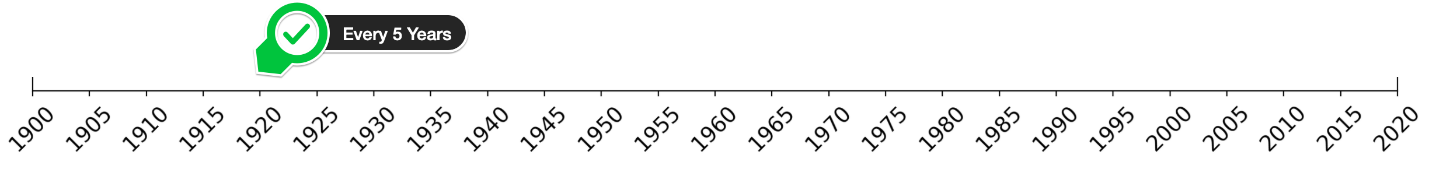
**Using Matplotlib to Visualise Trends over Time**

Now let's look at how things have changed over time. This will give us a chance to review what we learnt about creating charts with two y-axes in Matplotlib and generating arrays with NumPy.

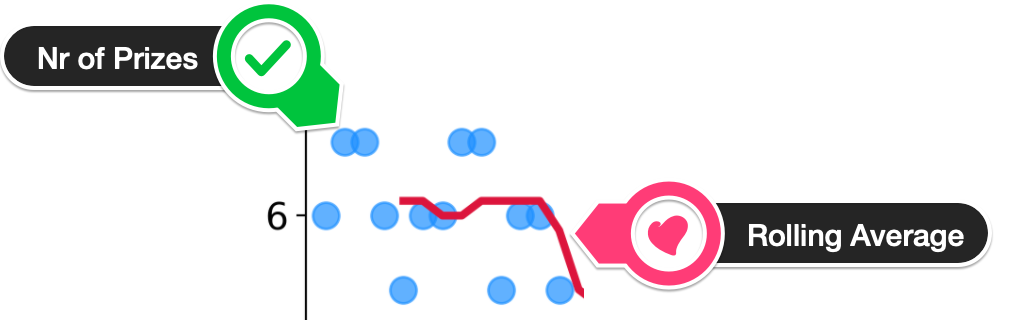
Challenge 1

Are more prizes awarded recently than when the prize was first created? Show the trend in awards visually.

* Count the number of prizes awarded every year.
* Create a 5 year rolling average of the number of prizes (Hint: see previous lessons analysing Google Trends).
* Using Matplotlib superimpose the rolling average on a scatter plot.
* Show a tick mark on the x-axis for every 5 years from 1900 to 2020. (Hint: you'll need to use NumPy).



* Use the [named colours](https://matplotlib.org/3.1.0/gallery/color/named_colors.html) to draw the data points in dogerblue while the rolling average is coloured in crimson.



* Looking at the chart, did the first and second world wars have an impact on the number of prizes being given out?
* What could be the reason for the trend in the chart?

Challenge 2

Investigate if more prizes are shared than before.

* Calculate the average prize share of the winners on a year by year basis.
* Calculate the 5 year rolling average of the percentage share.
* Copy-paste the cell from the chart you created above.
* Modify the code to add a secondary axis to your Matplotlib chart.
* Plot the rolling average of the prize share on this chart.
* See if you can invert the secondary y-axis to make the relationship even more clear.

.

.

..

...

..

.

.

**Solution 1: Number of Prizes Awarded over Time**

First, we have to count the number of Nobel prizes that are awarded each year.

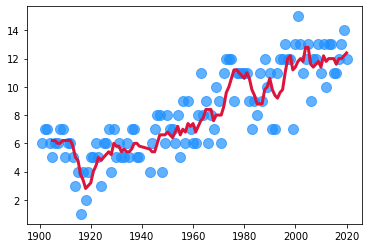
1. prize\_per\_year = df\_data.groupby(by='year').count().prize

This just involves grouping the data so that we can count the number of entries per year. To calculate the 5-year moving average we use .rolling() and .mean() like we did with the Google Trend data.

1. moving\_average = prize\_per\_year.rolling(window=5).mean()

Now we can create a Matplotlib chart that superimposes the two:

1. plt.scatter(x=prize\_per\_year.index,
2. y=prize\_per\_year.values,
3. c='dodgerblue',
4. alpha=0.7,
5. s=100,)
7. plt.plot(prize\_per\_year.index,
8. moving\_average.values,
9. c='crimson',
10. linewidth=3,)
12. plt.show()



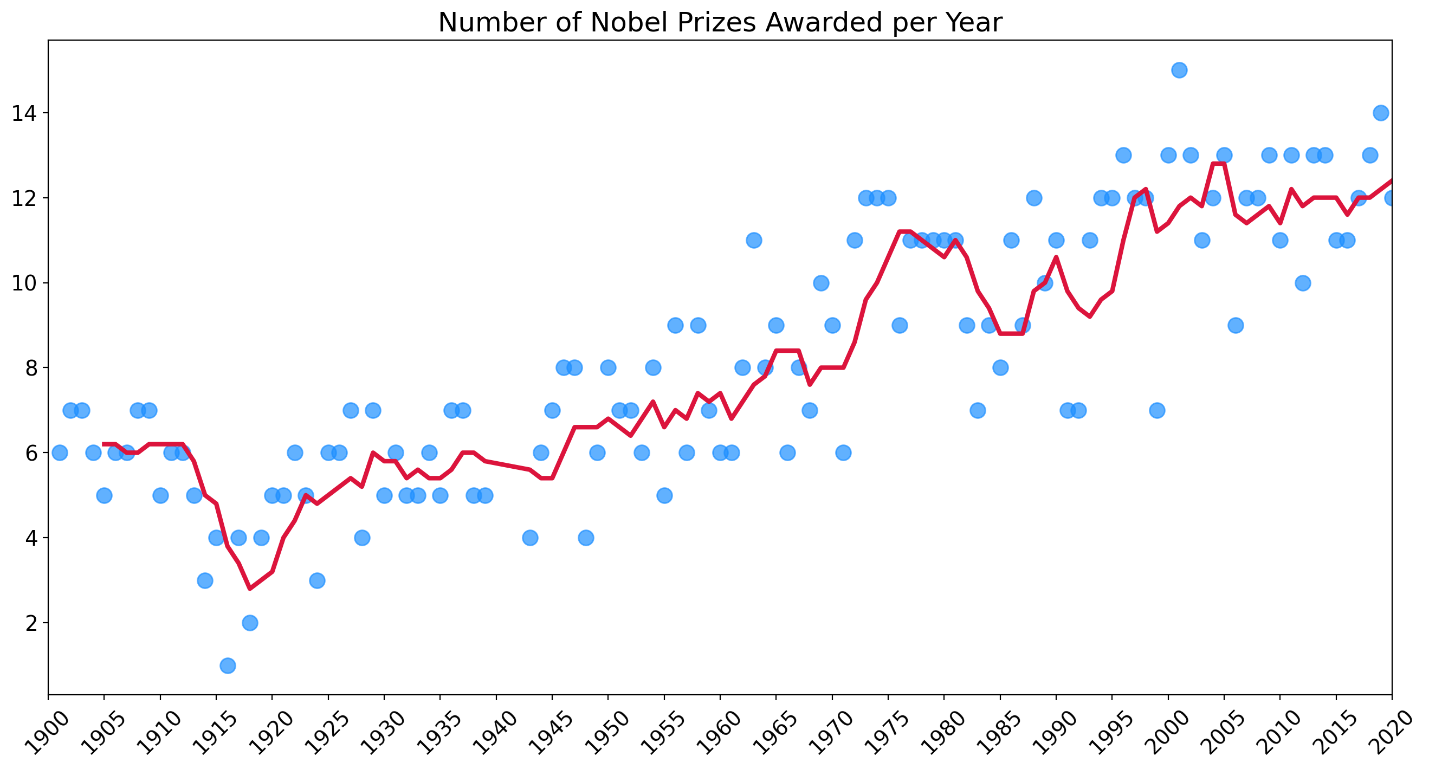
With the help of a little styling, this chart could look better. To create 5-year tick marks on the x-axis, we generate an array using NumPy:

1. np.arange(1900, 2021, step=5)

Then we tap into functions like the .figure(), the .title(), the .xticks(), and .yticks() to fine-tune the chart.

In addition, we will shortly be adding a second y-axis, so we can use an Axes object to draw our scatter and line plots.

1. plt.figure(figsize=(16,8), dpi=200)
2. plt.title('Number of Nobel Prizes Awarded per Year', fontsize=18)
3. plt.yticks(fontsize=14)
4. plt.xticks(ticks=np.arange(1900, 2021, step=5),
5. fontsize=14,
6. rotation=45)
8. ax = plt.gca() # get current axis
9. ax.set\_xlim(1900, 2020)
11. ax.scatter(x=prize\_per\_year.index,
12. y=prize\_per\_year.values,
13. c='dodgerblue',
14. alpha=0.7,
15. s=100,)
17. ax.plot(prize\_per\_year.index,
18. moving\_average.values,
19. c='crimson',
20. linewidth=3,)
22. plt.show()



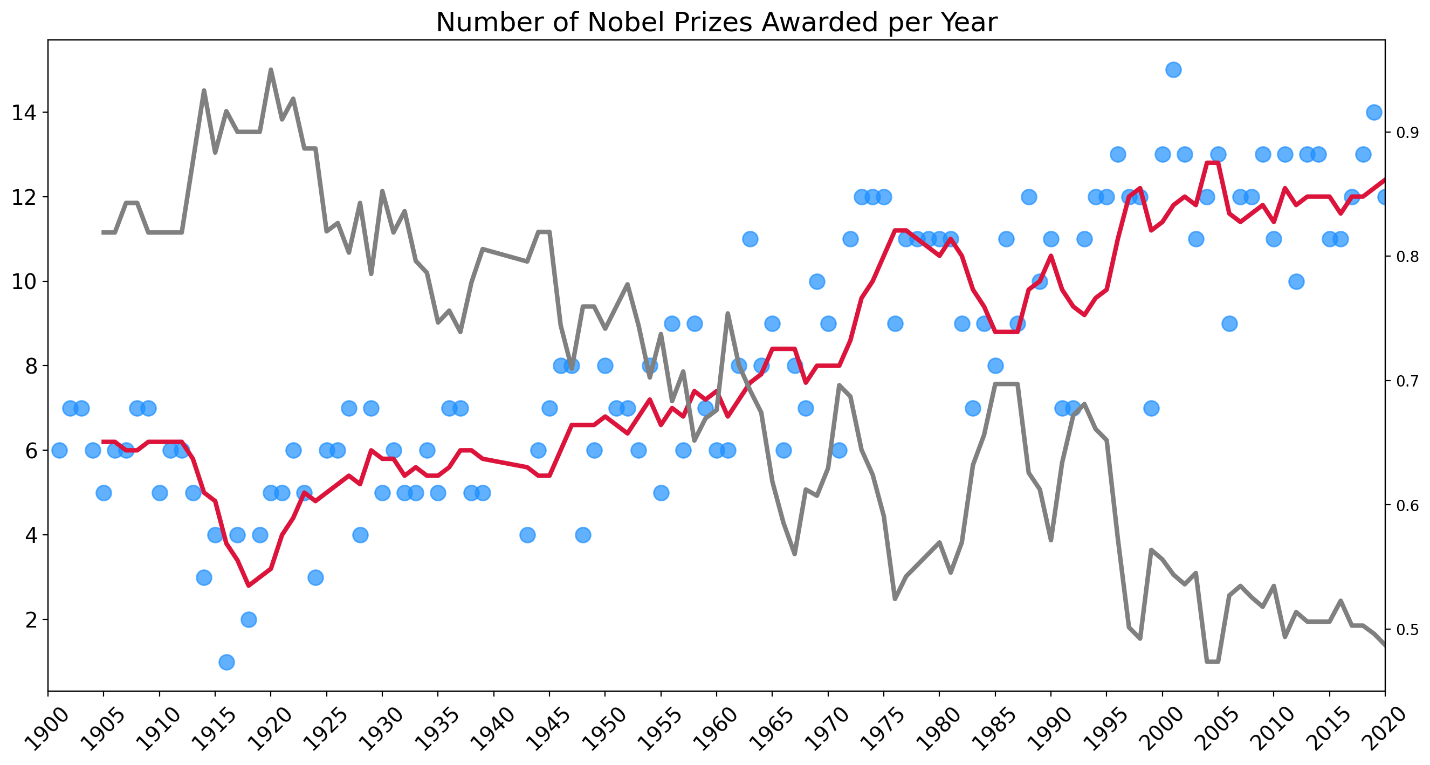
**Solution 2: The Prize Share of Laureates over Time**

Now we can work out the rolling average of the percentage share of the prize. If more prizes are given out, perhaps it is because the prize is split between more people.

1. yearly\_avg\_share = df\_data.groupby(by='year').agg({'share\_pct': pd.Series.mean})
2. share\_moving\_average = yearly\_avg\_share.rolling(window=5).mean()

If more people get the prize, then the average share should go down, right?

1. plt.figure(figsize=(16,8), dpi=200)
2. plt.title('Number of Nobel Prizes Awarded per Year', fontsize=18)
3. plt.yticks(fontsize=14)
4. plt.xticks(ticks=np.arange(1900, 2021, step=5),
5. fontsize=14,
6. rotation=45)
8. ax1 = plt.gca()
9. ax2 = ax1.twinx() # create second y-axis
10. ax1.set\_xlim(1900, 2020)
12. ax1.scatter(x=prize\_per\_year.index,
13. y=prize\_per\_year.values,
14. c='dodgerblue',
15. alpha=0.7,
16. s=100,)
18. ax1.plot(prize\_per\_year.index,
19. moving\_average.values,
20. c='crimson',
21. linewidth=3,)
23. # Adding prize share plot on second axis
24. ax2.plot(prize\_per\_year.index,
25. share\_moving\_average.values,
26. c='grey',
27. linewidth=3,)
29. plt.show()



To see the relationship between the number of prizes and the laureate share even more clearly we can invert the second y-axis.

1. plt.figure(figsize=(16,8), dpi=200)
2. plt.title('Number of Nobel Prizes Awarded per Year', fontsize=18)
3. plt.yticks(fontsize=14)
4. plt.xticks(ticks=np.arange(1900, 2021, step=5),
5. fontsize=14,
6. rotation=45)
8. ax1 = plt.gca()
9. ax2 = ax1.twinx()
10. ax1.set\_xlim(1900, 2020)
12. # Can invert axis
13. ax2.invert\_yaxis()
15. ax1.scatter(x=prize\_per\_year.index,
16. y=prize\_per\_year.values,
17. c='dodgerblue',
18. alpha=0.7,
19. s=100,)
21. ax1.plot(prize\_per\_year.index,
22. moving\_average.values,
23. c='crimson',
24. linewidth=3,)
26. ax2.plot(prize\_per\_year.index,
27. share\_moving\_average.values,
28. c='grey',
29. linewidth=3,)
31. plt.show()

What do we see on the chart? Well, there is clearly an upward trend in the number of prizes being given out as more and more prizes are shared. Also, more prizes are being awarded from 1969 onwards because of the addition of the economics category. We also see that very few prizes were awarded during the first and second world wars. Note that instead of there being a zero entry for those years, we instead see the effect of the wards as missing blue dots.

