# Pandas real life solution to problems

## importing pandas, numpy and matplotlib.pyplot

In [2]:

```
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
```

## Initial cleaning of data

In [3]:

```
#getting the csv file from the folder path as specified below
yearly_sales=pd.read_csv("D:\\python jupyter\data files\yearly_sales.csv")
```

In [5]:

```
#for dropping nan valued rows

yearly_sales=yearly_sales.dropna(axis=0,how="all")
```

In [6]:

```
#for remonving rows with "Or" in Order Date

yearly_sales=yearly_sales.loc[~(yearly_sales["Order Date"].str.contains("Or"))]
```

In [7]:

```
#keith's method

yearly_sales["month"]=yearly_sales["Order Date"].str[0:2]
yearly_sales["month"]=yearly_sales["month"].astype("int32")
yearly_sales.head()
```

Out[7]:

| | Order ID | Product | Quantity Ordered | Price Each | Order Date | Purchase Address | month |
|---|---|---|---|---|---|---|---|
| **0** | 176558 | USB-C Charging Cable | 2 | 11.95 | 04/19/19 08:46 | 917 1st St, Dallas, TX 75001 | 4 |
| **2** | 176559 | Bose SoundSport Headphones | 1 | 99.99 | 04/07/19 22:30 | 682 Chestnut St, Boston, MA 02215 | 4 |
| **3** | 176560 | Google Phone | 1 | 600 | 04/12/19 14:38 | 669 Spruce St, Los Angeles, CA 90001 | 4 |
| **4** | 176560 | Wired Headphones | 1 | 11.99 | 04/12/19 14:38 | 669 Spruce St, Los Angeles, CA 90001 | 4 |
| **5** | 176561 | Wired Headphones | 1 | 11.99 | 04/30/19 09:27 | 333 8th St, Los Angeles, CA 90001 | 4 |

## Changing the Quantity Ordered and the Price Each to the required type i.e int,float resp.

In [8]:

```
yearly_sales["Quantity Ordered"]=yearly_sales["Quantity Ordered"].astype("int32")
yearly_sales["Price Each"]=yearly_sales["Price Each"].astype("float64")
yearly_sales["sales"]=yearly_sales["Quantity Ordered"]*yearly_sales["Price Each"]
ysales=yearly_sales.groupby(["month"]).sum()["sales"]
ysales
```

Out[8]:

```
month
1     1.822257e+06
2     2.202022e+06
3     2.807100e+06
4     3.390670e+06
5     3.152607e+06
6     2.577802e+06
7     2.647776e+06
8     2.244468e+06
9     2.097560e+06
10    3.736727e+06
11    3.199603e+06
12    4.613443e+06
Name: sales, dtype: float64
```

## Using apply function to get each cell values and manipulate it further

In [9]:

```
def get_state(name):
     return name.split(",")[1]
def get_city(name):
    return name.split(",")[2].split(" ")[1]
yearly_sales["City"]=yearly_sales["Purchase Address"].apply(lambda x:f"{get_state(x)}({get_city(x)})")
yearly_sales.tail()
```

Out[9]:

| | Order ID | Product | Quantity Ordered | Price Each | Order Date | Purchase Address | month | sales | City |
|---|---|---|---|---|---|---|---|---|---|
| **186845** | 259353 | AAA Batteries (4-pack) | 3 | 2.99 | 09/17/19 20:56 | 840 Highland St, Los Angeles, CA 90001 | 9 | 8.97 | Los Angeles(CA) |
| **186846** | 259354 | iPhone | 1 | 700.00 | 09/01/19 16:00 | 216 Dogwood St, San Francisco, CA 94016 | 9 | 700.00 | San Francisco(CA) |
| **186847** | 259355 | iPhone | 1 | 700.00 | 09/23/19 07:39 | 220 12th St, San Francisco, CA 94016 | 9 | 700.00 | San Francisco(CA) |
| **186848** | 259356 | 34in Ultrawide Monitor | 1 | 379.99 | 09/19/19 17:30 | 511 Forest St, San Francisco, CA 94016 | 9 | 379.99 | San Francisco(CA) |
| **186849** | 259357 | USB-C Charging Cable | 1 | 11.95 | 09/30/19 00:18 | 250 Meadow St, San Francisco, CA 94016 | 9 | 11.95 | San Francisco(CA) |

In [10]:

```
csales=yearly_sales.groupby(["City"]).sum()["sales"]
csales
```

Out[10]:

```
City
 Atlanta(GA)          2.795499e+06
 Austin(TX)           1.819582e+06
 Boston(MA)           3.661642e+06
 Dallas(TX)           2.767975e+06
 Los Angeles(CA)      5.452571e+06
 New York City(NY)    4.664317e+06
 Portland(ME)         4.497583e+05
 Portland(OR)         1.870732e+06
 San Francisco(CA)    8.262204e+06
 Seattle(WA)          2.747755e+06
Name: sales, dtype: float64
```

## Changing the order time in datetime object to extract the date and time in better way

In [11]:

```
yearly_sales["Order Date"]=pd.to_datetime(yearly_sales["Order Date"])
```

In [141]:

```
yearly_sales["Hour"]=yearly_sales["Order Date"].dt.hour
yearly_sales["minute"]=yearly_sales["Order Date"].dt.minute
myfavcity=yearly_sales.loc[yearly_sales["City"]==" San Francisco(CA)"]
hsales=myfavcity.groupby(["City","Hour"]).count()
hsales
```

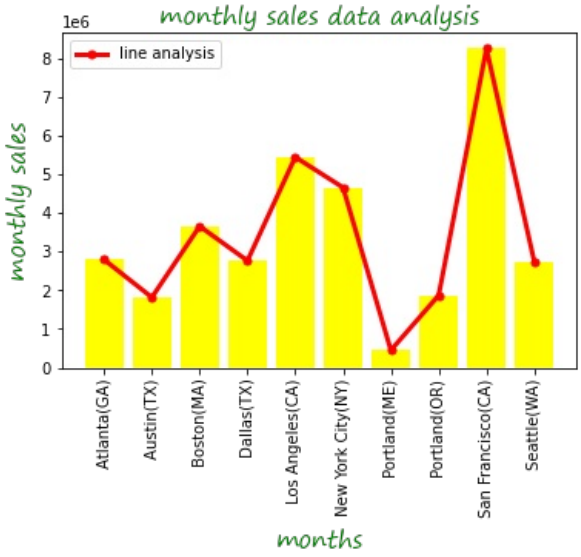
Out[141]:

| | | Order ID | Product | Quantity Ordered | Price Each | Order Date | Purchase Address | month | sales | minute |
|---|---|---|---|---|---|---|---|---|---|---|
| **City** | **Hour** | | | | | | | | | |
| **San Francisco(CA)** | **0** | 940 | 940 | 940 | 940 | 940 | 940 | 940 | 940 | 940 |
| | **1** | 570 | 570 | 570 | 570 | 570 | 570 | 570 | 570 | 570 |
| | **2** | 315 | 315 | 315 | 315 | 315 | 315 | 315 | 315 | 315 |
| | **3** | 220 | 220 | 220 | 220 | 220 | 220 | 220 | 220 | 220 |
| | **4** | 217 | 217 | 217 | 217 | 217 | 217 | 217 | 217 | 217 |
| | **5** | 344 | 344 | 344 | 344 | 344 | 344 | 344 | 344 | 344 |
| | **6** | 630 | 630 | 630 | 630 | 630 | 630 | 630 | 630 | 630 |
| | **7** | 989 | 989 | 989 | 989 | 989 | 989 | 989 | 989 | 989 |
| | **8** | 1471 | 1471 | 1471 | 1471 | 1471 | 1471 | 1471 | 1471 | 1471 |
| | **9** | 2064 | 2064 | 2064 | 2064 | 2064 | 2064 | 2064 | 2064 | 2064 |
| | **10** | 2695 | 2695 | 2695 | 2695 | 2695 | 2695 | 2695 | 2695 | 2695 |
| | **11** | 2964 | 2964 | 2964 | 2964 | 2964 | 2964 | 2964 | 2964 | 2964 |
| | **12** | 2998 | 2998 | 2998 | 2998 | 2998 | 2998 | 2998 | 2998 | 2998 |
| | **13** | 2865 | 2865 | 2865 | 2865 | 2865 | 2865 | 2865 | 2865 | 2865 |
| | **14** | 2642 | 2642 | 2642 | 2642 | 2642 | 2642 | 2642 | 2642 | 2642 |
| | **15** | 2486 | 2486 | 2486 | 2486 | 2486 | 2486 | 2486 | 2486 | 2486 |
| | **16** | 2484 | 2484 | 2484 | 2484 | 2484 | 2484 | 2484 | 2484 | 2484 |
| | **17** | 2634 | 2634 | 2634 | 2634 | 2634 | 2634 | 2634 | 2634 | 2634 |
| | **18** | 2988 | 2988 | 2988 | 2988 | 2988 | 2988 | 2988 | 2988 | 2988 |
| | **19** | 3106 | 3106 | 3106 | 3106 | 3106 | 3106 | 3106 | 3106 | 3106 |
| | **20** | 2945 | 2945 | 2945 | 2945 | 2945 | 2945 | 2945 | 2945 | 2945 |
| | **21** | 2595 | 2595 | 2595 | 2595 | 2595 | 2595 | 2595 | 2595 | 2595 |
| | **22** | 2108 | 2108 | 2108 | 2108 | 2108 | 2108 | 2108 | 2108 | 2108 |
| | **23** | 1462 | 1462 | 1462 | 1462 | 1462 | 1462 | 1462 | 1462 | 1462 |

## Monthly sales of cities data analysis

In [13]:

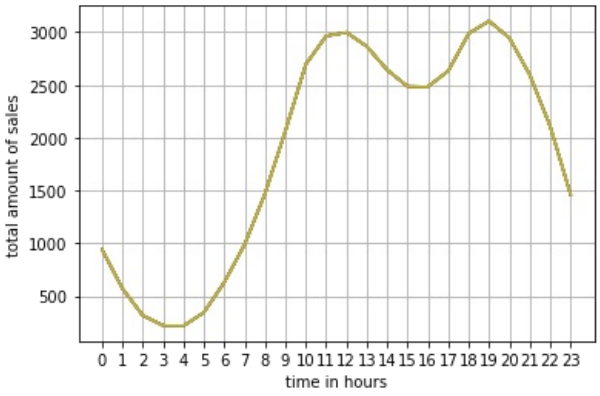
```
months=[city for city,df in yearly_sales.groupby("City")]
plt.bar(months,csales,color="yellow")
plt.xticks(months,rotation="vertical")
plt.xlabel("months",color="green",fontname="segoe print",fontsize=15)
plt.ylabel("monthly sales",color="green",fontname="segoe print",fontsize=15)
plt.title("monthly sales data analysis",color="green",fontname="segoe print",fontsize=15)
plt.plot(months,csales,marker=".",color="red",markersize=10,linewidth=3,label="line analysis")
plt.legend()
plt.show()
```




### Yearly sales of Los Angeles(CA)

In [143]:

```
hours=[hour for hour,df in yearly_sales.groupby("Hour")]
plt.plot(hours,hsales)
plt.xticks(hours)
plt.xlabel("time in hours")
plt.ylabel("total amount of sales")
plt.grid()
plt.show()
```




### Transforming the products into one row which have the same order ID

In [15]:

```
df=yearly_sales.loc[yearly_sales["Order ID"].duplicated(keep=False)]

df["Groups"]=df.groupby(["Order ID"])["Product"].transform(lambda x:",".join(x))
df=df[["Order ID","Groups"]].drop_duplicates()
df.head(10)
```

```
<ipython-input-15-6e609fc564b3>:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexi
ng.html#returning-a-view-versus-a-copy
  df["Groups"]=df.groupby(["Order ID"])["Product"].transform(lambda x:",".join(x))
```

Out[15]:

| | Order ID | Groups |
|---|---|---|
| 3 | 176560 | Google Phone,Wired Headphones |
| 18 | 176574 | Google Phone,USB-C Charging Cable |
| 30 | 176585 | Bose SoundSport Headphones,Bose SoundSport Hea... |
| 32 | 176586 | AAA Batteries (4-pack),Google Phone |
| 119 | 176672 | Lightning Charging Cable,USB-C Charging Cable |
| 129 | 176681 | Apple Airpods Headphones,ThinkPad Laptop |
| 138 | 176689 | Bose SoundSport Headphones,AAA Batteries (4-pack) |
| 189 | 176739 | 34in Ultrawide Monitor,Google Phone |
| 225 | 176774 | Lightning Charging Cable,USB-C Charging Cable |
| 233 | 176781 | iPhone,Lightning Charging Cable |

## Using combinations and Counter methods to get the maximum order product in a group

In [96]:

```
from itertools import combinations
from collections import Counter
count=Counter()
for rows in df["Groups"]:
    row_list=rows.split(",")
    count.update(Counter(combinations(row_list,3)))
for x,y in count.most_common():
    print(x,y)
```

```
('Google Phone', 'USB-C Charging Cable', 'Wired Headphones') 87
('iPhone', 'Lightning Charging Cable', 'Wired Headphones') 62
('iPhone', 'Lightning Charging Cable', 'Apple Airpods Headphones') 47
('Google Phone', 'USB-C Charging Cable', 'Bose SoundSport Headphones') 35
('Vareebadd Phone', 'USB-C Charging Cable', 'Wired Headphones') 33
('iPhone', 'Apple Airpods Headphones', 'Wired Headphones') 27
('Google Phone', 'Bose SoundSport Headphones', 'Wired Headphones') 24
('Vareebadd Phone', 'USB-C Charging Cable', 'Bose SoundSport Headphones') 16
('USB-C Charging Cable', 'Bose SoundSport Headphones', 'Wired Headphones') 5
('Vareebadd Phone', 'Bose SoundSport Headphones', 'Wired Headphones') 5
('Lightning Charging Cable', 'Apple Airpods Headphones', 'Wired Headphones') 4
('iPhone', 'Apple Airpods Headphones', 'AAA Batteries (4-pack)') 3
('iPhone', 'Lightning Charging Cable', 'Lightning Charging Cable') 3
('iPhone', 'Lightning Charging Cable', 'AA Batteries (4-pack)') 3
('Google Phone', 'USB-C Charging Cable', 'AAA Batteries (4-pack)') 3
('Google Phone', 'USB-C Charging Cable', 'AA Batteries (4-pack)') 3
('iPhone', 'Lightning Charging Cable', 'USB-C Charging Cable') 3
('Google Phone', 'USB-C Charging Cable', 'USB-C Charging Cable') 2
('Google Phone', 'Wired Headphones', 'USB-C Charging Cable') 2
('Google Phone', 'USB-C Charging Cable', '27in FHD Monitor') 2
('Google Phone', 'USB-C Charging Cable', 'Apple Airpods Headphones') 2
('iPhone', 'Lightning Charging Cable', 'Google Phone') 2
('Google Phone', 'Wired Headphones', 'AA Batteries (4-pack)') 2
('USB-C Charging Cable', 'Wired Headphones', 'USB-C Charging Cable') 1
('iPhone', 'Lightning Charging Cable', 'Vareebadd Phone') 1
('Google Phone', 'USB-C Charging Cable', 'Vareebadd Phone') 1
('iPhone', 'Wired Headphones', 'AA Batteries (4-pack)') 1
('Lightning Charging Cable', 'Wired Headphones', 'AA Batteries (4-pack)') 1
('Vareebadd Phone', 'USB-C Charging Cable', 'iPhone') 1
('Google Phone', 'USB-C Charging Cable', '34in Ultrawide Monitor') 1
('Google Phone', 'Bose SoundSport Headphones', '34in Ultrawide Monitor') 1
('USB-C Charging Cable', 'Bose SoundSport Headphones', '34in Ultrawide Monitor') 1
('Google Phone', 'Wired Headphones', 'Apple Airpods Headphones') 1
('USB-C Charging Cable', 'Wired Headphones', 'Apple Airpods Headphones') 1
('iPhone', 'Lightning Charging Cable', 'iPhone') 1
('Vareebadd Phone', 'Wired Headphones', 'iPhone') 1
('Vareebadd Phone', 'Wired Headphones', 'Apple Airpods Headphones') 1
('iPhone', 'Apple Airpods Headphones', 'Google Phone') 1
('iPhone', 'Wired Headphones', 'Google Phone') 1
('Lightning Charging Cable', 'Apple Airpods Headphones', 'Google Phone') 1
('Lightning Charging Cable', 'Wired Headphones', 'Google Phone') 1
('Apple Airpods Headphones', 'Wired Headphones', 'Google Phone') 1
('Vareebadd Phone', 'Wired Headphones', '27in 4K Gaming Monitor') 1
('Google Phone', 'Wired Headphones', 'Wired Headphones') 1
('USB-C Charging Cable', 'Wired Headphones', 'Wired Headphones') 1
('Google Phone', 'Wired Headphones', 'Macbook Pro Laptop') 1
('Google Phone', 'Wired Headphones', '27in FHD Monitor') 1
('USB-C Charging Cable', 'Wired Headphones', '27in FHD Monitor') 1
('iPhone', 'Lightning Charging Cable', '34in Ultrawide Monitor') 1
('Google Phone', 'Bose SoundSport Headphones', '27in FHD Monitor') 1
('iPhone', 'Lightning Charging Cable', '27in 4K Gaming Monitor') 1
('USB-C Charging Cable', 'Wired Headphones', 'AA Batteries (4-pack)') 1
('iPhone', 'Apple Airpods Headphones', 'Bose SoundSport Headphones') 1
('iPhone', 'Wired Headphones', 'USB-C Charging Cable') 1
('Lightning Charging Cable', 'Wired Headphones', 'USB-C Charging Cable') 1
('iPhone', 'Wired Headphones', 'Lightning Charging Cable') 1
('Google Phone', 'Bose SoundSport Headphones', 'Apple Airpods Headphones') 1
('Vareebadd Phone', 'USB-C Charging Cable', 'Apple Airpods Headphones') 1
('Google Phone', 'USB-C Charging Cable', 'Lightning Charging Cable') 1
('iPhone', 'Lightning Charging Cable', 'AAA Batteries (4-pack)') 1
('Google Phone', 'Bose SoundSport Headphones', 'Lightning Charging Cable') 1
('Vareebadd Phone', 'Bose SoundSport Headphones', 'Flatscreen TV') 1
('iPhone', 'Lightning Charging Cable', 'Flatscreen TV') 1
('Google Phone', 'USB-C Charging Cable', 'iPhone') 1
('Google Phone', 'Wired Headphones', 'iPhone') 1
('USB-C Charging Cable', 'Wired Headphones', 'iPhone') 1
```

### Grouping dataframe by product

In [111]:

```
product=yearly_sales.groupby(["Product"])
quantity_ordered=product.sum()
quantity_ordered
```

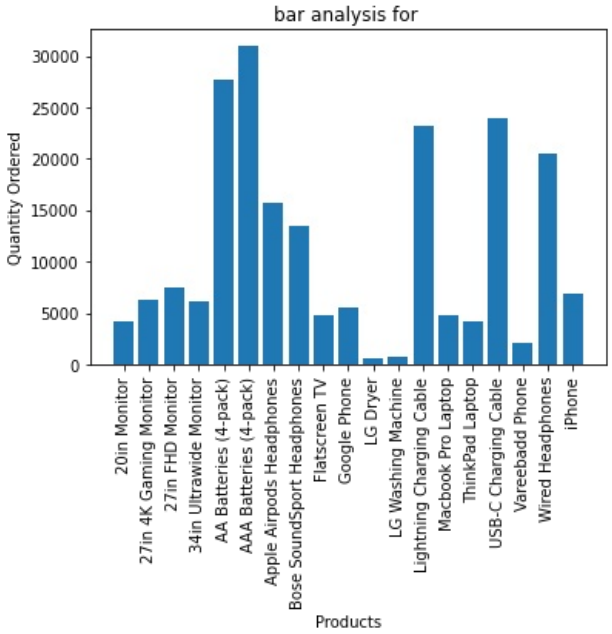
Out[111]:

| Product | Quantity Ordered | Price Each | month | sales | Hour | minute |
|---|---|---|---|---|---|---|
| 20in Monitor | 4129 | 451068.99 | 29336 | 454148.71 | 58764 | 122252 |
| 27in 4K Gaming Monitor | 6244 | 2429637.70 | 44440 | 2435097.56 | 90916 | 184331 |
| 27in FHD Monitor | 7550 | 1125974.93 | 52558 | 1132424.50 | 107540 | 219948 |
| 34in Ultrawide Monitor | 6199 | 2348718.19 | 43304 | 2355558.01 | 89076 | 183480 |
| AA Batteries (4-pack) | 27635 | 79015.68 | 145558 | 106118.40 | 298342 | 609039 |
| AAA Batteries (4-pack) | 31017 | 61716.59 | 146370 | 92740.83 | 297332 | 612113 |
| Apple Airpods Headphones | 15661 | 2332350.00 | 109477 | 2349150.00 | 223304 | 455570 |
| Bose SoundSport Headphones | 13457 | 1332366.75 | 94113 | 1345565.43 | 192445 | 392603 |
| Flatscreen TV | 4819 | 1440000.00 | 34224 | 1445700.00 | 68815 | 142789 |
| Google Phone | 5532 | 3315000.00 | 38305 | 3319200.00 | 79479 | 162773 |
| LG Dryer | 646 | 387600.00 | 4383 | 387600.00 | 9326 | 19043 |
| LG Washing Machine | 666 | 399600.00 | 4523 | 399600.00 | 9785 | 19462 |
| Lightning Charging Cable | 23217 | 323787.10 | 153092 | 347094.15 | 312529 | 634442 |
| Macbook Pro Laptop | 4728 | 8030800.00 | 33548 | 8037600.00 | 68261 | 137574 |
| ThinkPad Laptop | 4130 | 4127958.72 | 28950 | 4129958.70 | 59746 | 121508 |
| USB-C Charging Cable | 23975 | 261740.85 | 154819 | 286501.25 | 314645 | 647586 |
| Vareebadd Phone | 2068 | 826000.00 | 14309 | 827200.00 | 29472 | 61835 |
| Wired Headphones | 20557 | 226395.18 | 133397 | 246478.43 | 271720 | 554023 |
| iPhone | 6849 | 4789400.00 | 47941 | 4794300.00 | 98657 | 201688 |

## bar graph for best selling product

In [123]:

```
products=[pr for pr,df in product]
plt.bar(products,quantity_ordered["Quantity Ordered"])
plt.xticks(products,rotation="vertical")
plt.xlabel("Products")
plt.ylabel("Quantity Ordered")
plt.title("analysis for best selling product")
plt.show()
```




## making subplots

In [140]:

```
prices=yearly_sales.groupby("Product").mean()["Price Each"]
fig,ax1=plt.subplots()a
ax2=ax1.twinx()
ax1.bar(products,quantity_ordered["Quantity Ordered"],color="r")
ax2.plot(products,prices,"b-")
ax1.set_xlabel("Product",color="green")
ax1.set_ylabel("Quantity Ordered",color="r")
ax2.set_ylabel("Prices(USD)",color="b")
ax1.set_xticklabels(products,rotation="vertical")
plt.show()
```

```
<ipython-input-140-d0456a4e3bb0>:9: UserWarning: FixedFormatter should only be used together with Fi
xedLocator
  ax1.set_xticklabels(products,rotation="vertical")
```