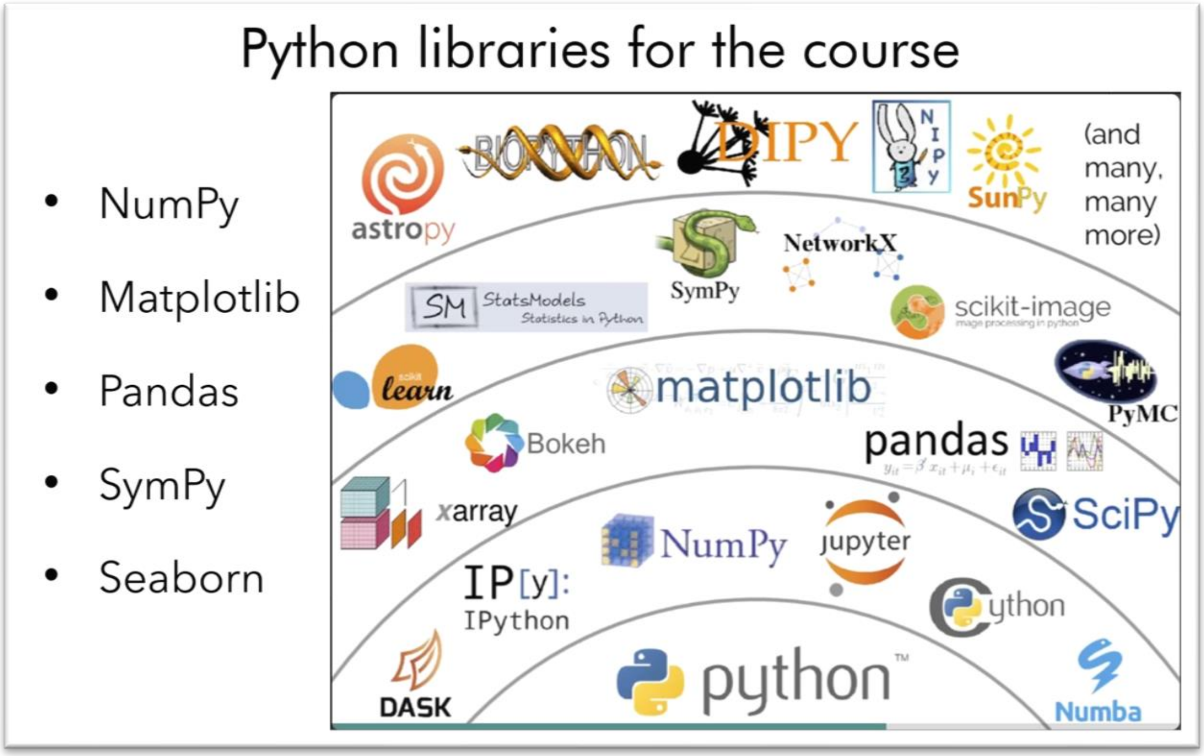
FOAM – Week 07

7.1 Introduction to NumPy, Pandas and Matplotlib



Python libraries for the course

- NumPy
- Matplotlib
- Pandas
- SymPy
- Seaborn

Introducing important python libraries

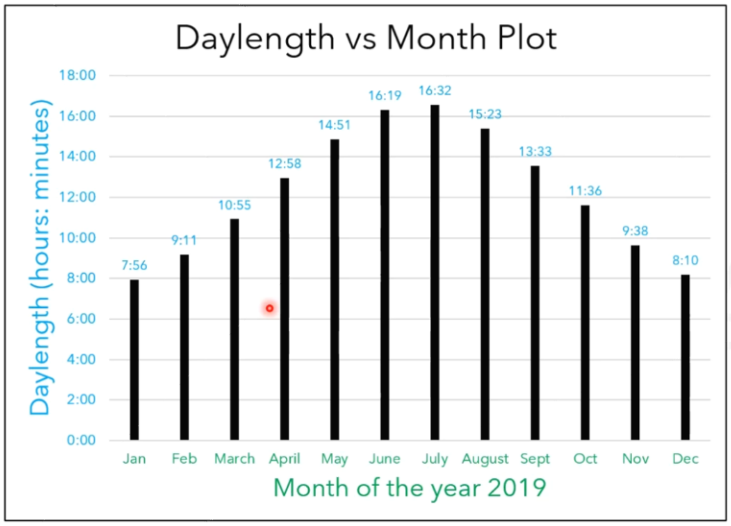
- **Matplotlib**
  - Plotting and visualization package
  - Matplotlib makes easy things easy and hard things possible
- **Pandas**
  - Python library wo work with data sets
  - Has modules for analyzing, cleaning, exploring and manipulating data
- **NumPy**
  - Fundamental package for scientific computing
  - Contains mathematical modules from varied domains
  - Very fast than native python

7.2 Introduction to NumPy, Pandas and Matplotlib

Learning through examples

- Demonstrate use of python libraries by working out same examples
- Write code snippets in simple and detailed way for pedagogical reasons
- Break the fear of programming and mathematics
- **Problem 1:** Seasonal variations of day length
- **Problem 2:** Luminescence of moon phases
- **Problem 3:** Covid data

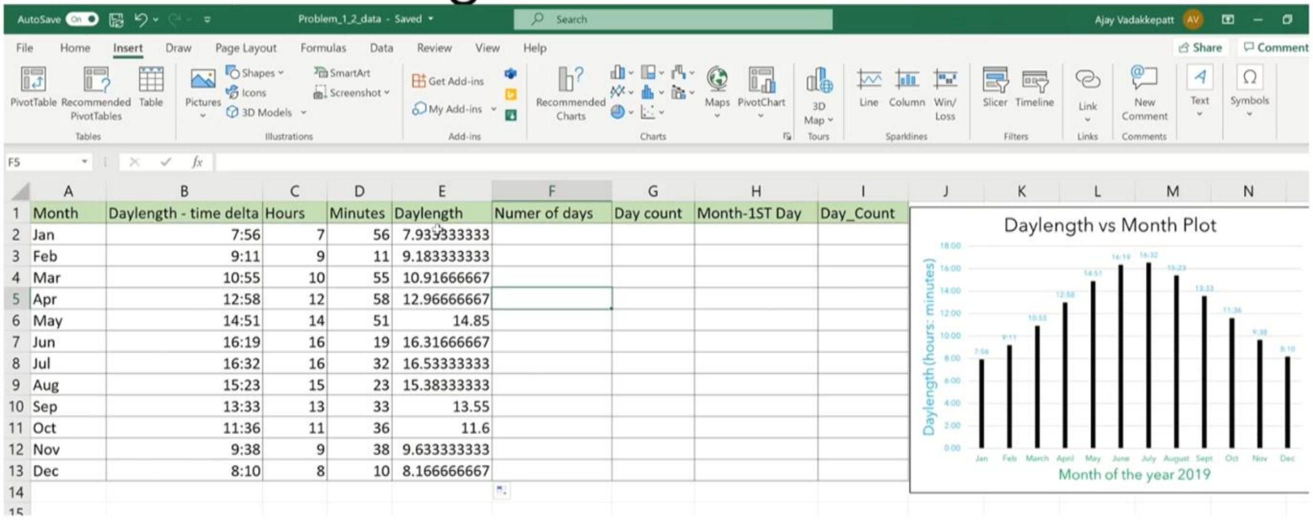
Seasonal variation of day length



Month of Year 2019	Daylength (hours: minutes)
Jan	7:56
Feb	9:11
March	10:55
April	12:58
May	14:51
June	16:19
July	16:32
August	15:23
Sept	13:33
Oct	11:36
Nov	9:38
Dec	8:10

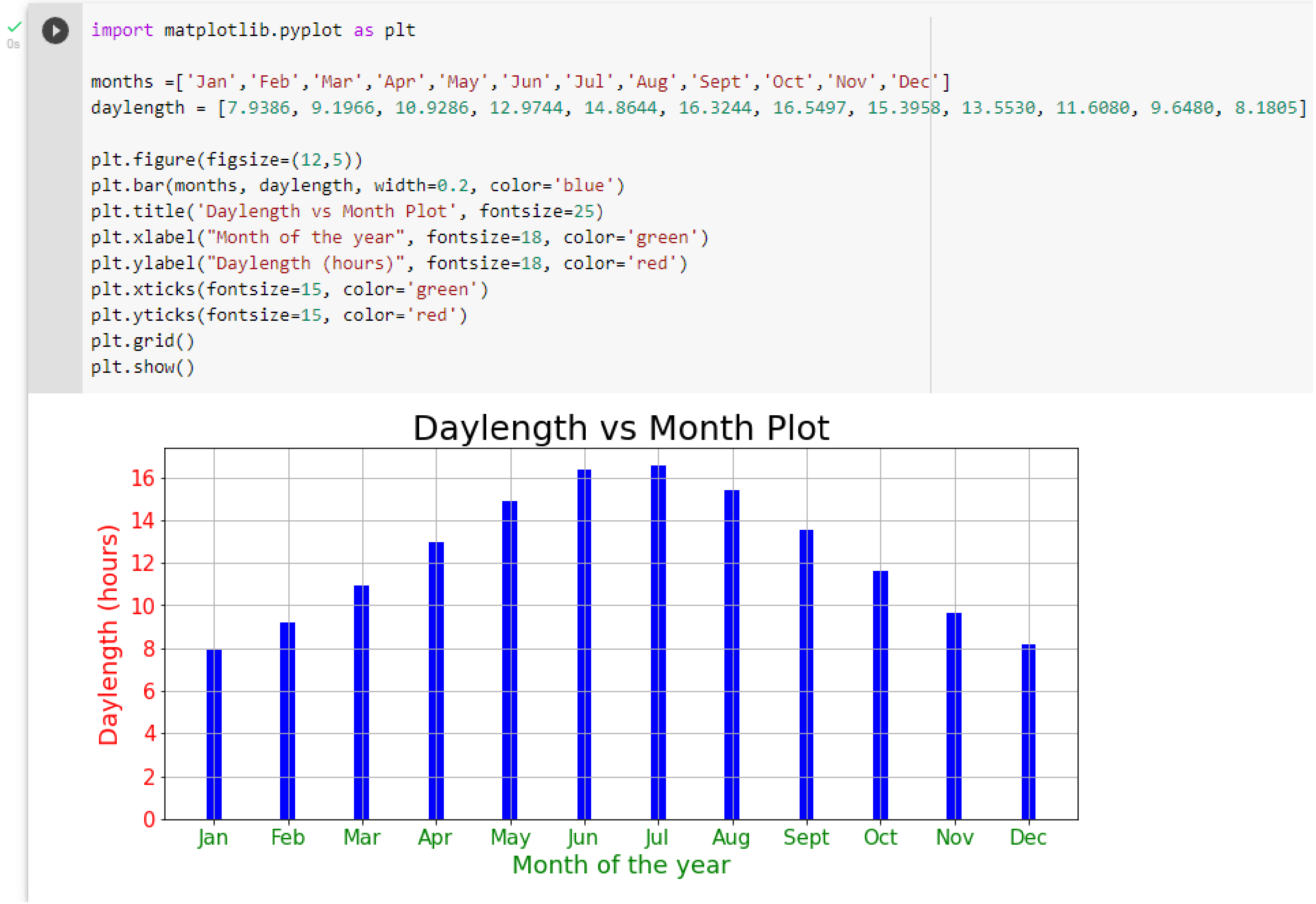
Height of the bar represents the magnitude of daylength

Converting hour: minutes to hours



	A	B	C	D	E	F	G	H	I
1	Month	Daylength - time delta	Hours	Minutes	Daylength	Numer of days	Day count	Month-1ST Day	Day_Count
2	Jan	7:56	7	56	7.933333333				
3	Feb	9:11	9	11	9.183333333				
4	Mar	10:55	10	55	10.91666667				
5	Apr	12:58	12	58	12.96666667				
6	May	14:51	14	51	14.85				
7	Jun	16:19	16	19	16.31666667				
8	Jul	16:32	16	32	16.53333333				
9	Aug	15:23	15	23	15.38333333				
10	Sep	13:33	13	33	13.55				
11	Oct	11:36	11	36	11.6				
12	Nov	9:38	9	38	9.633333333				
13	Dec	8:10	8	10	8.166666667				

Colab –



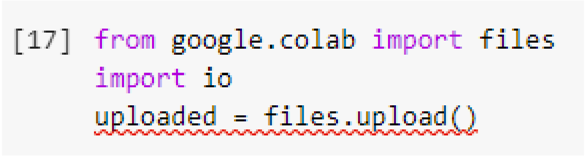
```
import matplotlib.pyplot as plt

months =['Jan','Feb','Mar','Apr','May','Jun','Jul','Aug','Sept','Oct','Nov','Dec']
daylength = [7.9386, 9.1966, 10.9286, 12.9744, 14.8644, 16.3244, 16.5497, 15.3958, 13.5530, 11.6080, 9.6480, 8.1805]

plt.figure(figsize=(12,5))
plt.bar(months, daylength, width=0.2, color='blue')
plt.title('Daylength vs Month Plot', fontsize=25)
plt.xlabel("Month of the year", fontsize=18, color='green')
plt.ylabel("Daylength (hours)", fontsize=18, color='red')
plt.xticks(fontsize=15, color='green')
plt.yticks(fontsize=15, color='red')
plt.grid()
plt.show()
```

7.3 Introduction to NumPy, Pandas and Matplotlib

Importing data from Excel

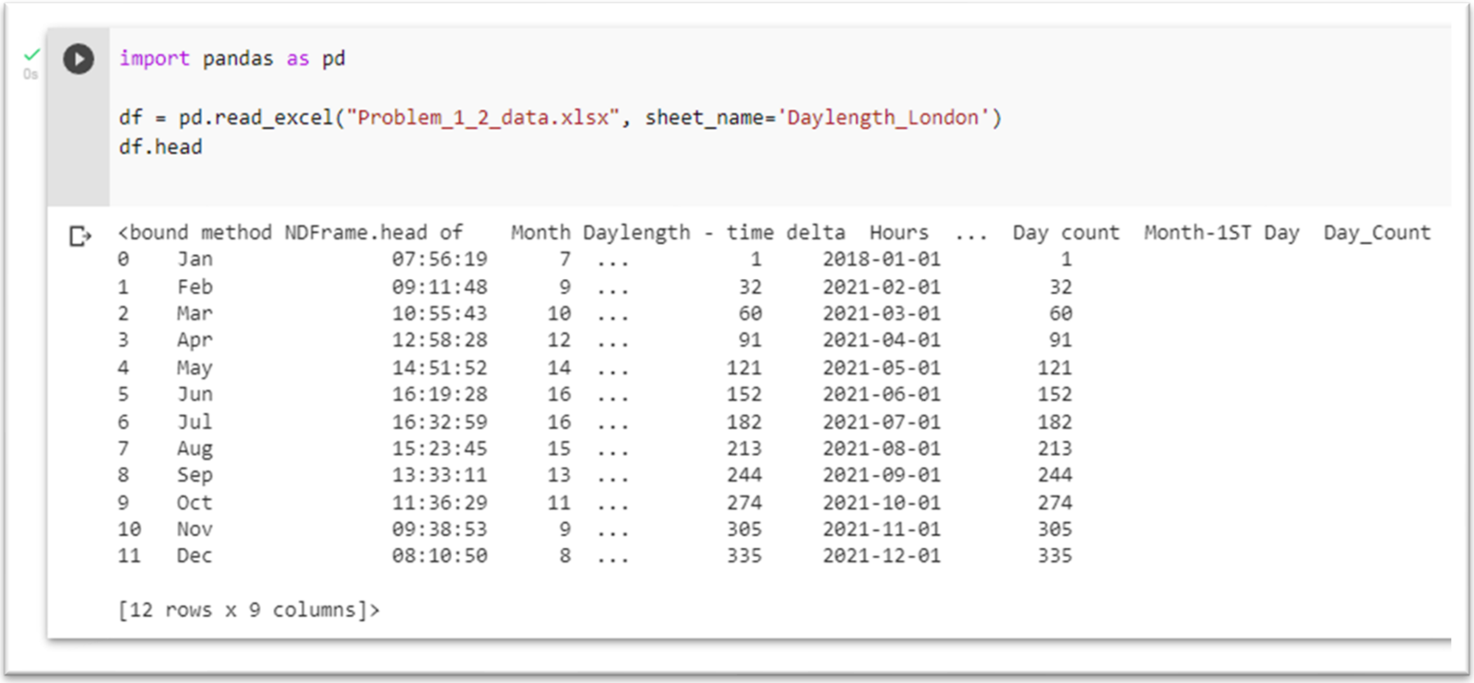
- Current example has only 12 data points, so can be typed up
- Real life examples have hundreds of thousands of records
- Pandas can be used to load data from excel or csv
- Pandas also has got a suite of functionalities for data processing

Uploading local file to colab –



```
[17] from google.colab import files
     import io
     uploaded = files.upload()
```

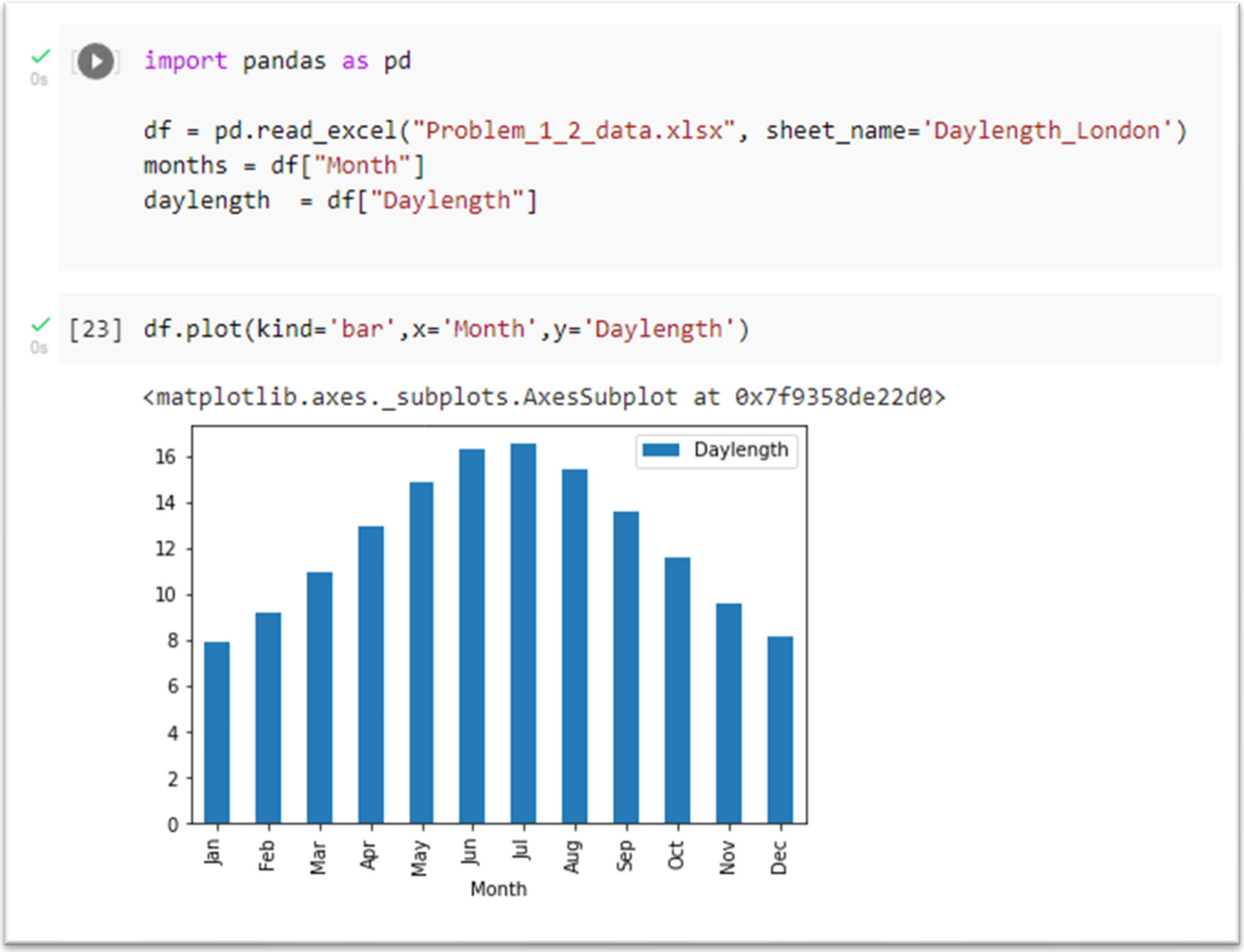
Import pandas and read data -



```
import pandas as pd

df = pd.read_excel("Problem_1_2_data.xlsx", sheet_name='Daylength_London')
df.head
```

```
<bound method NDFrame.head of   Month Daylength - time delta  Hours  ...  Day count  Month-1ST Day  Day_Count
0    Jan               07:56:19      7  ...          1     2018-01-01          1
1    Feb               09:11:48      9  ...         32     2021-02-01         32
2    Mar               10:55:43     10  ...         60     2021-03-01         60
3    Apr               12:58:28     12  ...         91     2021-04-01         91
4    May               14:51:52     14  ...        121     2021-05-01        121
5    Jun               16:19:28     16  ...        152     2021-06-01        152
6    Jul               16:32:59     16  ...        182     2021-07-01        182
7    Aug               15:23:45     15  ...        213     2021-08-01        213
8    Sep               13:33:11     13  ...        244     2021-09-01        244
9    Oct               11:36:29     11  ...        274     2021-10-01        274
10   Nov               09:38:53      9  ...        305     2021-11-01        305
11   Dec               08:10:50      8  ...        335     2021-12-01        335

[12 rows x 9 columns]>
```

Plotting using pandas –

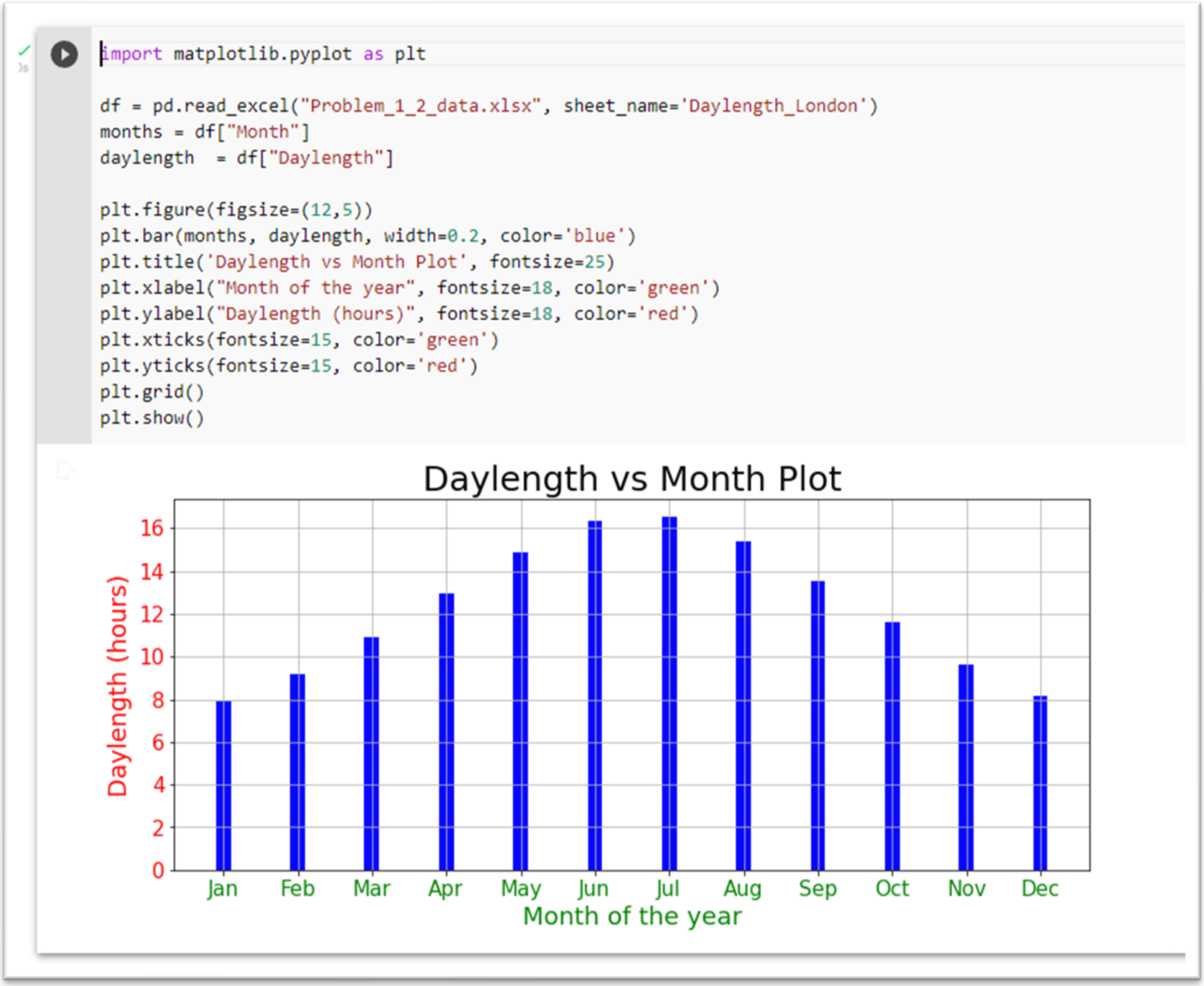


```
import pandas as pd

df = pd.read_excel("Problem_1_2_data.xlsx", sheet_name='Daylength_London')
months = df["Month"]
daylength  = df["Daylength"]
```

```
[23] df.plot(kind='bar',x='Month',y='Daylength')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f9358de22d0>
```

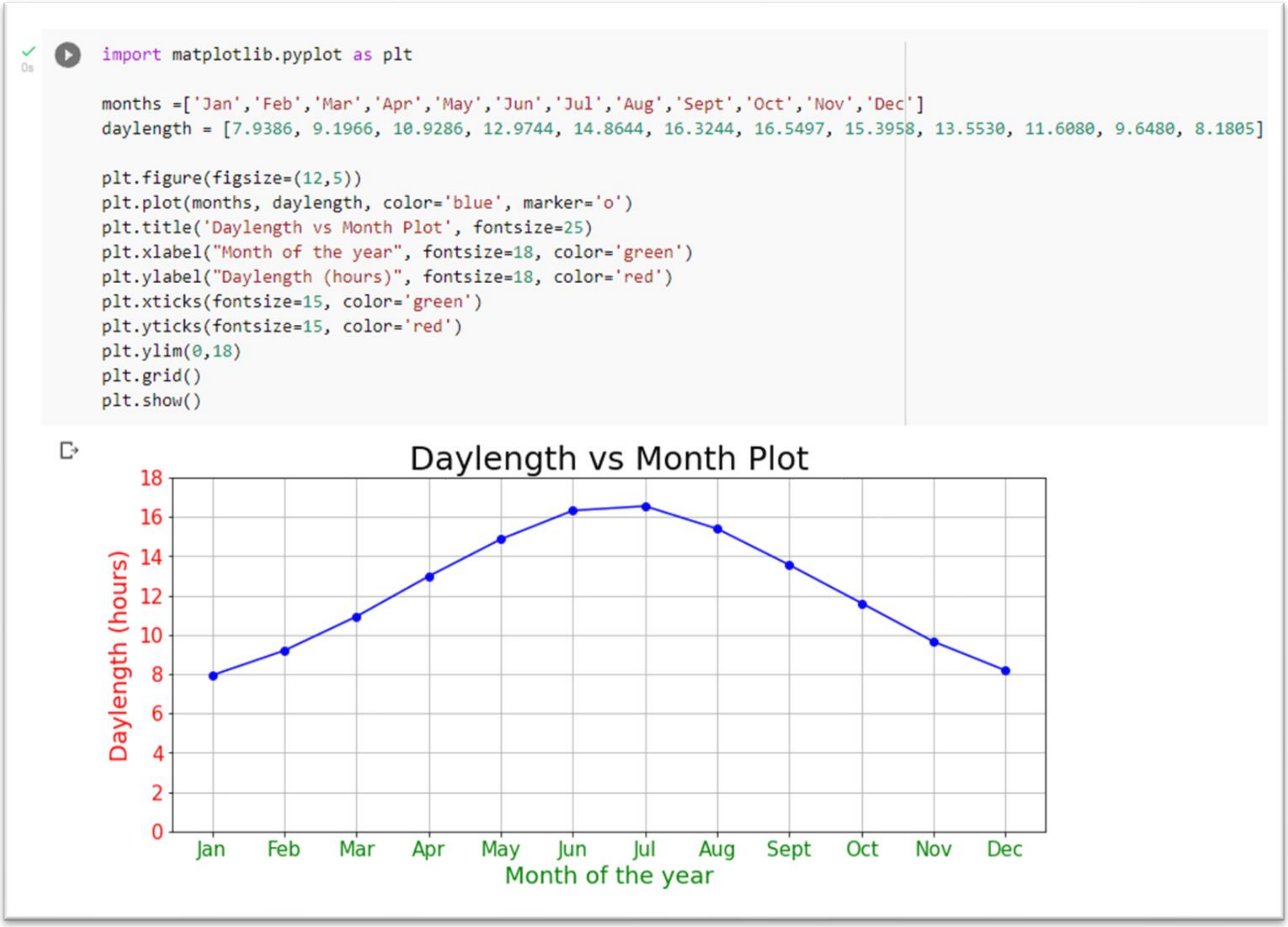
Plotting using Matplotlib –



```
import matplotlib.pyplot as plt

df = pd.read_excel("Problem_1_2_data.xlsx", sheet_name='Daylength_London')
months = df["Month"]
daylength  = df["Daylength"]

plt.figure(figsize=(12,5))
plt.bar(months, daylength, width=0.2, color='blue')
plt.title('Daylength vs Month Plot', fontsize=25)
plt.xlabel("Month of the year", fontsize=18, color='green')
plt.ylabel("Daylength (hours)", fontsize=18, color='red')
plt.xticks(fontsize=15, color='green')
plt.yticks(fontsize=15, color='red')
plt.grid()
plt.show()
```

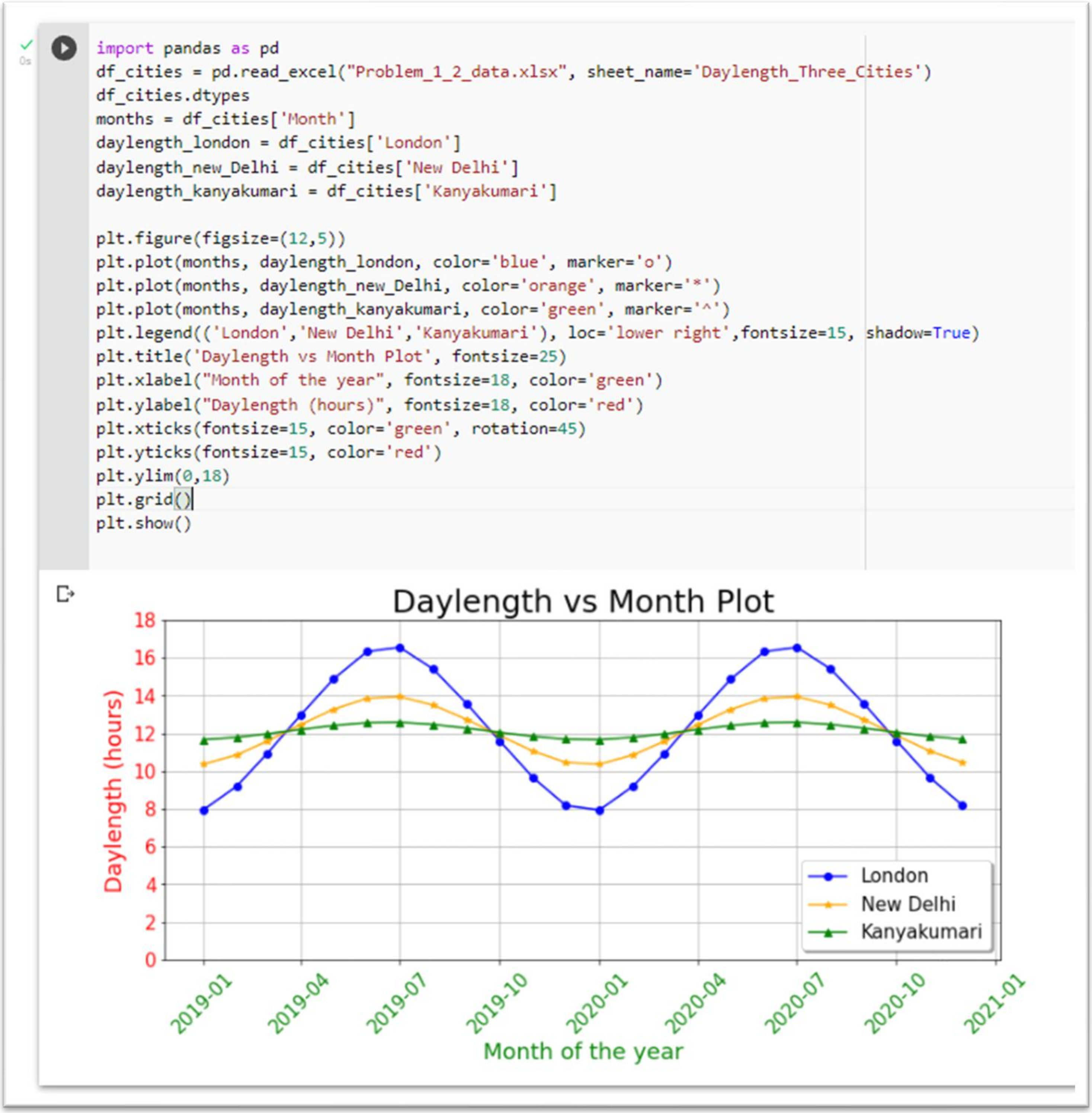
7.4 Introduction to NumPy, Pandas and Matplotlib

Plotting Line Charts



```
import matplotlib.pyplot as plt

months =['Jan','Feb','Mar','Apr','May','Jun','Jul','Aug','Sept','Oct','Nov','Dec']
daylength = [7.9386, 9.1966, 10.9286, 12.9744, 14.8644, 16.3244, 16.5497, 15.3958, 13.5530, 11.6080, 9.6480, 8.1805]

plt.figure(figsize=(12,5))
plt.plot(months, daylength, color='blue', marker='o')
plt.title('Daylength vs Month Plot', fontsize=25)
plt.xlabel("Month of the year", fontsize=18, color='green')
plt.ylabel("Daylength (hours)", fontsize=18, color='red')
plt.xticks(fontsize=15, color='green')
plt.yticks(fontsize=15, color='red')
plt.ylim(0,18)
plt.grid()
plt.show()
```

Plotting multiple line Charts

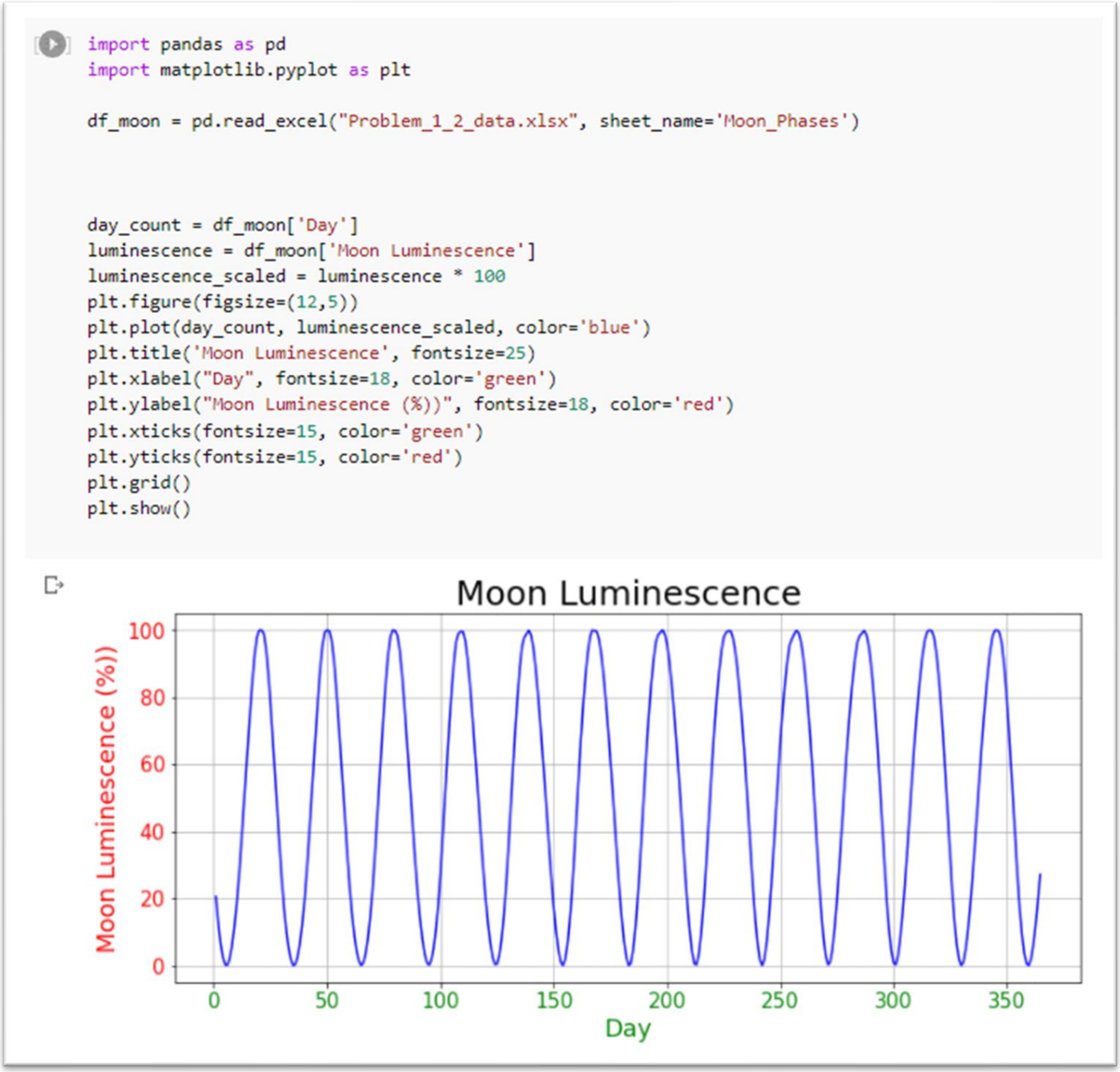


```
import pandas as pd
df_cities = pd.read_excel("Problem_1_2_data.xlsx", sheet_name='Daylength_Three_Cities')
df_cities.dtypes
months = df_cities['Month']
daylength_london = df_cities['London']
daylength_new_Delhi = df_cities['New Delhi']
daylength_kanyakumari = df_cities['Kanyakumari']

plt.figure(figsize=(12,5))
plt.plot(months, daylength_london, color='blue', marker='o')
plt.plot(months, daylength_new_Delhi, color='orange', marker='*')
plt.plot(months, daylength_kanyakumari, color='green', marker='^')
plt.legend(('London','New Delhi','Kanyakumari'), loc='lower right',fontsize=15, shadow=True)
plt.title('Daylength vs Month Plot', fontsize=25)
plt.xlabel("Month of the year", fontsize=18, color='green')
plt.ylabel("Daylength (hours)", fontsize=18, color='red')
plt.xticks(fontsize=15, color='green', rotation=45)
plt.yticks(fontsize=15, color='red')
plt.ylim(0,18)
plt.grid()
plt.show()
```

7.5 Introduction to Numpy, Pandas and Matplotlib

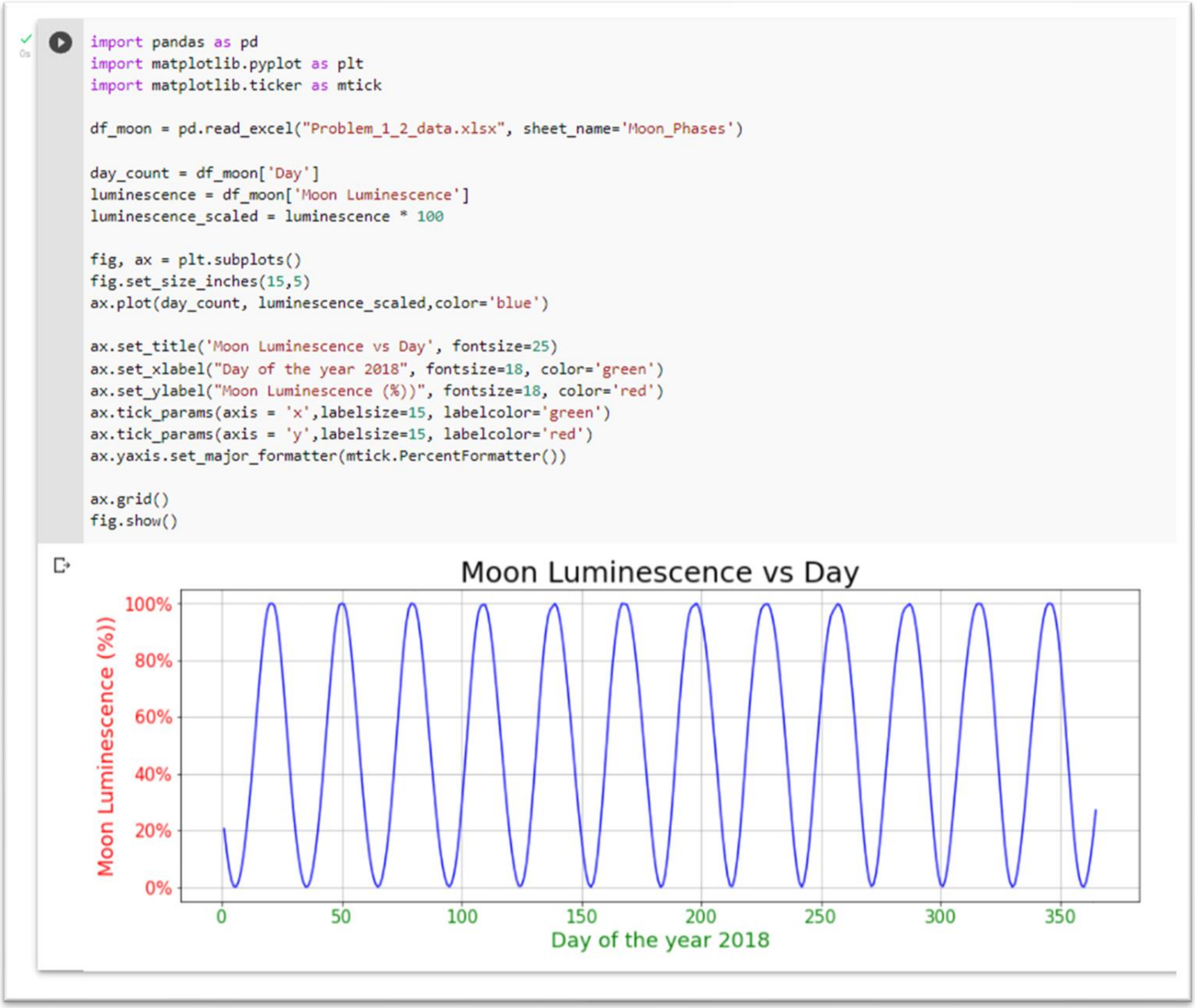
Problem 2 – phases of the moon

- **Classification of variables of interest**
- **Loading excel file to Colab workspace**
- **Loading data to pandas data frame**
- **Plotting the moon luminescence**



```
import pandas as pd
import matplotlib.pyplot as plt

df_moon = pd.read_excel("Problem_1_2_data.xlsx", sheet_name='Moon_Phases')



day_count = df_moon['Day']
luminescence = df_moon['Moon Luminescence']
luminescence_scaled = luminescence * 100
plt.figure(figsize=(12,5))
plt.plot(day_count, luminescence_scaled, color='blue')
plt.title('Moon Luminescence', fontsize=25)
plt.xlabel("Day", fontsize=18, color='green')
plt.ylabel("Moon Luminescence (%))", fontsize=18, color='red')
plt.xticks(fontsize=15, color='green')
plt.yticks(fontsize=15, color='red')
plt.grid()
plt.show()
```

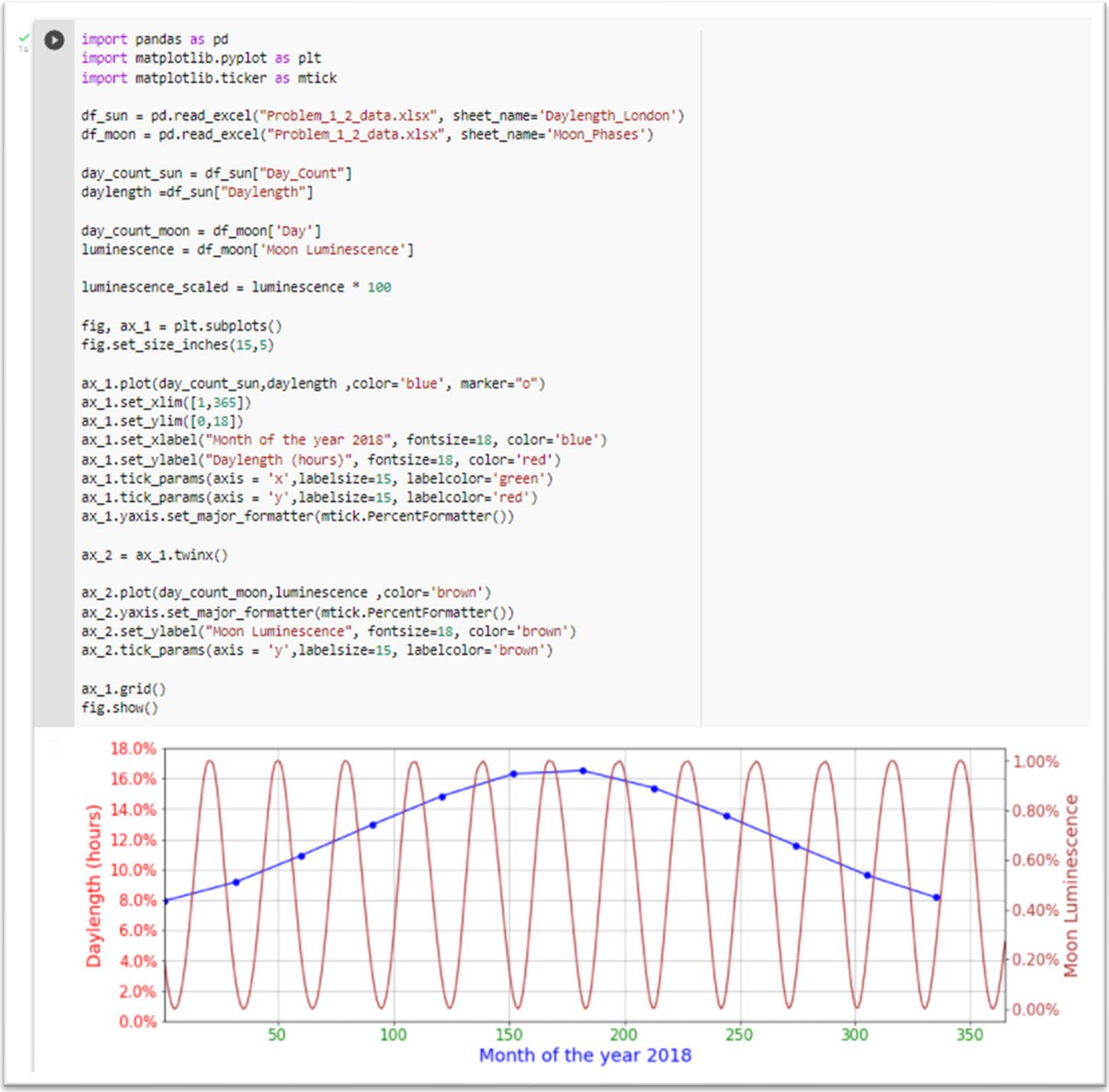
- **Finer control pf plots with fig,ax=subplots()**



```python
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib.ticker as mtick

df_moon = pd.read_excel("Problem_1_2_data.xlsx", sheet_name='Moon_Phases')

day_count = df_moon['Day']
luminescence = df_moon['Moon Luminescence']
luminescence_scaled = luminescence * 100

fig, ax = plt.subplots()
fig.set_size_inches(15,5)
ax.plot(day_count, luminescence_scaled,color='blue')

ax.set_title('Moon Luminescence vs Day', fontsize=25)
ax.set_xlabel("Day of the year 2018", fontsize=18, color='green')
ax.set_ylabel("Moon Luminescence (%))", fontsize=18, color='red')
ax.tick_params(axis = 'x',labelsize=15, labelcolor='green')
ax.tick_params(axis = 'y',labelsize=15, labelcolor='red')
ax.yaxis.set_major_formatter(mtick.PercentFormatter())

ax.grid()
fig.show()
```

7.6 Introduction to Numpy, Pandas and Matplotlib

Combining two problems –

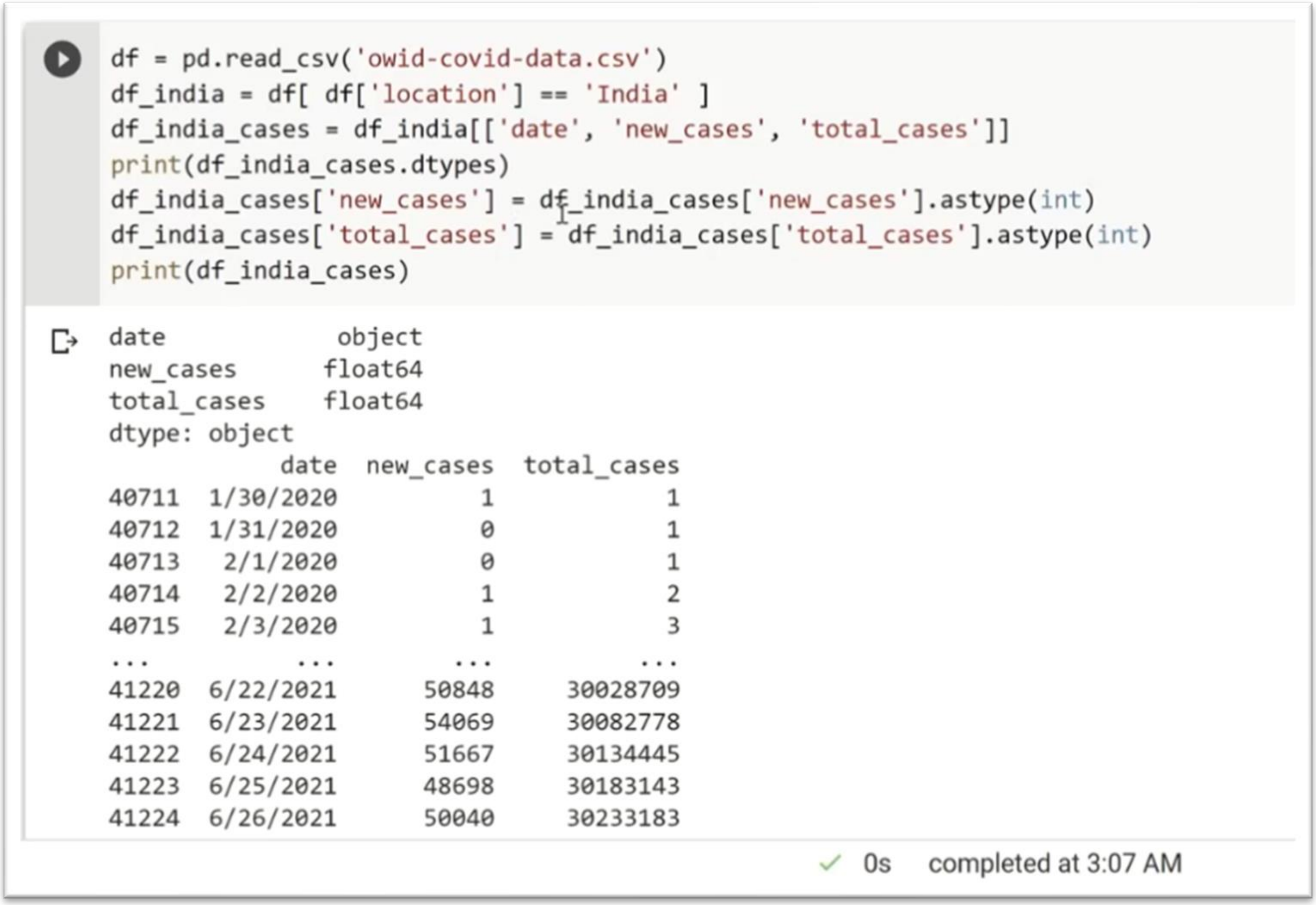


```
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib.ticker as mtick

df_sun = pd.read_excel("Problem_1_2_data.xlsx", sheet_name='Daylength_London')
df_moon = pd.read_excel("Problem_1_2_data.xlsx", sheet_name='Moon_Phases')

day_count_sun = df_sun["Day_Count"]
daylength =df_sun["Daylength"]

day_count_moon = df_moon['Day']
luminescence = df_moon['Moon Luminescence']

luminescence_scaled = luminescence * 100

fig, ax_1 = plt.subplots()
fig.set_size_inches(15,5)

ax_1.plot(day_count_sun,daylength ,color='blue', marker="o")
ax_1.set_xlim([1,365])
ax_1.set_ylim([0,18])
ax_1.set_xlabel("Month of the year 2018", fontsize=18, color='blue')
ax_1.set_ylabel("Daylength (hours)", fontsize=18, color='red')
ax_1.tick_params(axis = 'x',labelsize=15, labelcolor='green')
ax_1.tick_params(axis = 'y',labelsize=15, labelcolor='red')
ax_1.yaxis.set_major_formatter(mtick.PercentFormatter())

ax_2 = ax_1.twinx()

ax_2.plot(day_count_moon,luminescence ,color='brown')
ax_2.yaxis.set_major_formatter(mtick.PercentFormatter())
ax_2.set_ylabel("Moon Luminescence", fontsize=18, color='brown')
ax_2.tick_params(axis = 'y',labelsize=15, labelcolor='brown')

ax_1.grid()
fig.show()
```

7.7 Introduction to Numpy, Pandas and Matplotlib

Problem 3 – Covid data

Creating consecutive integers with NumPy



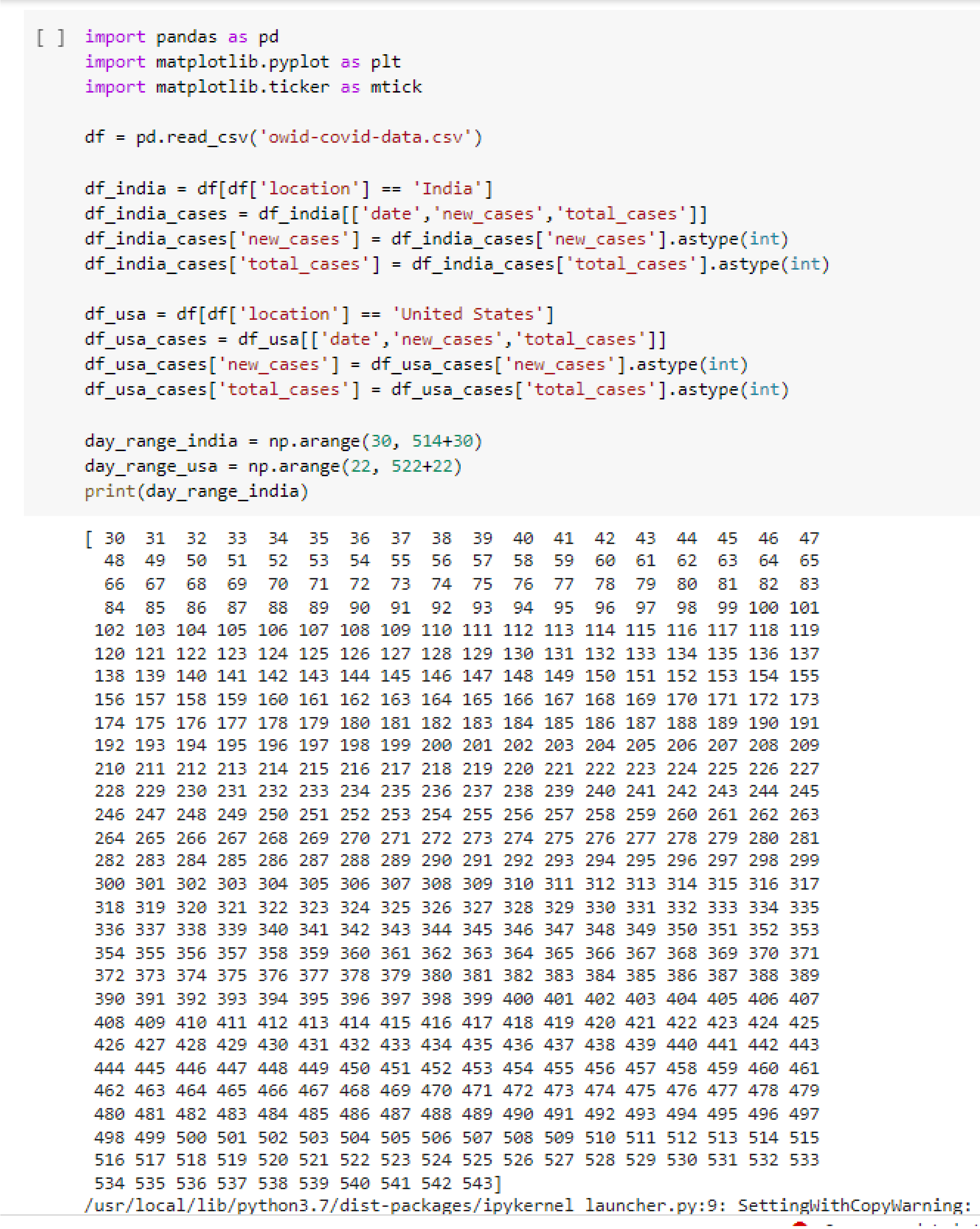
```
df = pd.read_csv('owid-covid-data.csv')
df_india = df[ df['location'] == 'India' ]
df_india_cases = df_india[['date', 'new_cases', 'total_cases']]
print(df_india_cases.dtypes)
df_india_cases['new_cases'] = df_india_cases['new_cases'].astype(int)
df_india_cases['total_cases'] = df_india_cases['total_cases'].astype(int)
print(df_india_cases)
```

```
date             object
new_cases       float64
total_cases     float64
dtype: object
            date  new_cases  total_cases
40711  1/30/2020          1            1
40712  1/31/2020          0            1
40713   2/1/2020          0            1
40714   2/2/2020          1            2
40715   2/3/2020          1            3
...          ...        ...          ...
41220  6/22/2021      50848     30028709
41221  6/23/2021      54069     30082778
41222  6/24/2021      51667     30134445
41223  6/25/2021      48698     30183143
41224  6/26/2021      50040     30233183
```

7.8 Introduction to Numpy, Pandas and Matplotlib

Covid data – Classification of variables

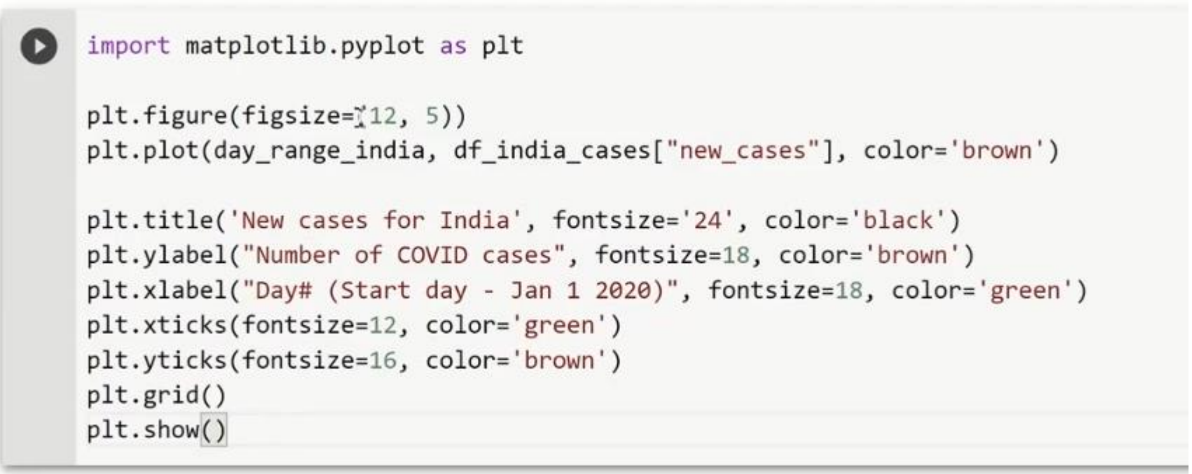
Generating the x-axis range



```
[ ]  import pandas as pd
     import matplotlib.pyplot as plt
     import matplotlib.ticker as mtick

     df = pd.read_csv('owid-covid-data.csv')

     df_india = df[df['location'] == 'India']
     df_india_cases = df_india[['date','new_cases','total_cases']]
     df_india_cases['new_cases'] = df_india_cases['new_cases'].astype(int)
     df_india_cases['total_cases'] = df_india_cases['total_cases'].astype(int)

     df_usa = df[df['location'] == 'United States']
     df_usa_cases = df_usa[['date','new_cases','total_cases']]
     df_usa_cases['new_cases'] = df_usa_cases['new_cases'].astype(int)
     df_usa_cases['total_cases'] = df_usa_cases['total_cases'].astype(int)

     day_range_india = np.arange(30, 514+30)
     day_range_usa = np.arange(22, 522+22)
     print(day_range_india)
```

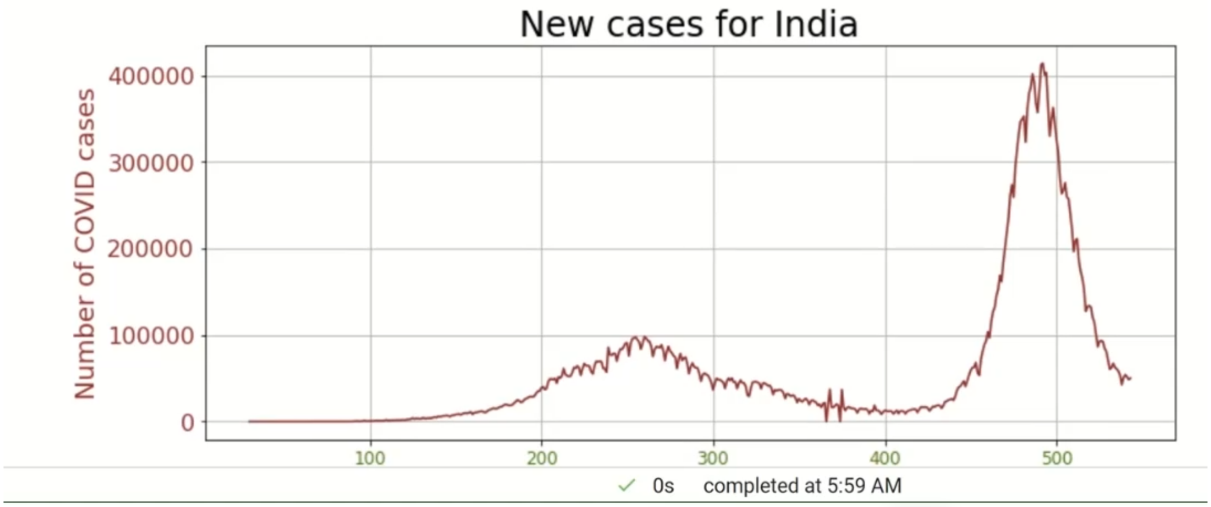
```
[ 30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47
  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65
  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83
  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99 100 101
 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119
 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137
 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155
 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173
 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191
 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209
 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227
 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245
 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260 261 262 263
 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281
 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299
 300 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317
 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335
 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351 352 353
 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371
 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389
 390 391 392 393 394 395 396 397 398 399 400 401 402 403 404 405 406 407
 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425
 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443
 444 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460 461
 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479
 480 481 482 483 484 485 486 487 488 489 490 491 492 493 494 495 496 497
 498 499 500 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515
 516 517 518 519 520 521 522 523 524 525 526 527 528 529 530 531 532 533
 534 535 536 537 538 539 540 541 542 543]
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:9: SettingWithCopyWarning:
```

Plotting the new cases for both countries –



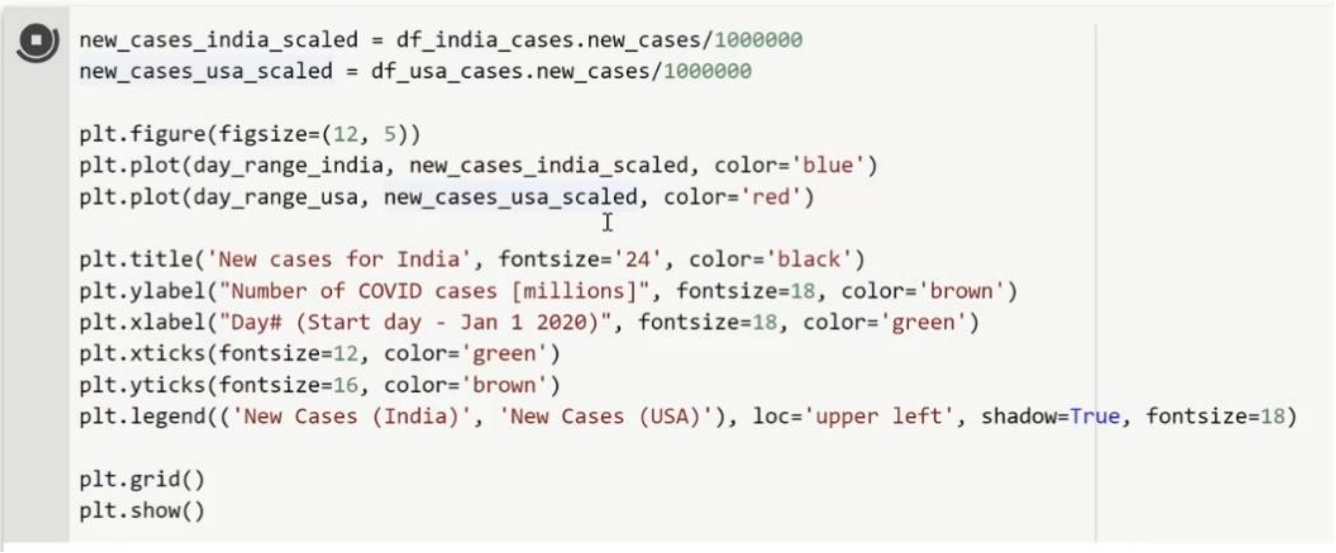
```
import matplotlib.pyplot as plt

plt.figure(figsize=(12, 5))
plt.plot(day_range_india, df_india_cases["new_cases"], color='brown')

plt.title('New cases for India', fontsize='24', color='black')
plt.ylabel("Number of COVID cases", fontsize=18, color='brown')
plt.xlabel("Day# (Start day - Jan 1 2020)", fontsize=18, color='green')
plt.xticks(fontsize=12, color='green')
plt.yticks(fontsize=16, color='brown')
plt.grid()
plt.show()
```

Result –



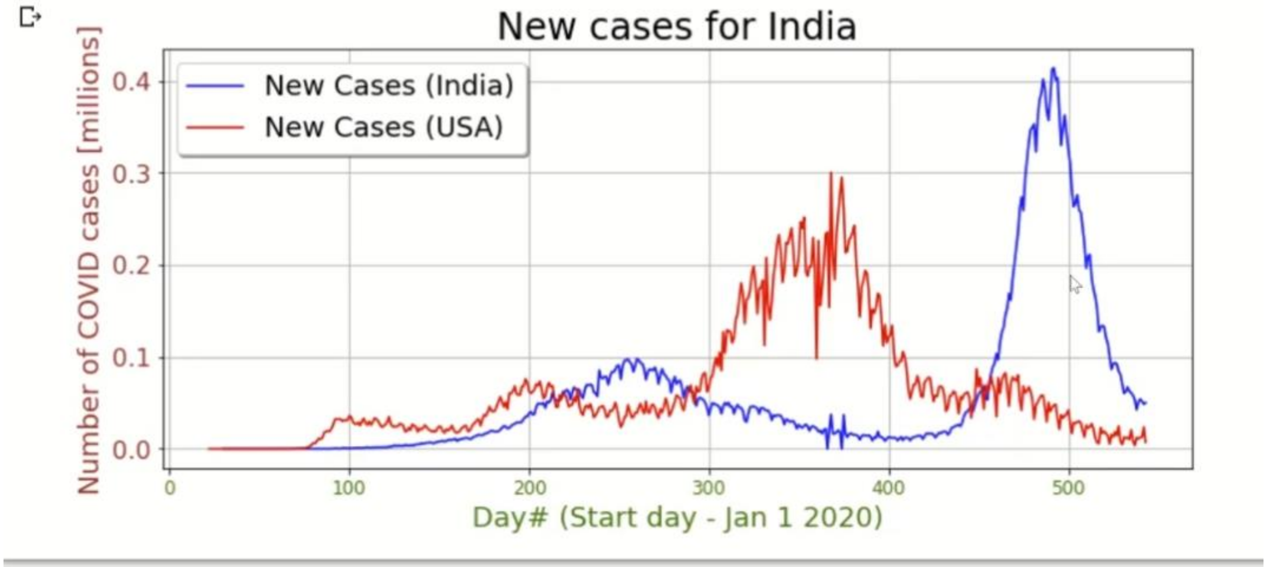
✓ 0s completed at 5:59 AM

Combined -

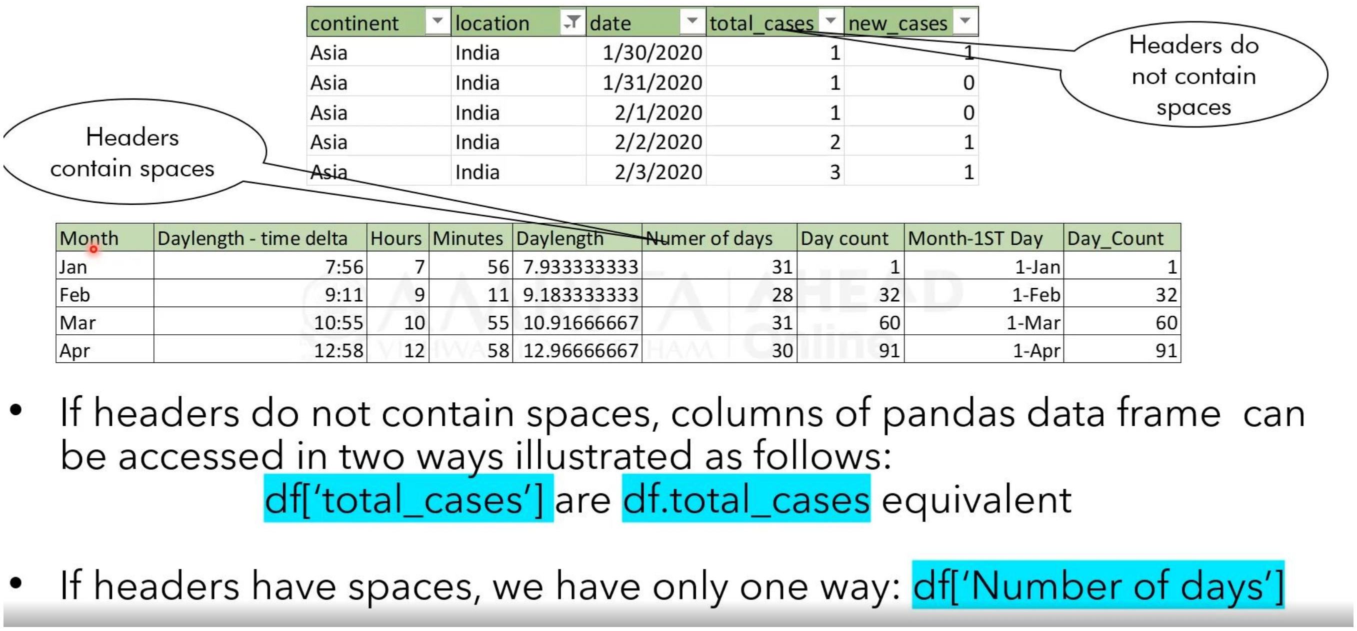


```
new_cases_india_scaled = df_india_cases.new_cases/1000000
new_cases_usa_scaled = df_usa_cases.new_cases/1000000

plt.figure(figsize=(12, 5))
plt.plot(day_range_india, new_cases_india_scaled, color='blue')
plt.plot(day_range_usa, new_cases_usa_scaled, color='red')

plt.title('New cases for India', fontsize='24', color='black')
plt.ylabel("Number of COVID cases [millions]", fontsize=18, color='brown')
plt.xlabel("Day# (Start day - Jan 1 2020)", fontsize=18, color='green')
plt.xticks(fontsize=12, color='green')
plt.yticks(fontsize=16, color='brown')
plt.legend(('New Cases (India)', 'New Cases (USA)'), loc='upper left', shadow=True, fontsize=18)

plt.grid()
plt.show()
```

Result –



Two ways of accessing columns of pandas dataframe



continent	location	date	total_cases	new_cases
Asia	India	1/30/2020	1	1
Asia	India	1/31/2020	1	0
Asia	India	2/1/2020	1	0
Asia	India	2/2/2020	2	1
Asia	India	2/3/2020	3	1

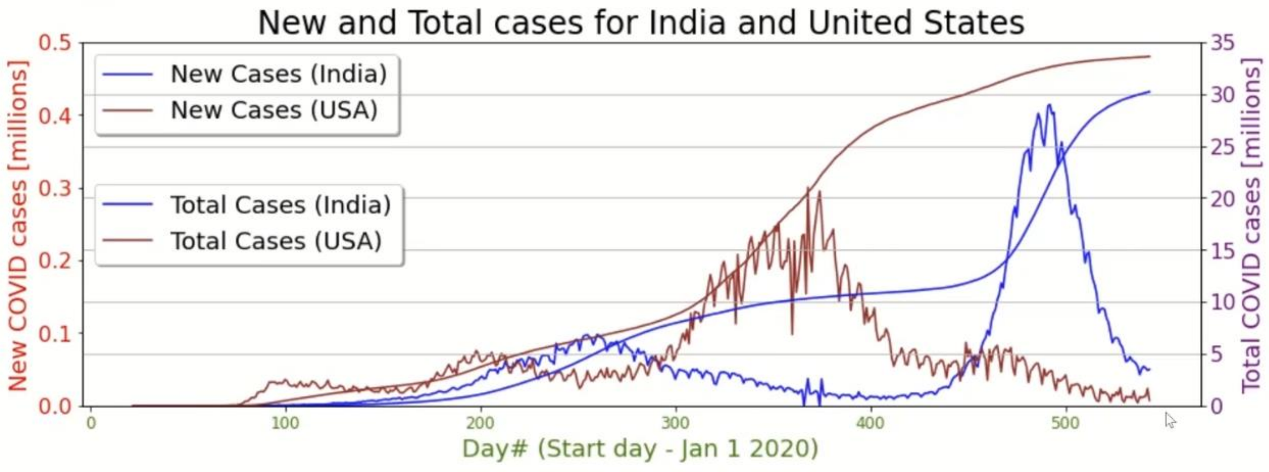
Headers do not contain spaces

Headers contain spaces

Month	Daylength - time delta	Hours	Minutes	Daylength	Numer of days	Day count	Month-1ST Day	Day_Count
Jan	7:56	7	56	7.933333333	31	1	1-Jan	1
Feb	9:11	9	11	9.183333333	28	32	1-Feb	32
Mar	10:55	10	55	10.91666667	31	60	1-Mar	60
Apr	12:58	12	58	12.96666667	30	91	1-Apr	91

- If headers do not contain spaces, columns of pandas data frame can be accessed in two ways illustrated as follows:
  df['total_cases'] are df.total_cases equivalent
- If headers have spaces, we have only one way: df['Number of days']

All curves combined –


New and Total cases for India and United States
New Cases (India)
New Cases (USA)
Total Cases (India)
Total Cases (USA)
New COVID cases [millions]
Total COVID cases [millions]
Day# (Start day - Jan 1 2020)