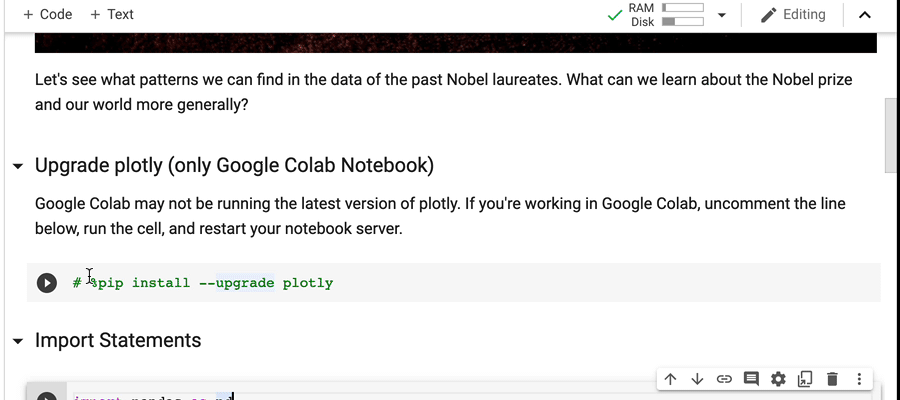
**Update Packages in Google Colab & Explore and Clean the Dataset**

**Using Google Colab? -> Upgrade plotly**

Google Colab comes with a lot of Python packages pre-installed and working. However, you can also install new packages as well as upgrade existing packages. In our case, the plotly package is outdated and will cause a problem with our starburst chart. Here's how to updated a package in Google Colab:



Uncomment the cell with the pip install command. Run the cell. And then comment the code out again.

Challenge 1

Preliminary data exploration.

* What is the shape of df\_data? How many rows and columns?
* What are the column names and what kind of data is inside of them?
* In which year was the Nobel prize first awarded?
* Which year is the latest year included in the dataset?

Challenge 2

* Are there any duplicate values in the dataset?
* Are there NaN values in the dataset?
* Which columns tend to have NaN values?
* How many NaN values are there per column?
* Why do these columns have NaN values?

Challenge 3

* Convert the birth\_date column to Pandas Datetime objects
* Add a Column called share\_pct which has the laureates' share as a percentage in the form of a floating-point number.

.

.

..

...

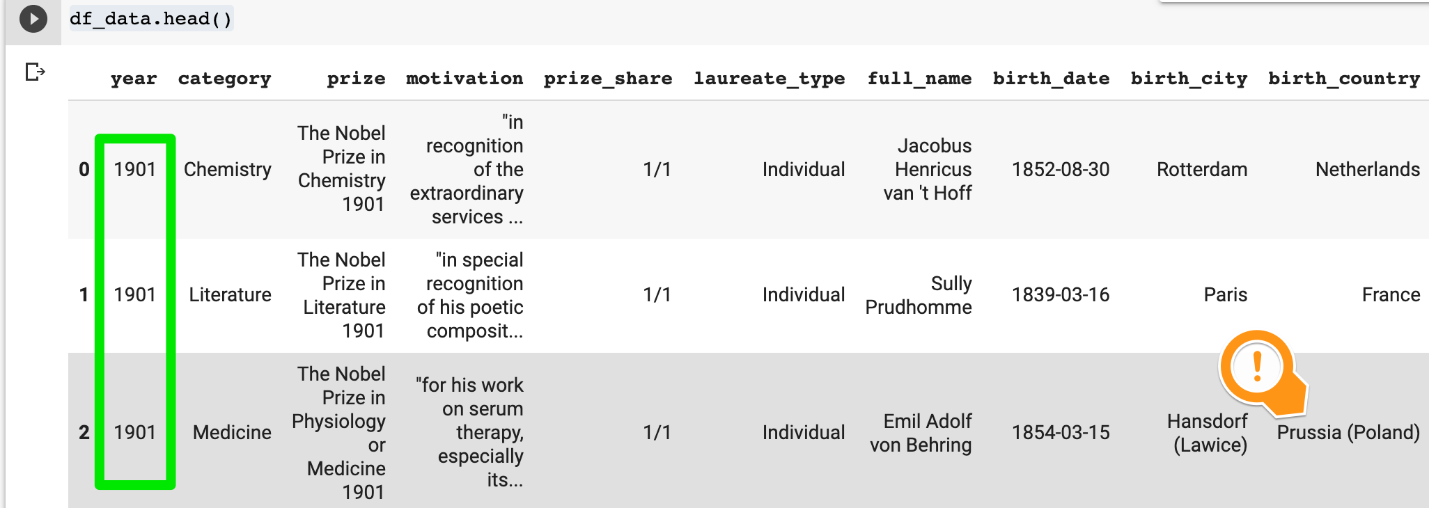
..

.

.

**Solution 1: Preliminary Data Exploration**

When we run df\_data.shape, df\_data.tail(), and df\_data.head(), we see that there are 962 rows and 16 columns. The first Nobel prizes were awarded in 1901 and the data goes up to 2020.



We notice that the columns contain the following information:

**birth\_date**: date in string format

**motivation**: description of what the prize is for

**prize\_share**: given as a fraction

**laureate\_type:**individual or organisation

**birth\_country**: has countries that no longer exist

**birth\_country\_current**: current name of the country where the birth city is located

**ISO:**three-letter international country code

**organization\_name**: research institution where the discovery was made

**organization\_city:**location of the institution

**Solution 2: NaN values**

There are no duplicates in the dataset:

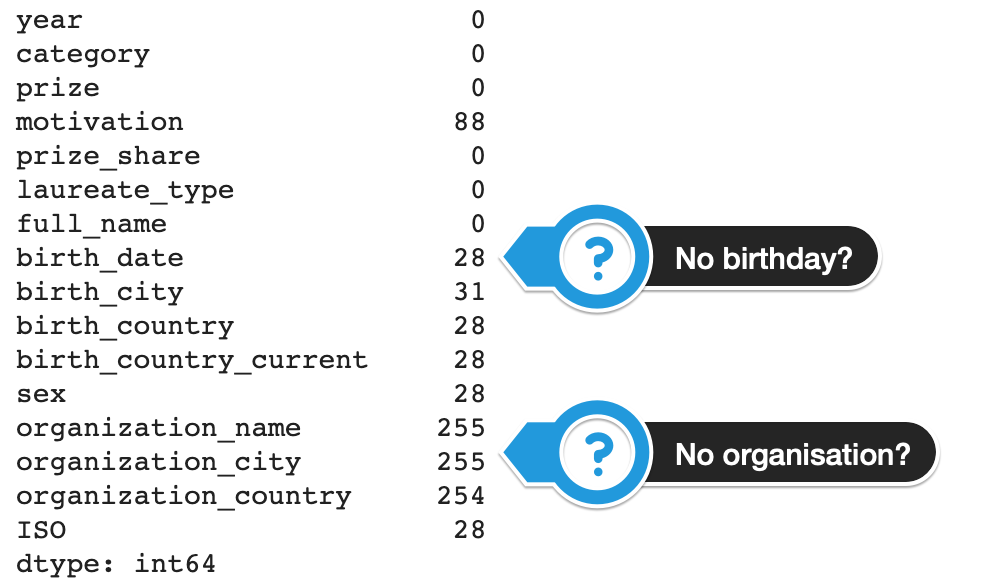
1. print(f'Any duplicates? {df\_data.duplicated().values.any()}')

However, there are a number of NaN values

1. print(f'Any NaN values among the data? {df\_data.isna().values.any()}')

We can get a count of the NaN values per column using

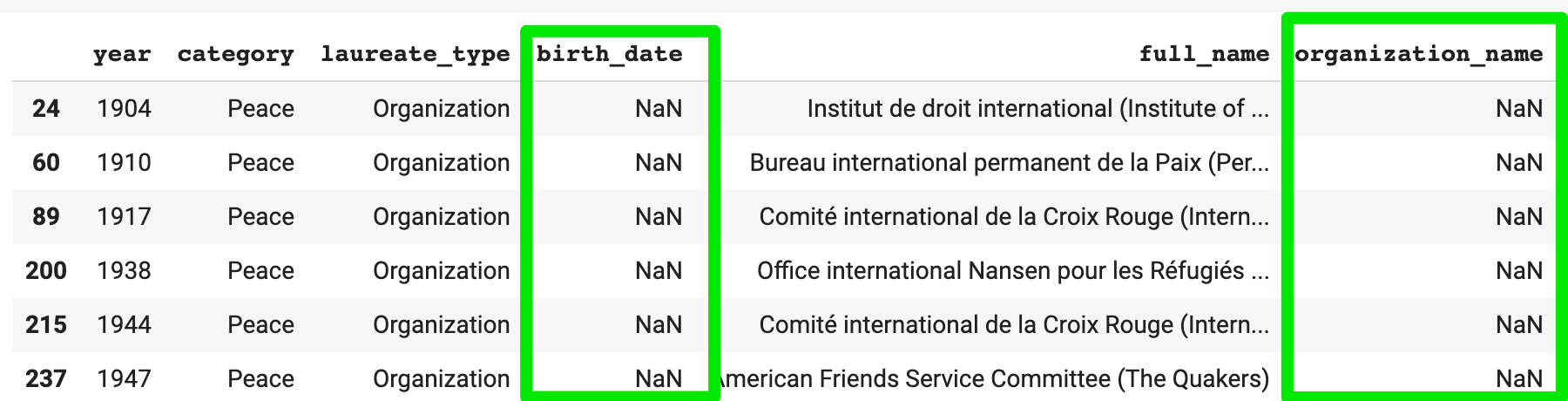
1. df\_data.isna().sum()



Why are there so many NaN values for the birth date? And why are there so many missing values among the organisation columns?

Filtering on the NaN values in the birth date column we see that we get back a bunch of organisations, like the UN or the Red Cross.

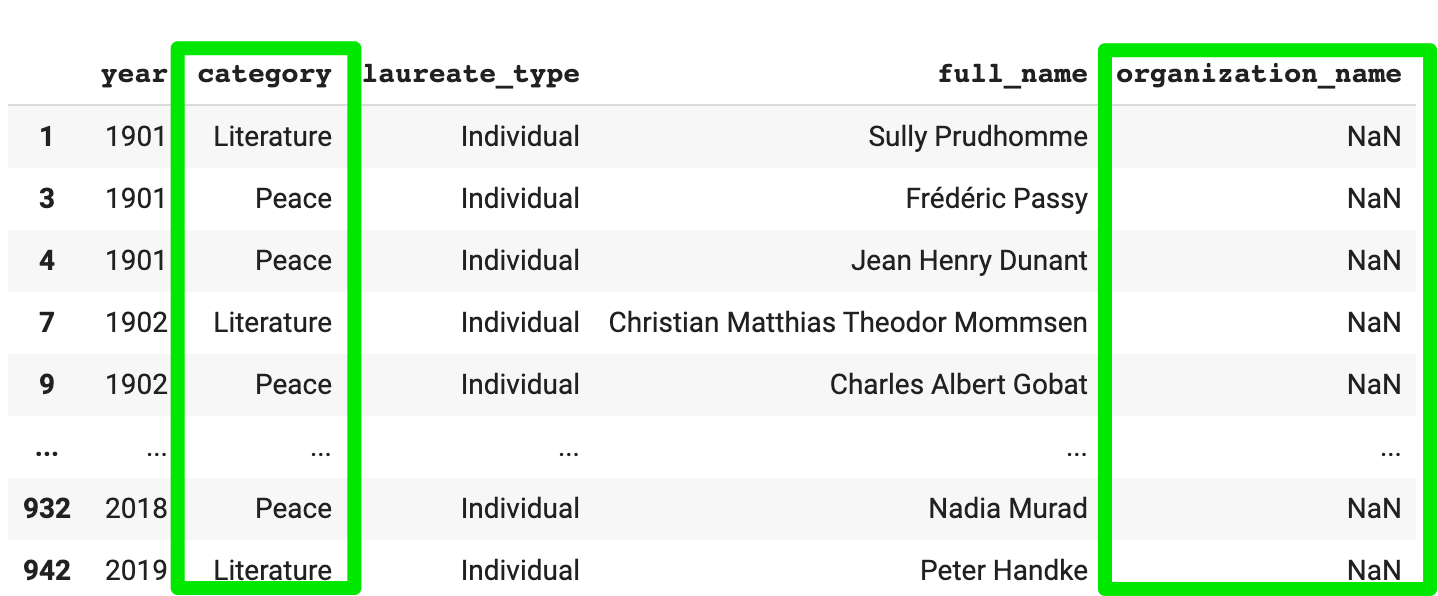
1. col\_subset = ['year','category', 'laureate\_type',
2. 'birth\_date','full\_name', 'organization\_name']
3. df\_data.loc[df\_data.birth\_date.isna()][col\_subset]



That makes sense. We also see that since the organisation's name is in the full\_name column, the organisation\_name column contains NaN.

In addition, when we look at for rows where the organization\_name column has no value, we also see that many prizes went to people who were not affiliated with a university or research institute. This includes many of the Literature and Peace prize winners.

1. col\_subset = ['year','category', 'laureate\_type','full\_name', 'organization\_name']
2. df\_data.loc[df\_data.organization\_name.isna()][col\_subset]



**Solution 3: Type Conversions**

We can use pandas to convert the birth\_date to a Datetime object with a single line:

1. df\_data.birth\_date = pd.to\_datetime(df\_data.birth\_date)

Adding a column that contains the percentage share as first requires that we split the string on the forward slash. Then we can convert to a number. And finally, we can do the division.

1. separated\_values = df\_data.prize\_share.str.split('/', expand=True)
2. numerator = pd.to\_numeric(separated\_values[0])
3. denomenator = pd.to\_numeric(separated\_values[1])
4. df\_data['share\_pct'] = numerator / denomenator

Now we can check if our type conversions were successful:

