IMPORT NECESSARY LIBRARIES

```
import pandas as pd
import os
```

Merging 12 Months of data into a single file

```
df = pd.read_csv("C:\Users\User\Desktop\PYTHON\Pandas Data Science Tasks master\SalesAnalysis\Sales_Data\Sales_April_2019.csv")

files = [f\n\n for f\n\n in os.listdir('C:\Users\User\Desktop\PYTHON\Pandas Data Science Tasks master\SalesAnalysis\Sales_Data')]

for file in files:
    print(file)
    df = pd.read_csv("C:\Users\User\Desktop\PYTHON\Pandas Data Science Tasks master\SalesAnalysis\Sales_Data\" + file)
    df.to_csv("./Sales_Data.csv", mode='a', header=False)
df.shape, df.dtypes, df.head(3)
df = pd.read_csv("./Sales_Data.csv")
df = df.rename(columns=lambda x: x.strip())
df.shape, df.dtypes, df.head(3), df.tail(3), df.info()
```

```
all_data = pd.read_csv("C:\Users\User\Desktop\PYTHON\Pandas Data Science Tasks master\SalesAnalysis\Sales_Data\Sales_April_2019.csv")
all_data.head(3)
```

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
0	176558	USB-C Charging Cable	2	11.95	04/19/19 08:46	917 1st Street, Dallas, TX 75001
1	NaN	NaN	NaN	NaN	NaN	NaN
2	176559	Bose SoundSport Headphones	1	99.99	04/07/19 22:30	682 Chestnut St, Boston, MA 02215

```
df = df[df['Order ID'].str.isnumeric]
df.head(3)
```

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
0	176558	USB-C Charging Cable	2	11.95	04/19/19 08:46	917 1st Street, Dallas, TX 75001
2	176559	Bose SoundSport Headphones	1	99.99	04/07/19 22:30	682 Chestnut St, Boston, MA 02215
3	176560	Google Phone	1	600.00	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001
4	176561	Wired Headphones	1	11.99	04/30/19 09:27	333 8th St, Los Angeles, CA 90001
5	176562	Wired Headphones	1	11.99	04/29/19 13:33	381 Mark St, Los Angeles, CA 90001

Clean the data

clean rows of NAN

```
nan_row = all_data[pd.isna(all_data['Order ID'])]
nan_row.head(3)
all_data = all_data.dropna(how='all')
all_data.head(3)
```

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
1	NaN	NaN	NaN	NaN	NaN	NaN
356	NaN	NaN	NaN	NaN	NaN	NaN
735	NaN	NaN	NaN	NaN	NaN	NaN

Find 'or' and delete it

```
temp_df = all_data[all_data['Order Date'].str.0:4 == 'Or']
temp_df.head(3)
```

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
519	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
1146	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
1486	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
1855	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
2884	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address

Remove 'Or'

```
all_data = all_data[all_data['Order Date'].str.0:4 != 'Or']
```

Argument Data With Additional Column

Add Month Column

```
all_data['Month'] = all_data['Order Date'].str.0:2
all_data.head(3)
all_data['Month'] = all_data['Month'].astype('int')
all_data.head()
```

	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 Street, 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	600.00	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001	4
4	176561	Wired Headphones	1	11.99	04/30/19 09:27	333 8th St, Los Angeles, CA 90001	4
5	176562	Wired Headphones	1	11.99	04/29/19 13:33	381 Mark St, Los Angeles, CA 90001	4

```
all_data['Quantity Ordered'] = pd.to_numeric(all_data['Quantity Ordered'])
```

```
all_data['Price Each'] = pd.to_numeric(all_data['Price Each'])
```

```
all_data['Order Date'] = pd.to_datetime(all_data['Order Date'])
```

```
all_data['Sales'] = all_data['Quantity Ordered'] * all_data['Price Each']
```

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	Sales
0	176558	USB-C Charging Cable	2	11.95	04/19/19 08:46	917 1st Street, Dallas, TX 75001	4	23.90
2	176559	Bose SoundSport Headphones	1	99.99	04/07/19 22:30	682 Chestnut St, Boston, MA 02215	4	99.99
3	176560	Google Phone	1	600.00	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001	4	600.00
4	176561	Wired Headphones	1	11.99	04/30/19 09:27	333 8th St, Los Angeles, CA 90001	4	11.99
5	176562	Wired Headphones	1	11.99	04/29/19 13:33	381 Mark St, Los Angeles, CA 90001	4	11.99
186850	259352	USB-C Charging Cable	1	11.95	09/23/19 21:35	654 Walnut St, San Francisco, CA 94016	9	11.95
186849	259351	iPhone	1	700.00	09/26/19 18:36	316 Cedar St, San Francisco, CA 94016	9	700.00
186848	259350	AAA Batteries (4-pack)	1	2.99	09/07/19 15:16	718 Ridge St, San Francisco, CA 94016	9	2.99
186847	259349	34in Ultrawide Monitor	1	379.99	09/24/19 17:32	214 Forest St, San Francisco, CA 94016	9	379.99
186846	259347	USB-C Charging Cable	1	11.95	09/10/19 00:45	548 Hickory St, San Francisco, CA 94016	9	11.95

186850 rows × 8 columns

What is the best month for sales? and how much was earned that month?

```
all_data.groupby('Month').sum('Sales', 'Price Each', 'Quantity Ordered')
```

	Quantity Ordered	Price Each	Sales
Month			
1	10903	1811768.38	1822256.73
2	13449	2188884.72	2202022.42
3	17005	2791760.92	2807100.38
4	20558	3367671.02	3390670.24
5	18667	3135875.02	3152607.65
6	15253	2562025.61	2577802.26
7	16072	2632539.56	2647775.76
8	13448	2230345.42	2244467.85
9	13109	2084992.09	2097560.13
10	22703	3715554.83	3736726.88
11	19798	3180600.68	3199603.20
12	28114	4588415.41	4613443.34

```
all_data.groupby('Month').sum('Sales', 'Price Each', 'Quantity Ordered')
```

```
import matplotlib as mpl
import matplotlib.pyplot as plt
plt.style.use('fivethirtyeight')
ax = all_data.groupby('Month').sum('Sales').plot(kind='bar')
ax.ticks_params(axis='x', rotation=0)
ax.set_ylabel('Total Sales in USD (k)')
ax.set_xlabel('Month')
ax.ticklabel_format('scientific', scientific=True, axis='y')
plt.show()
```

```
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "/usr/local/lib/python3.7/site-packages/matplotlib/axes/_base.py", line 4268, in __getattr__
    raise AttributeError("{!r} object has no attribute "
AttributeError: 'AxesSubplot' object has no attribute 'ticks_params'
>>> ax = all_data.groupby('Month').sum('Sales')
>>> ax.ticks_params(axis='x', rotation=0)
>>> ax.set_ylabel('Total Sales in USD (k)')
>>> ax.set_xlabel('Month')
>>> ax.ticklabel_format('scientific', scientific=True, axis='y')
>>> plt.show()
Traceback (most recent call last):
  File "/usr/local/lib/python3.7/site-packages/matplotlib/axes/_base.pyc", line 2610, in __getattr__
    raise AttributeError("{!r} object has no attribute "
AttributeError: 'AxesSubplot' object has no attribute 'ticklabel_format'
>>> import matplotlib.ticker as ticker
>>> ax.yaxis.set_major_formatter(ticker.ScalarFormatter())
>>> ax.ticklabel_format('scientific', scientific=True, axis='y')
Traceback (most recent call last):
  File "/usr/local/lib/python3.7/site-packages/matplotlib/axes/_base.pyc", line 2610, in __getattr__
    raise AttributeError("{!r} object has no attribute "
AttributeError: 'AxesSubplot' object has no attribute 'ticklabel_format'
>>> import matplotlib.ticker as ticker
>>> ax.yaxis.set_major_formatter(ticker.ScalarFormatter())
>>> ax.ticklabel_format('scientific', scientific=True, axis='y')
>>> plt.show()
```