importing libraries

```
import pandas as pd
```

reading only required data

```
df = pd.read_csv("https://raw.githubusercontent.com/ieee8023/covid-chestxray-dataset/refs/heads/master/me
```

selecting required columns

```
df
```

	view	filename
0	PA	auntminnie-a-2020_01_28_23_51_6665_2020_01_28_...
1	PA	auntminnie-b-2020_01_28_23_51_6665_2020_01_28_...
2	PA	auntminnie-c-2020_01_28_23_51_6665_2020_01_28_...
3	PA	auntminnie-d-2020_01_28_23_51_6665_2020_01_28_...
4	PA	nejmc2001573_f1a.jpeg
...	...	...
945	AP	072ecaf8c60a81980abb57150a8016_jumbo-9.jpeg
946	AP	ff33c406392b968d483174c97eb857_jumbo-9.jpeg
947	PA	000001-266.jpg
948	AP	000001-272.jpg
949	L	000002-268.jpg

950 rows × 2 columns

Next steps: Generate code with df | New interactive sheet

metadata info

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 950 entries, 0 to 949
Data columns (total 2 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   view      950 non-null    object
 1   filename  950 non-null    object
dtypes: object(2)
memory usage: 15.0+ KB
```

metadata description

```
df.describe()
```

	view	filename
count	950	950
unique	7	950
top	PA	000002-268.jpg
freq	344	1

```
df.view.value_counts()
```

view	count
PA	344
AP Supine	234
AP	203
L	84
Axial	68
Coronal	16
AP Erect	1

dtype: int64

selecting only `AP group` and `PA` views

```
df = df[df.view.isin(["AP","AP Supine","AP Erect","PA"])].copy()
```

changing all AP groups' view to `AP`

```
df.loc[df["view"].isin(["AP Supine","AP Erect"]),"view"] = "AP"
```

selecting only `AP` and `PA` value views

```
final_df = df
final_df
```

	view	filename
0	PA	auntminnie-a-2020_01_28_23_51_6665_2020_01_28_...
1	PA	auntminnie-b-2020_01_28_23_51_6665_2020_01_28_...
2	PA	auntminnie-c-2020_01_28_23_51_6665_2020_01_28_...
3	PA	auntminnie-d-2020_01_28_23_51_6665_2020_01_28_...
4	PA	nejmc2001573_f1a.jpeg
...	...	...
944	AP	d2c8a74b37d8d1581ea2a8fe865ef3_jumbo-10.jpeg
945	AP	072ecaf8c60a81980abb57150a8016_jumbo-9.jpeg
946	AP	ff33c406392b968d483174c97eb857_jumbo-9.jpeg
947	PA	000001-266.jpg
948	AP	000001-272.jpg

782 rows × 2 columns

Next steps: Generate code with `df` New interactive sheet

```
final_df.info()
final_df.describe()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 782 entries, 0 to 948
Data columns (total 2 columns):
 #   Column    Non-Null Count  Dtype 
---  ------    --------------  ----- 
 0   view      782 non-null    object
 1   filename  782 non-null    object
dtypes: object(2)
memory usage: 18.3+ KB
```

	view	filename
count	782	782
unique	2	782
top	AP	000001-272.jpg
freq	438	1

```
ap = df[df["view"] == "AP"]
pa = df[df["view"] == "PA"]
ap_filename = ap["filename"]
pa_filename = pa["filename"]
```

```
print("AP dataframe")
ap.info()
print("\n")
print("PA dataframe")
pa.info()
```

```
AP dataframe
<class 'pandas.core.frame.DataFrame'>
Index: 438 entries, 9 to 948
Data columns (total 2 columns):
 #   Column    Non-Null Count  Dtype 
---  ------    --------------  ----- 
 0   view      438 non-null    object
 1   filename  438 non-null    object
dtypes: object(2)
memory usage: 10.3+ KB


PA dataframe
<class 'pandas.core.frame.DataFrame'>
Index: 344 entries, 0 to 947
Data columns (total 2 columns):
 #   Column    Non-Null Count  Dtype 
---  ------    --------------  ----- 
 0   view      344 non-null    object
 1   filename  344 non-null    object
dtypes: object(2)
memory usage: 8.1+ KB
```

mounting google drive

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive"
```

```
import urllib
import os
from concurrent.futures import ThreadPoolExecutor
from functools import partial
import os
```

making directories to save images

```
baseDir = "/content/drive/MyDrive/Colab Notebooks/dm"

saveDirs = ["/pract2_images","/pract2_images/AP","/pract2_images/PA"]
for sd in saveDirs:
    os.makedirs(baseDir+sd,exist_ok=True)
```

downloading images

```
url_base = "https://raw.githubusercontent.com/ieee8023/covid-chestxray-dataset/refs/heads/master/images/"


def downloadImg(imgName:str,imgType:str):
    fullUrl = f"{url_base}{imgName}"
    fileDest = f"{baseDir}/pract2_images/{imgType}/{imgName}"
    if os.path.exists(fileDest):
        return
    try:
        urllib.request.urlretrieve(fullUrl, fileDest)
    except Exception as e:
        return f"Error downloading {imgName}:{e}"
```

parallelly downloading images

```
with ThreadPoolExecutor(max_workers=10) as executor:
    func_with_args = partial(downloadImg,imgType="AP")
    executor.map(func_with_args,ap_filename.tolist())

with ThreadPoolExecutor(max_workers=10) as executor:
    func_with_args = partial(downloadImg,imgType="PA")
    executor.map(func_with_args,pa_filename.tolist())
```

Summary

```
print("Total images in original dataset :",df.filename.count())
print("No of images after preprocessing :",final_df.filename.count())
print("\nDistribution of views\n",final_df.view.value_counts())
```

```
Total images in original dataset : 782
No of images after preprocessing : 782

Distribution of views
 view
AP    438
PA    344
Name: count, dtype: int64
```