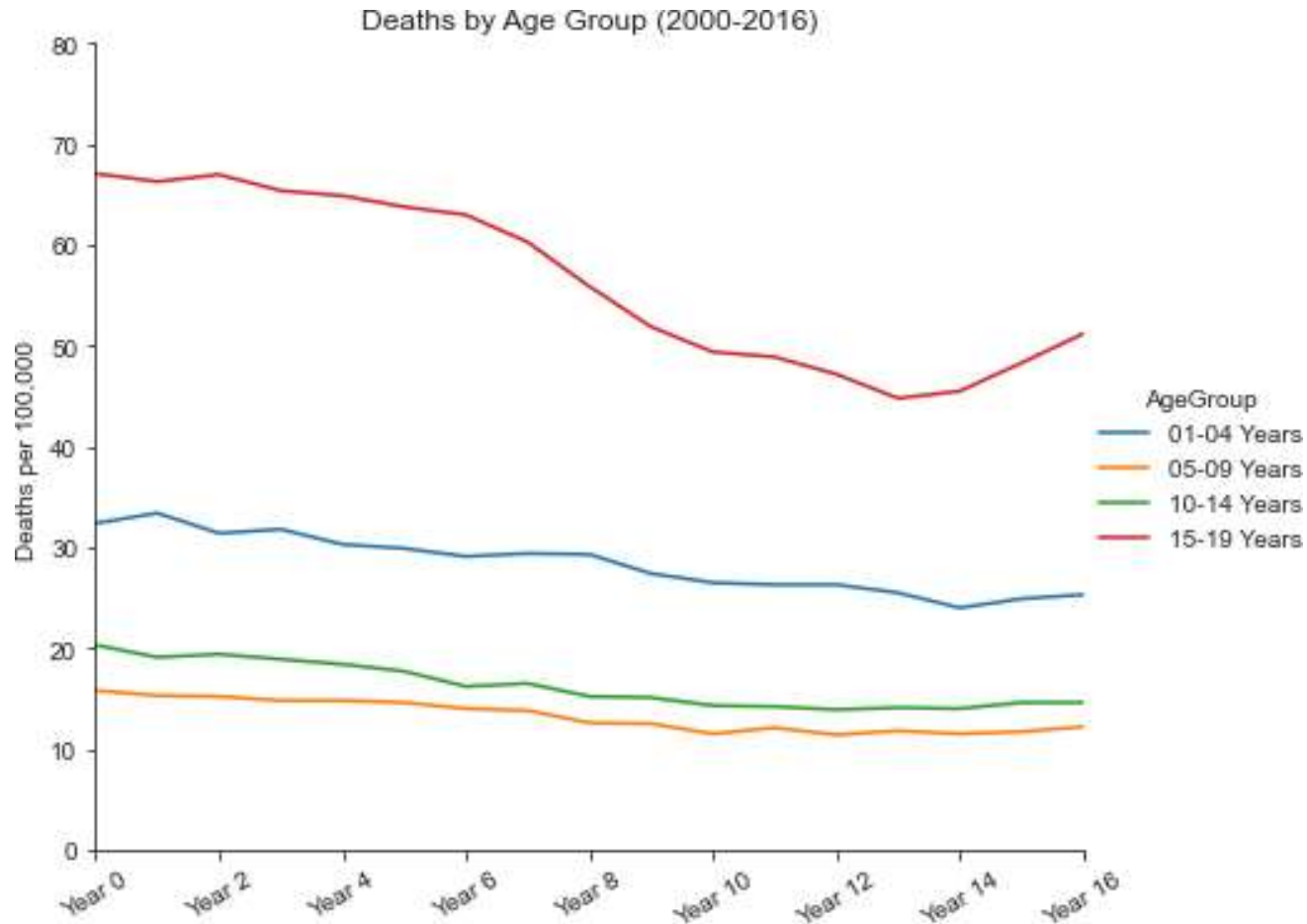
## Some of the other Axes methods

```
set_title()  
set_xlabel()  
set_xticks()  
set_xticklabels()  
set_ylabel()  
set_yticks()  
set_yticklabels()  
tick_params()  
set_xlim()  
set_ylim()
```

## How to enhance a general plot with the Axes methods

```
g = sns.relplot(data=mortality_data, kind='line',
                x='Year', y='DeathRate', hue='AgeGroup',
                aspect=1.25)
for ax in g.axes.flat:
    ax.set_title('Deaths by Age Group (2000-2016)')
    ax.set_xlabel('')
    ax.set_xticks([x for x in range(2000,2017,2)])
    ax.set_xticklabels(['Year ' + str(x) \
                        for x in range(0,17,2)])
    ax.tick_params('x', labelrotation=30)
    ax.set_ylabel('Deaths per 100,000')
    ax.set_xlim(2000,2016)
    ax.set_ylim(0,80)
```

# A plot enhanced with the Axes methods



# The `annotate()` method of the Axes object

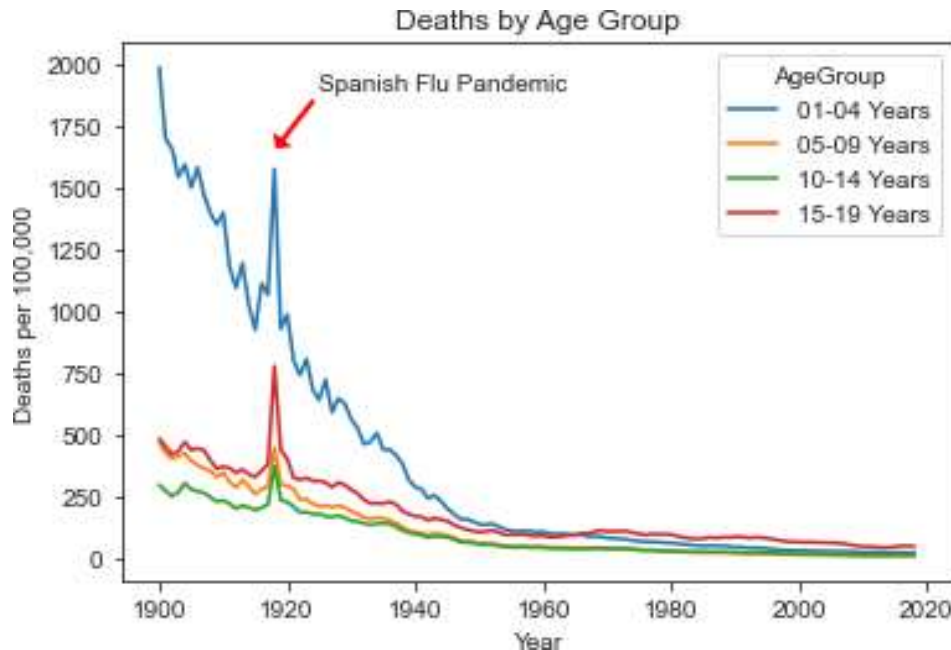
Method	Description
<code>annotate(params)</code>	Adds an annotation to the Axes object.

## Parameters of the `annotate()` method

Parameter	Description
<code>text</code>	The string for the annotation.
<code>xy</code>	The x and y coordinates for the point to annotate.
<code>xytext</code>	The x and y coordinates for the start of the text for the annotation.
<code>arrowprops</code>	The properties of the arrow with measurements in points: facecolor, width, headwidth, and headlength.

# How to add an annotation to a plot

```
ax = sns.lineplot(data=mortality_data,  
                  x='Year', y='DeathRate', hue='AgeGroup')  
ax.set_title('Deaths by Age Group')  
ax.set_ylabel('Deaths per 100,000')  
ax.annotate(text='Spanish Flu Pandemic',  
            xy=(1918, 1650), xytext=(1925, 1900),  
            arrowprops=dict(facecolor='red', width=3, headwidth=12,  
                             headlength=6))
```



# The `color_palette()` and `set_palette()` methods

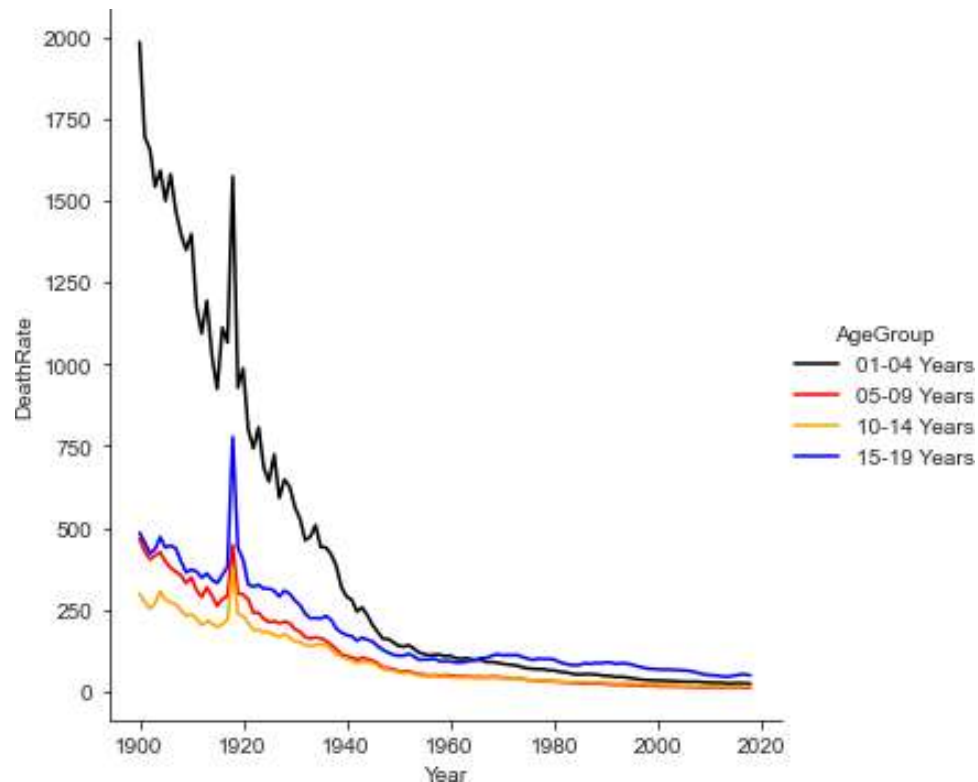
`color_palette(colors)`

`set_palette(palette)`



# How to change the palette for the plots that follow

```
colors = ['black', 'red', 'orange', 'blue']  
sns.set_palette(sns.color_palette(colors))  
sns.relplot(data=mortality_data, kind='line',  
            x='Year', y='DeathRate', hue='AgeGroup')
```



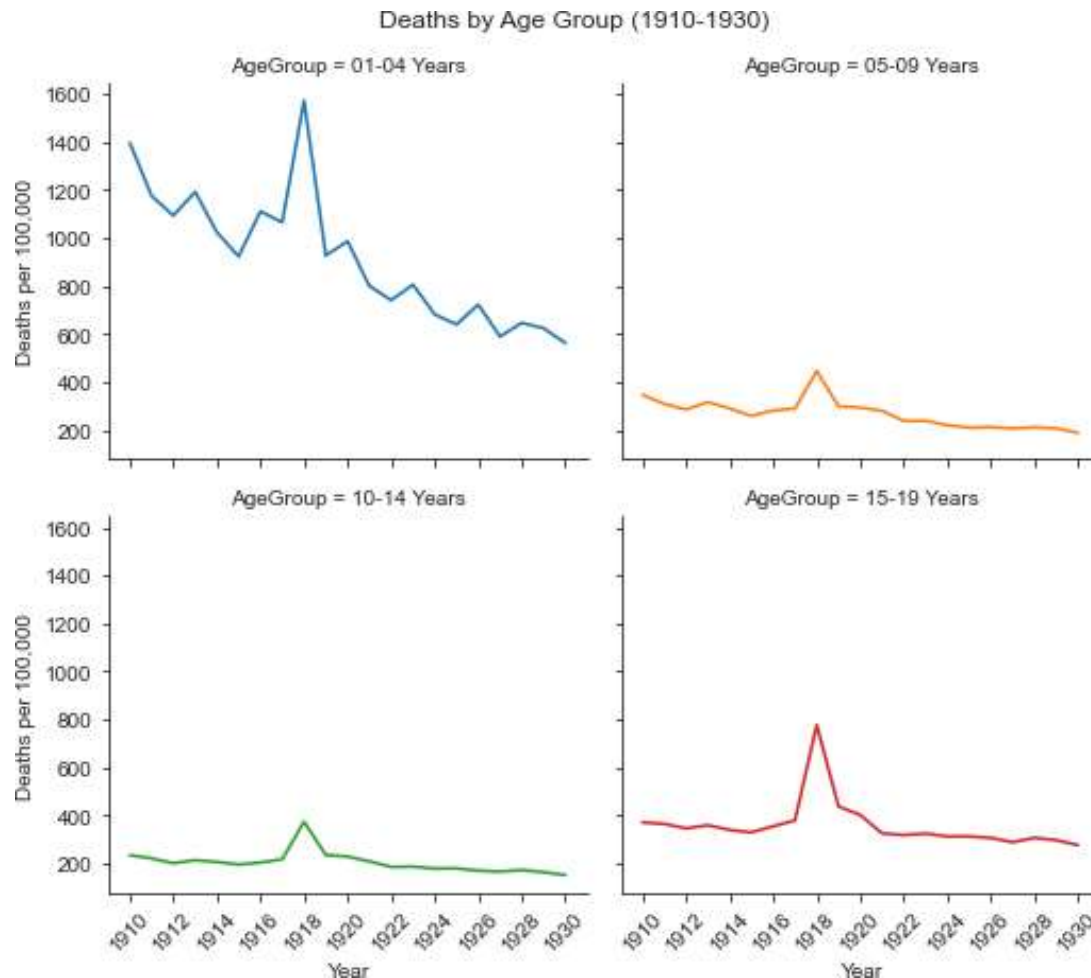
# How to reset the color palette to the default

```
sns.set_palette(sns.color_palette('tab10'))
```

## How to enhance a plot that has subplots

```
g = sns.relplot(  
    data=mortality_data.query('Year >= 1910 and Year <= 1930'),  
    kind='line', x='Year', y='DeathRate', hue='AgeGroup', legend=False,  
    col='AgeGroup', col_wrap=2, height=3, aspect=1.2)  
g.fig.suptitle('Deaths by Age Group (1910-1930)', y=1.025)  
for ax in g.axes.flat:  
    ax.set_ylabel('Deaths per 100,000')  
    ax.set_xticks([x for x in range(1910, 1931, 2)])  
    ax.tick_params('x', labelrotation=45)
```

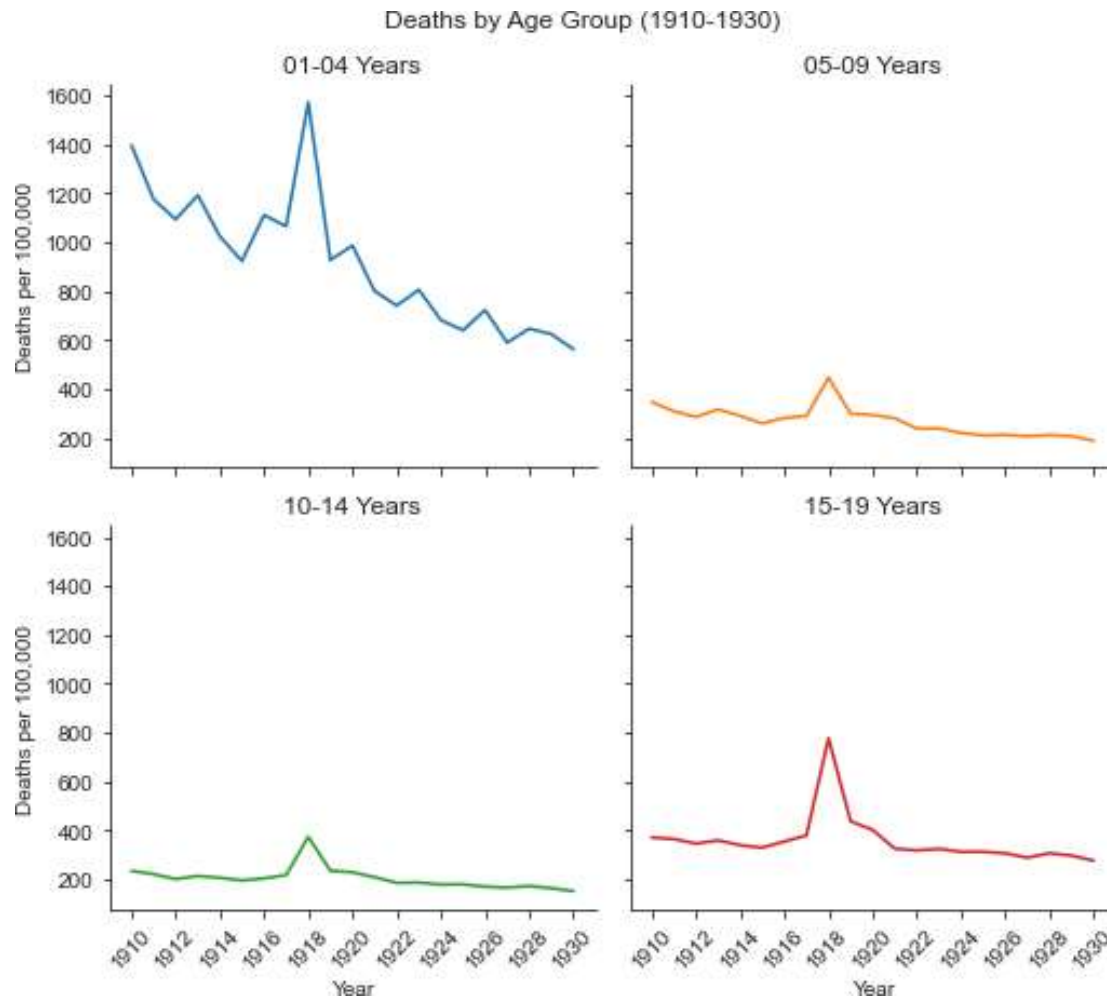
# The enhanced plot with subplots



## How to create specific titles for subplots

```
g = sns.relplot(  
    data=mortality_data.query('Year >= 1910 and Year <= 1930'),  
    kind='line', x='Year', y='DeathRate', hue='AgeGroup', legend=False,  
    col='AgeGroup', col_wrap=2, height=3, aspect=1.2)  
g.fig.suptitle('Deaths by Age Group (1910-1930)', y=1.025)  
ageGroups = mortality_data['AgeGroup'].drop_duplicates().tolist()  
for index, ax in enumerate(g.axes.flat):  
    ax.set_title(ageGroups[index])  
    ax.set_ylabel('Deaths per 100,000')  
    ax.set_xticks([x for x in range(1910, 1931, 2)])  
    ax.tick_params('x', labelrotation=45)
```

# The subplots with specific titles



# The figsize parameter of Matplotlib's subplots() method

Parameter	Description
<b>figsize</b>	Uses a tuple to set the width and height of the figure in inches.

# How to set the figure size of the plot and save the plot

```
# import Matplotlib's pyplot module and set the figure size
import matplotlib.pyplot as plt
fig, ax = plt.subplots(figsize=(10,7.5))

# create the line plot
sns.lineplot(data=mortality_data, x='Year', y='DeathRate',
             hue='AgeGroup')

# use the ax and fig objects to enhance and save the plot
ax.set(title='Deaths by Age Group (1910-1930)',
       ylabel='Deaths per 100,000',
       xticks=[x for x in range(1910, 1931, 2)],
       xlim=(1910,1930), ylim=(0,1750)),
ax.tick_params('x', labelrotation=45)
fig.savefig('deaths_by_age_group_line.png')
```



# The plot with a figure size

