info1111

May 25, 2024

[1]:
```
from IPython.display import HTML
HTML('''
    <style> body {font-family: "Roboto Condensed Light", "Roboto Condensed";}␣
 ↪h2 {padding: 10px 12px; background-color: #E64626; position: static; color:␣
 ↪#ffffff; font-size: 40px;} .text_cell_render p { font-size: 15px; } .
 ↪text_cell_render h1 { font-size: 30px; } h1 {padding: 10px 12px;␣
 ↪background-color: #E64626; color: #ffffff; font-size: 40px;} .
 ↪text_cell_render h3 { padding: 10px 12px; background-color: #0148A4;␣
 ↪position: static; color: #ffffff; font-size: 20px;} h4:before{
    content: "@"; font-family:"Wingdings"; font-style:regular; margin-right:␣
 ↪4px;} .text_cell_render h4 {padding: 8px; font-family: "Roboto Condensed␣
 ↪Light"; position: static; font-style: italic; background-color: #FFB800;␣
 ↪color: #ffffff; font-size: 18px; text-align: center; border-radius: 5px;
 ↪}input[type=submit] {background-color: #E64626; border: solid; border-color:␣
 ↪#734036; color: white; padding: 8px 16px; text-decoration: none; margin: 4px␣
 ↪2px; cursor: pointer; border-radius: 20px;}</style>
''')
```

[1]:
```
<IPython.core.display.HTML object>
```

1 Data Analysis with Python and Jupyter Notebook

In this notebook, we will taking a look at how to analyse data into python libraries such as pandas and numpy.

When creating the notebook, we set the kernel to python. We start by importing the required libraries. We can see this in the code block below.

[14]:
```
import pandas as pd
import numpy as np
```

Once our libraries have been imported. We can start working on our dataset. For this notebook, we will use a csv file we found online. Before executing the next command, make sure that the csv file is in the same folder as the notebook.

We will now use a magic command to get the directory of where the notebook is stored.

[15]:
```
current_directory = %pwd
```

Now we read the file from the same directory as the notebook.

[16]:
```
df = pd.read_csv(current_directory + "/people.csv")
```

Once our data is loaded into python, we can use some functions of pandas library such as .head() to show the first 5 values and .info() to show some info abou each column.

[19]:
```
df.head()
```

[19]:
```
   Index          User Id First Name Last Name     Sex  \
0      1  8717bbf45cCDbEe     Shelia   Mahoney    Male
1      2  3d5AD30A4cD38ed         Jo    Rivers  Female
2      3  810Ce0F276Badec     Sheryl    Lowery  Female
3      4  BF2a889C00f0cE1    Whitney    Hooper    Male
4      5  9afFEafAe1CBBB9    Lindsey      Rice  Female

                           Email               Phone Date of birth  \
0            pwarner@example.org        857.139.8239    2014-01-27
1  fergusonkatherine@example.net     +1-950-759-8687    1931-07-26
2            fhoward@example.org       (599)782-0605    2013-11-25
3          zjohnston@example.com     +1-939-130-6258    2012-11-17
4              elin@example.net  (390)417-1635x3010    1923-04-15

                 Job Title
0        Probation officer
1                   Dancer
2                     Copy
3  Counselling psychologist
4       Biomedical engineer
```

[20]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 9 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Index          1000 non-null   int64
 1   User Id        1000 non-null   object
 2   First Name     1000 non-null   object
 3   Last Name      1000 non-null   object
 4   Sex            1000 non-null   object
 5   Email          1000 non-null   object
 6   Phone          1000 non-null   object
 7   Date of birth  1000 non-null   object
 8   Job Title      1000 non-null   object
dtypes: int64(1), object(8)
memory usage: 70.4+ KB
```

The next step in working with data is data cleaning. This can also be done with the python library pandas. We can use the .dropna() which drops all null values.

[21]:
```
df = df.dropna()
```

Once our data is clean, we start performing some basic data manipulation tasks. We will start by converting the Date of birth column from type object to type datetime.

[22]:
```
df['Date of birth'] = pd.to_datetime(df['Date of birth'], errors='coerce')
```

After this we will sort our data from to only include people born after 2000. We can do this with a simple panda command.

[23]:
```
df = df[df['Date of birth'] > '2000-01-01']
```

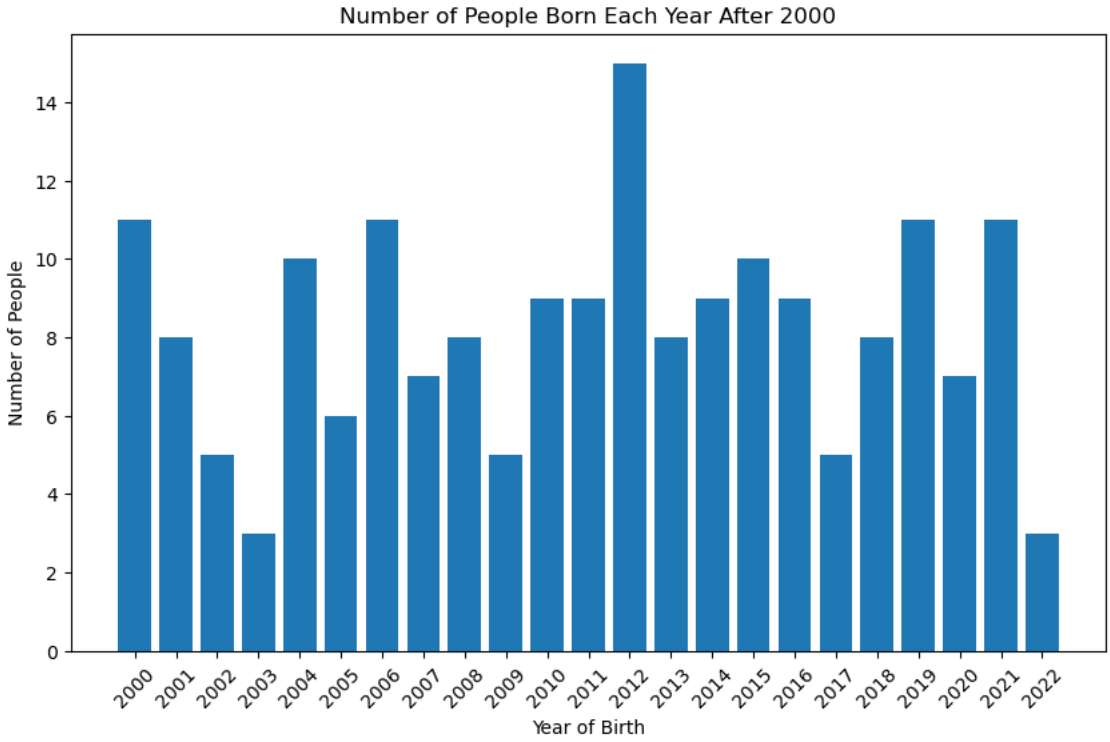
Now that we have some data, we can use another library in python to visualize our data. In the next code block, we will start by importing a function from the matplotlib library.

[24]:
```
import matplotlib.pyplot as plt
```

Now, we will use the dataframe we have cleaned and performed operations on so far and we will visualize data from it. We can analyze trends and understand the data better. We need to keep in mind that this graph only represents the number of people born after 2000 in our dataset and this is not representative for the whole population.

[25]:
```
df['Year of Birth'] = df['Date of birth'].dt.year
birth_year_counts = df['Year of Birth'].value_counts().sort_index()

plt.figure(figsize=(10, 6))
plt.bar(birth_year_counts.index, birth_year_counts.values)
plt.title('Number of People Born Each Year After 2000')
plt.xlabel('Year of Birth')
plt.ylabel('Number of People')
plt.xticks(birth_year_counts.index, rotation=45)
plt.show()
```



In this notebook, we saw how to go from data ingestion to data visualisation with python. We can do this for various different datasets and we also saw how effective Jupyter Notebooks are at teaching new concepts due its blend of markdown, code, and visualisation.

This artefact demonstartes one of the many practical use cases foe Jupyter Notebooks.

Finally, we will convert our notebook to a pdf format.

[13]:
```
# Convert the current notebook to PDF using nbconvert
!jupyter nbconvert --to pdf info1111.ipynb
```

```
[NbConvertApp] Converting notebook info1111.ipynb to pdf
/Users/aaryanbansal/anaconda3/lib/python3.11/site-
packages/nbconvert/utils/pandoc.py:51: RuntimeWarning: You are using an
unsupported version of pandoc (3.1.12.3).
Your version must be at least (1.12.1) but less than (3.0.0).
Refer to https://pandoc.org/installing.html.
Continuing with doubts…
  check_pandoc_version()
[NbConvertApp] Support files will be in info1111_files/
[NbConvertApp] Making directory ./info1111_files
[NbConvertApp] Writing 32904 bytes to notebook.tex
[NbConvertApp] Building PDF
[NbConvertApp] Running xelatex 3 times: ['xelatex', 'notebook.tex', '-quiet']
[NbConvertApp] Running bibtex 1 time: ['bibtex', 'notebook']
```

```
[NbConvertApp] WARNING | bibtex had problems, most likely because there were no
citations
[NbConvertApp] PDF successfully created
[NbConvertApp] Writing 62366 bytes to info1111.pdf
```

[ ]: