# Bike store sales

## Hands on!

In [9]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

%matplotlib inline
```

In [10]:
```
df = pd.read_csv('/Users/LuckyDog/Desktop/sales_data.csv',header=0)
```

In [11]:
```
df.head()
```

Out[11]:

| | Date | Day | Month | Year | Customer_Age | Age_Group | Customer_Gender | Country | Stat |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 2013-11-26 | 26 | November | 2013 | 19 | Youth (<25) | M | Canada | Britisl Columbi |
| 1 | 2015-11-26 | 26 | November | 2015 | 19 | Youth (<25) | M | Canada | Britisl Columbi |
| 2 | 2014-03-23 | 23 | March | 2014 | 49 | Adults (35-64) | M | Australia | Nev Soutl Wale |
| 3 | 2016-03-23 | 23 | March | 2016 | 49 | Adults (35-64) | M | Australia | Nev Soutl Wale |
| 4 | 2014-05-15 | 15 | May | 2014 | 47 | Adults (35-64) | F | Australia | Nev Soutl Wale |

## What's the mean of `Customers_Age` ?

In [12]:
```
# your code goes here
df['Customer_Age'].mean()
```

Out[12]:
```
35.91921157861212
```

Why don't you try with `.mean()`

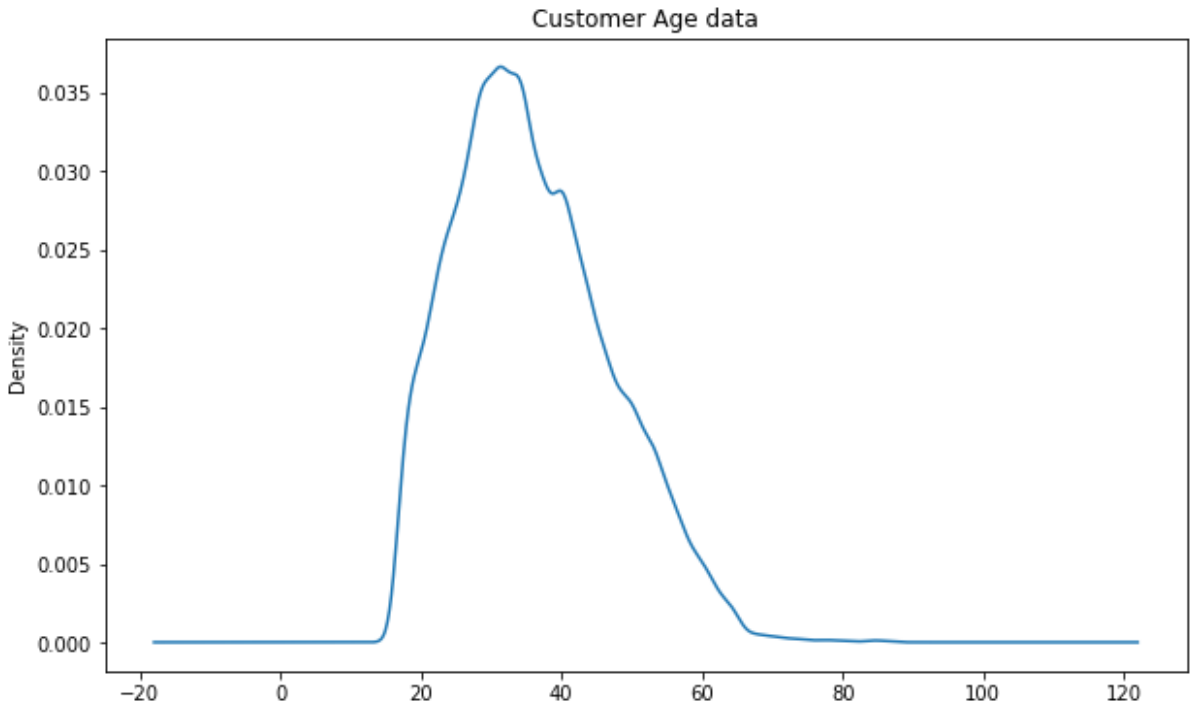
In [19]:
```
df['Customer_Age'].mean()
```

Out[19]:
```
35.91921157861212
```

Show a **density (KDE)** and a **box plot** with the `Customer_Age` data:

In [126]:
```
# your code goes here
df['Customer_Age'].plot(kind='kde',title='Customer Age data', figsize =
(10,6))
```

Out[126]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f94b20a1b90>
```




## What's the mean of `Order_Quantity` ?

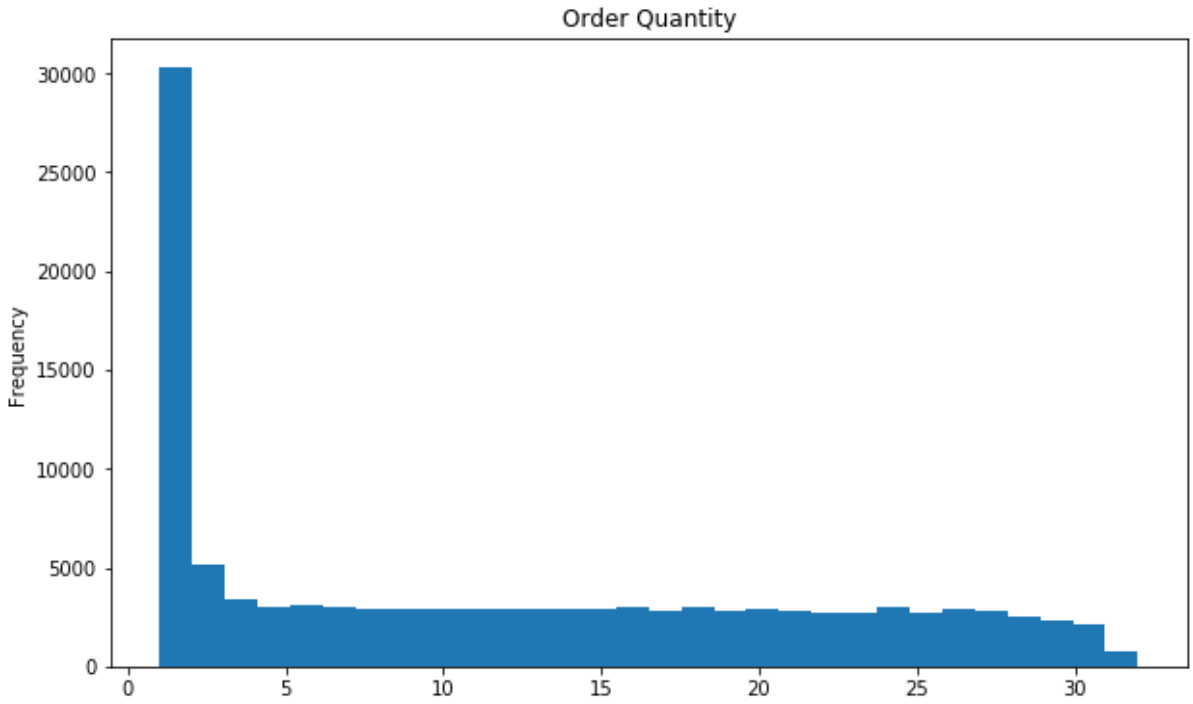
In [21]:
```
df['Order_Quantity'].mean()
```

Out[21]:
```
11.901659648253654
```

**Histogram** and a **BoxPlot** with the `Order_Quantity` data:

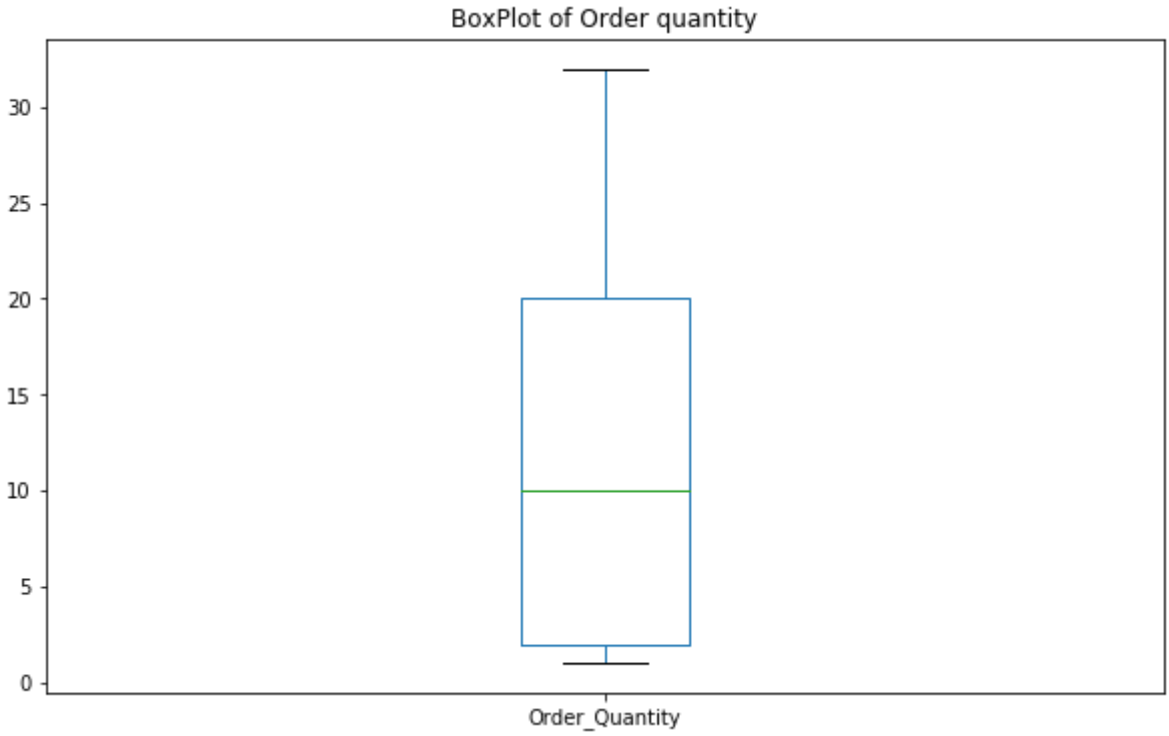
In [127]:
```
df['Order_Quantity'].plot(kind='hist',title='Order Quantity', bins=30,figsize=(10,6))
```

Out[127]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f94b077f1d0>
```



In [129]:
```
df['Order_Quantity'].plot(kind='box',title='BoxPlot of Order quantity',
figsize=(10,6))
```

Out[129]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f94b07e8e50>
```




## How many sales per year do we have?

In [24]:
```
df['Year'].value_counts()
```

Out[24]:
```
2016    29398
2014    29398
2015    24443
2013    24443
2012     2677
2011     2677
Name: Year, dtype: int64
```

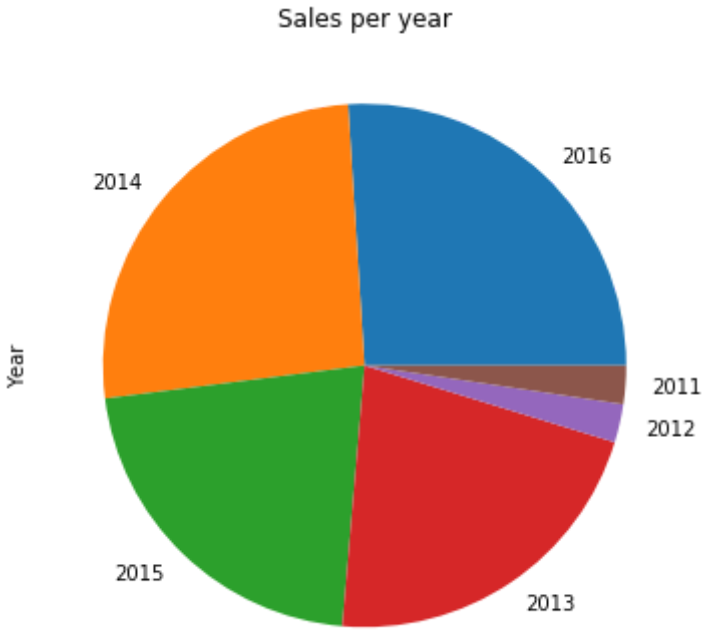
In [26]:
```
df['Year'].value_counts()
```

Out[26]:
```
2016    29398
2014    29398
2015    24443
2013    24443
2012     2677
2011     2677
Name: Year, dtype: int64
```

Go ahead and show a **pie plot** with the previous data:

In [117]:
```
df['Year'].value_counts().plot(kind='pie',title='Sales per year', figsize=(6,6))
```

Out[117]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f94af4ea5d0>
```



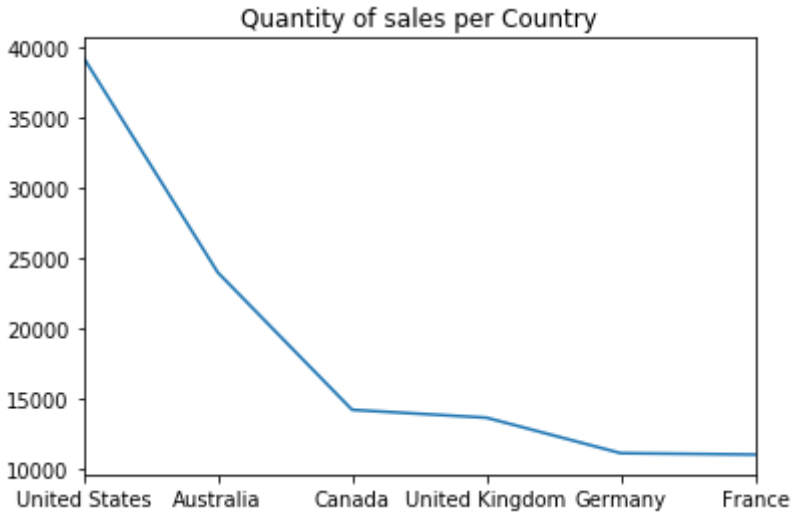

In [29]:
```
df['Month'].value_counts()
```

Out[29]:
```
June         11234
December     11200
May          11128
April        10182
March         9674
January       9284
February      9022
October       8750
November      8734
August        8200
September     8166
July          7462
Name: Month, dtype: int64
```

## Which country has the most sales `quantity of sales` ?

In [115]:
```
df['Country'].value_counts().plot(title='Quantity of sales per Country')
```

Out[115]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f94af064a50>
```




In [32]:
```
df['Country'].value_counts().head(1)
```

Out[32]:
```
United States    39206
Name: Country, dtype: int64
```

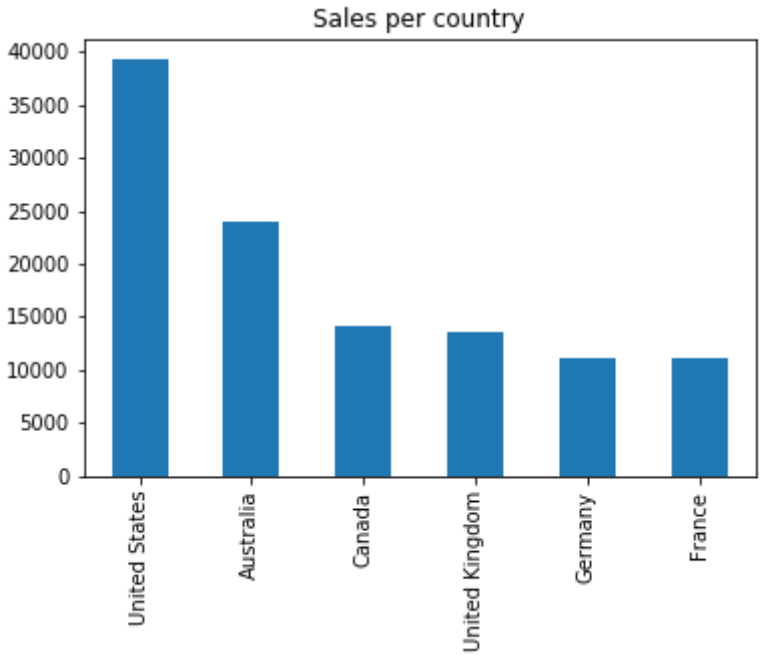
In [34]:
```
df['Country'].value_counts()
```

Out[34]:
```
United States     39206
Australia         23936
Canada            14178
United Kingdom    13620
Germany           11098
France            10998
Name: Country, dtype: int64
```

# Bar plot of the sales per country:

In [114]:
```
df['Country'].value_counts().plot(kind='bar', title='Sales per country')
```

Out[114]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f94af02cb90>
```




In [ ]:
```
sales['Country'].value_counts().plot(kind='bar', figsize=(14,6))
```

## Create a list of every product sold

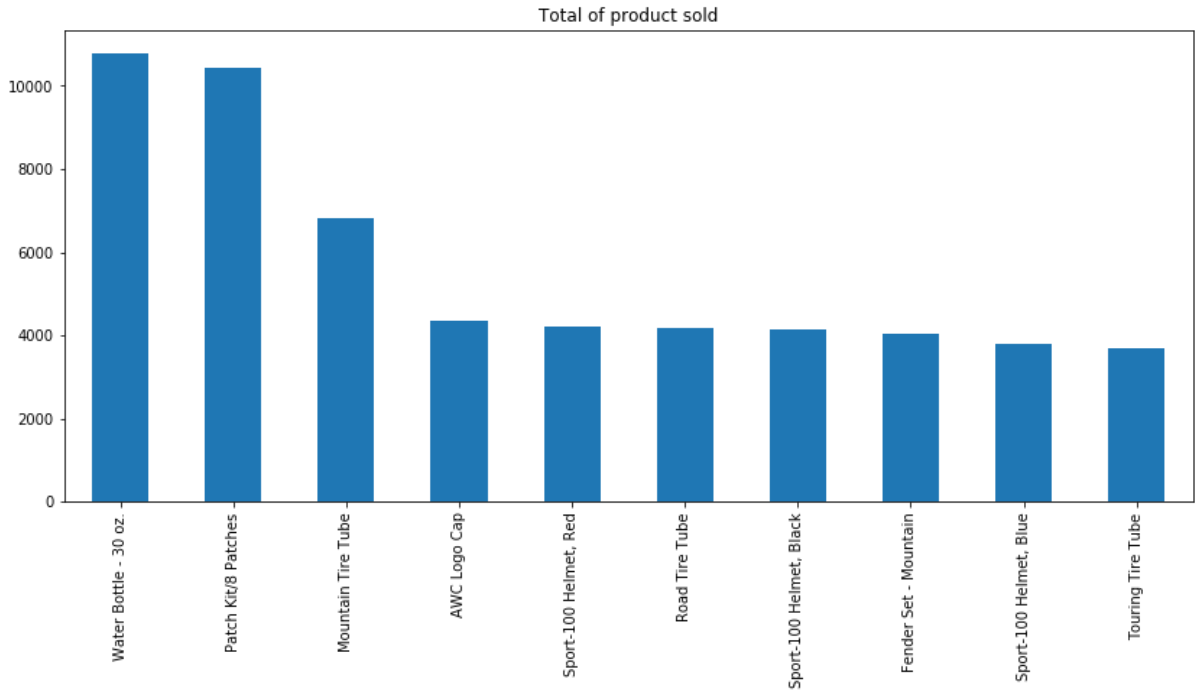
In [37]:
```
df['Product'].unique()
```

Out[37]:
```
array(['Hitch Rack - 4-Bike', 'All-Purpose Bike Stand',
       'Mountain Bottle Cage', 'Water Bottle - 30 oz.',
       'Road Bottle Cage', 'AWC Logo Cap', 'Bike Wash - Dissolver',
       'Fender Set - Mountain', 'Half-Finger Gloves, L',
       'Half-Finger Gloves, M', 'Half-Finger Gloves, S',
       'Sport-100 Helmet, Black', 'Sport-100 Helmet, Red',
       'Sport-100 Helmet, Blue', 'Hydration Pack - 70 oz.',
       'Short-Sleeve Classic Jersey, XL',
       'Short-Sleeve Classic Jersey, L', 'Short-Sleeve Classic Jersey,
M',
       'Short-Sleeve Classic Jersey, S', 'Long-Sleeve Logo Jersey, M',
       'Long-Sleeve Logo Jersey, XL', 'Long-Sleeve Logo Jersey, L',
       'Long-Sleeve Logo Jersey, S', 'Mountain-100 Silver, 38',
       'Mountain-100 Silver, 44', 'Mountain-100 Black, 48',
       'Mountain-100 Silver, 48', 'Mountain-100 Black, 38',
       'Mountain-200 Silver, 38', 'Mountain-100 Black, 44',
       'Mountain-100 Silver, 42', 'Mountain-200 Black, 46',
       'Mountain-200 Silver, 42', 'Mountain-200 Silver, 46',
       'Mountain-200 Black, 38', 'Mountain-100 Black, 42',
       'Mountain-200 Black, 42', 'Mountain-400-W Silver, 46',
       'Mountain-500 Silver, 40', 'Mountain-500 Silver, 44',
       'Mountain-500 Black, 48', 'Mountain-500 Black, 40',
       'Mountain-400-W Silver, 42', 'Mountain-500 Silver, 52',
       'Mountain-500 Black, 52', 'Mountain-500 Silver, 42',
       'Mountain-500 Black, 44', 'Mountain-500 Silver, 48',
       'Mountain-400-W Silver, 38', 'Mountain-400-W Silver, 40',
       'Mountain-500 Black, 42', 'Road-150 Red, 48', 'Road-150 Red, 6
2',
       'Road-750 Black, 48', 'Road-750 Black, 58', 'Road-750 Black, 5
2',
       'Road-150 Red, 52', 'Road-150 Red, 44', 'Road-150 Red, 56',
       'Road-750 Black, 44', 'Road-350-W Yellow, 40',
       'Road-350-W Yellow, 42', 'Road-250 Black, 44',
       'Road-250 Black, 48', 'Road-350-W Yellow, 48',
       'Road-550-W Yellow, 44', 'Road-550-W Yellow, 38',
       'Road-250 Black, 52', 'Road-550-W Yellow, 48', 'Road-250 Red, 5
8',
       'Road-250 Black, 58', 'Road-250 Red, 52', 'Road-250 Red, 48',
       'Road-250 Red, 44', 'Road-550-W Yellow, 42',
       'Road-550-W Yellow, 40', 'Road-650 Red, 48', 'Road-650 Red, 60',
       'Road-650 Black, 48', 'Road-350-W Yellow, 44', 'Road-650 Red, 5
2',
       'Road-650 Black, 44', 'Road-650 Red, 62', 'Road-650 Red, 58',
       'Road-650 Black, 60', 'Road-650 Black, 58', 'Road-650 Black, 5
2',
       'Road-650 Black, 62', 'Road-650 Red, 44',
       "Women's Mountain Shorts, M", "Women's Mountain Shorts, S",
       "Women's Mountain Shorts, L", 'Racing Socks, L', 'Racing Socks,
M',
       'Mountain Tire Tube', 'Touring Tire Tube', 'Patch Kit/8 Patche
s',
       'HL Mountain Tire', 'LL Mountain Tire', 'Road Tire Tube',
       'LL Road Tire', 'Touring Tire', 'ML Mountain Tire', 'HL Road Tir
e',
       'ML Road Tire', 'Touring-1000 Yellow, 50', 'Touring-1000 Blue, 4
6',
       'Touring-1000 Yellow, 60', 'Touring-1000 Blue, 50',
```

```
       'Touring-3000 Yellow, 50', 'Touring-3000 Blue, 54',
       'Touring-3000 Blue, 58', 'Touring-3000 Yellow, 44',
       'Touring-3000 Yellow, 54', 'Touring-3000 Blue, 62',
       'Touring-3000 Blue, 44', 'Touring-1000 Blue, 54',
       'Touring-1000 Yellow, 46', 'Touring-1000 Blue, 60',
       'Touring-3000 Yellow, 62', 'Touring-1000 Yellow, 54',
       'Touring-2000 Blue, 54', 'Touring-3000 Blue, 50',
       'Touring-3000 Yellow, 58', 'Touring-2000 Blue, 46',
       'Touring-2000 Blue, 50', 'Touring-2000 Blue, 60',
       'Classic Vest, L', 'Classic Vest, M', 'Classic Vest, S'],
      dtype=object)
```

In [113]:
```
df['Product'].value_counts().head(10).plot(kind='bar',title='Total of pr
oduct sold', figsize=(14,6))
```

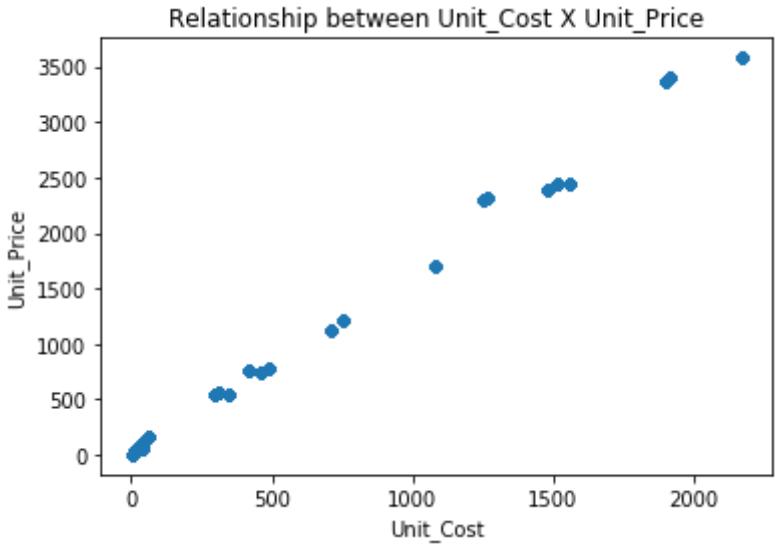
Out[113]: <matplotlib.axes._subplots.AxesSubplot at 0x7f94aec2b0d0>




## Can you see any relationship between `Unit_Cost` and `Unit_Price`?

In [112]:
```
df.plot(kind='scatter',x='Unit_Cost',y='Unit_Price', title='Relationship between Unit_Cost X Unit_Price')
```

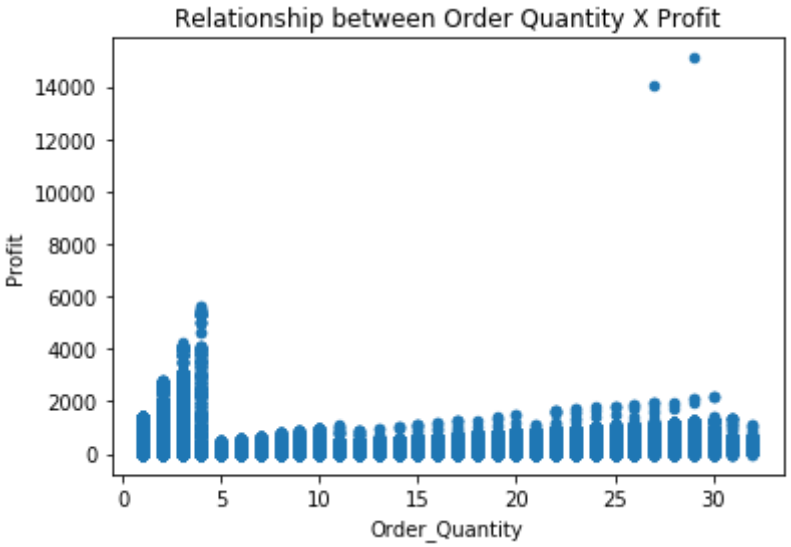
Out[112]: <matplotlib.axes._subplots.AxesSubplot at 0x7f94af18fe90>




## Can you see any relationship between `Order_Quantity` and `Profit`?

In [111]:
```
df.plot(kind='scatter',x='Order_Quantity',y='Profit', title='Relationship between Order Quantity X Profit')
```

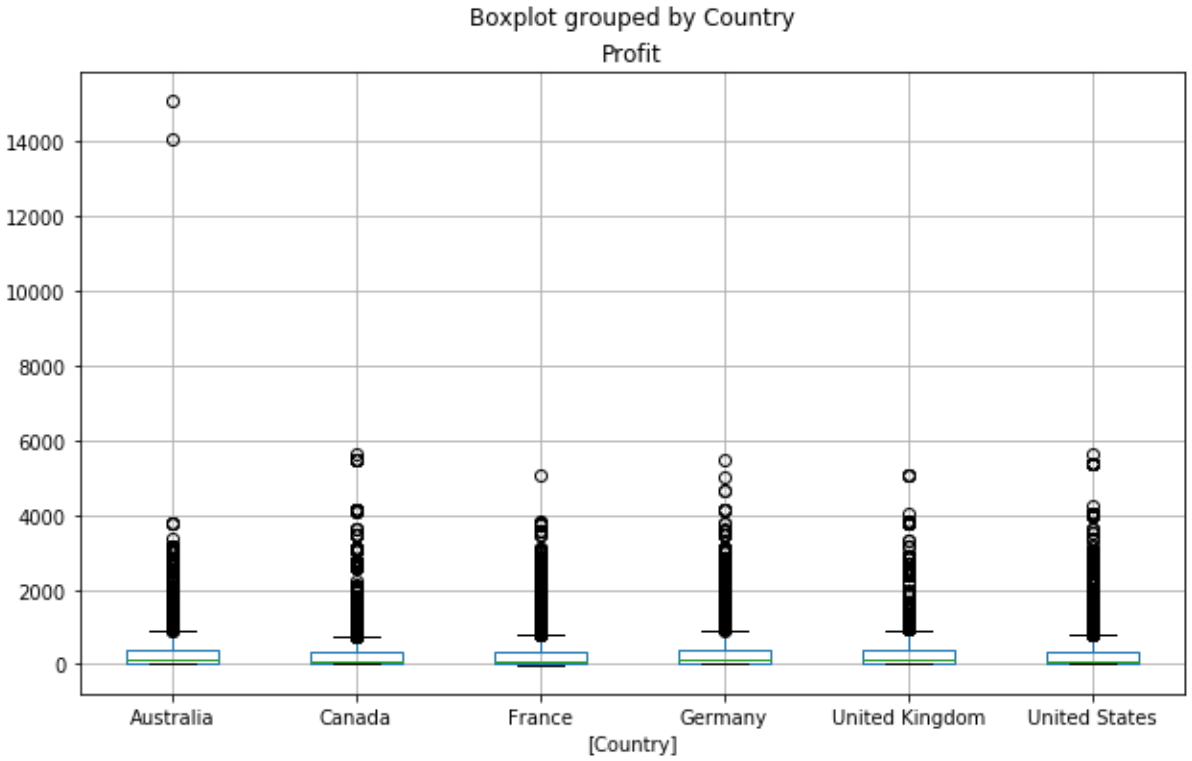
Out[111]: <matplotlib.axes._subplots.AxesSubplot at 0x7f94ad0b9b90>




## Can you see any relationship between `Profit` per `Country`?

In [62]:
```
df[['Profit','Country']].boxplot(by='Country',figsize=(10,6))
```

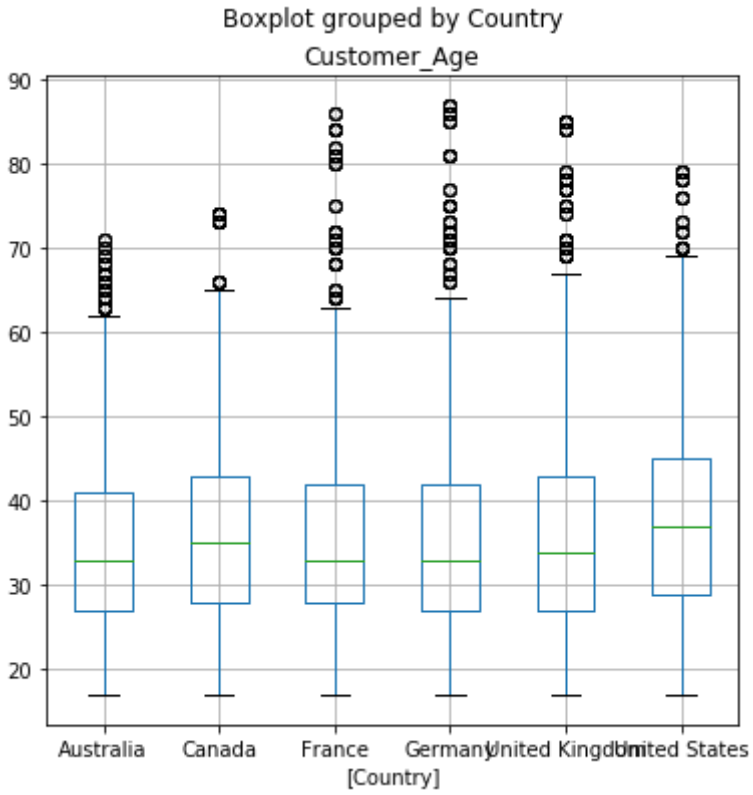
Out[62]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f94ad976390>
```




## Can you see any relationship between the `Customer_Age` per `Country`?

In [56]:
```
df[['Customer_Age','Country']].boxplot(by='Country',figsize=(6,6))
```

Out[56]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f94ac417090>
```




## Add and calculate a new `Calculated_Date` column

Use `Day`, `Month`, `Year` to create a `Date` column (`YYYY-MM-DD`).

In [73]:
```
df['Calculated_Date'] = df[['Year', 'Month', 'Day']].apply(lambda x:'{}-{}-{}'.format(x[0],x[1],x[2]), axis=1)
df['Calculated_Date'].head()
```

Out[73]:
```
0    2013-November-26
1    2015-November-26
2       2014-March-23
3       2016-March-23
4         2014-May-15
Name: Calculated_Date, dtype: object
```

## Parse your `Calculated_Date` column into a datetime object

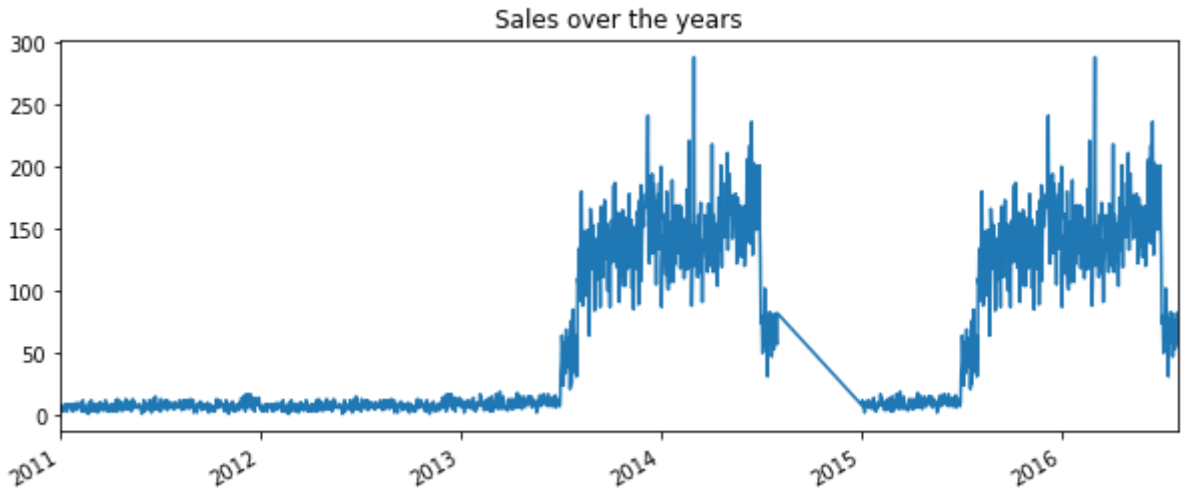
In [74]:
```
df['Calculated_Date']=pd.to_datetime(df['Calculated_Date'])
df['Calculated_Date'].head()
```

Out[74]:
```
0   2013-11-26
1   2015-11-26
2   2014-03-23
3   2016-03-23
4   2014-05-15
Name: Calculated_Date, dtype: datetime64[ns]
```

## How did sales evolve through the years?

In [108]:
```
df['Calculated_Date'].value_counts().plot(kind='line', title='Sales over the years',figsize=(10,4))
```

Out[108]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f94ae493dd0>
```




## Increase 50 U$S revenue to every sale

In [78]:
```
df['Revenue'] += 50
```

## How many orders were made in `Canada` or `France`?

In [96]:
```
df.loc[(df['Country']=='Canada') | (df['Country']=='France')].shape[0]
```

Out[96]:
```
25176
```

## How many `Bike Racks` orders were made from Canada?

In [87]:
```
df.loc[(df['Country']=='Canada') & (df['Sub_Category']=='Bike Racks')].shape[0]
```

Out[87]:
```
104
```

## How many orders were made in each region (state) of France?

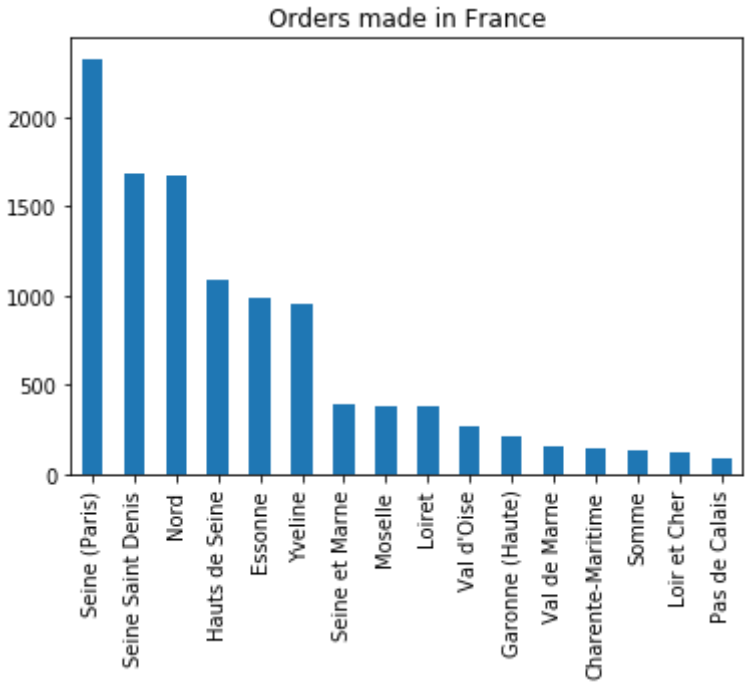
In [99]:
```
orders_france = df.loc[df['Country']=='France','State'].value_counts()
orders_france
```

Out[99]:
```
Seine (Paris)        2328
Seine Saint Denis    1684
Nord                 1670
Hauts de Seine       1084
Essonne               994
Yveline               954
Seine et Marne        394
Moselle               386
Loiret                382
Val d'Oise            264
Garonne (Haute)       208
Val de Marne          158
Charente-Maritime     148
Somme                 134
Loir et Cher          120
Pas de Calais          90
Name: State, dtype: int64
```

Go ahead and show a **bar plot** with the results:

In [107]:
```
orders_france.plot(kind='bar',title='Orders made in France', figsize=(6, 4))
```

Out[107]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f94aedba2d0>
```




## How many sales were made per category?

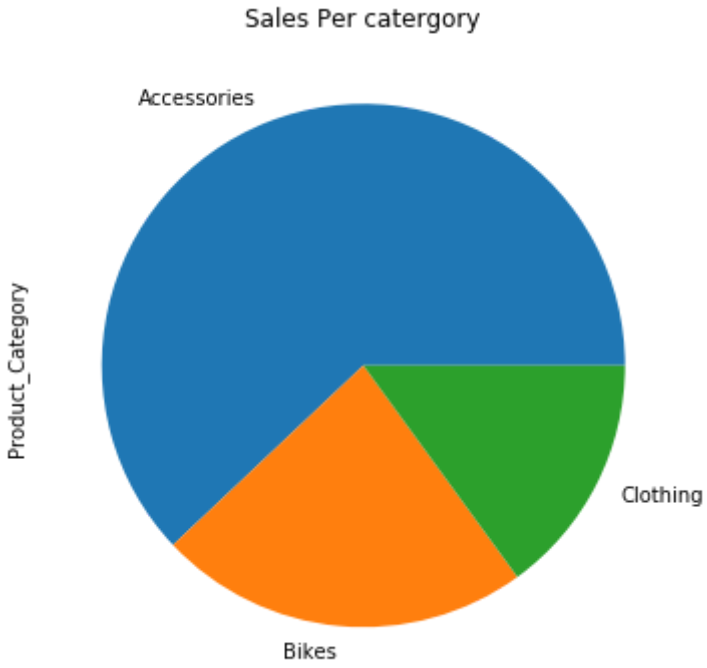
In [102]:
```
df['Product_Category'].value_counts()
```

Out[102]:
```
Accessories    70120
Bikes          25982
Clothing       16934
Name: Product_Category, dtype: int64
```

Go ahead and show a **pie plot** with the results:

In [106]:
```
df['Product_Category'].value_counts().plot(kind='pie',title='Sales Per catergory', figsize=(10,6))
```

Out[106]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f94ae973fd0>
```




## How many orders were made per accessory sub-categories?

In [131]:
```
# your code goes here
accessories=df.loc[df['Product_Category']=='Accessories','Sub_Category'].value_counts()
accessories
```

Out[131]:
```
Tires and Tubes      33870
Bottles and Cages    15876
Helmets              12158
Fenders               4032
Cleaners              1802
Hydration Packs       1334
Bike Racks             592
Bike Stands            456
Name: Sub_Category, dtype: int64
```

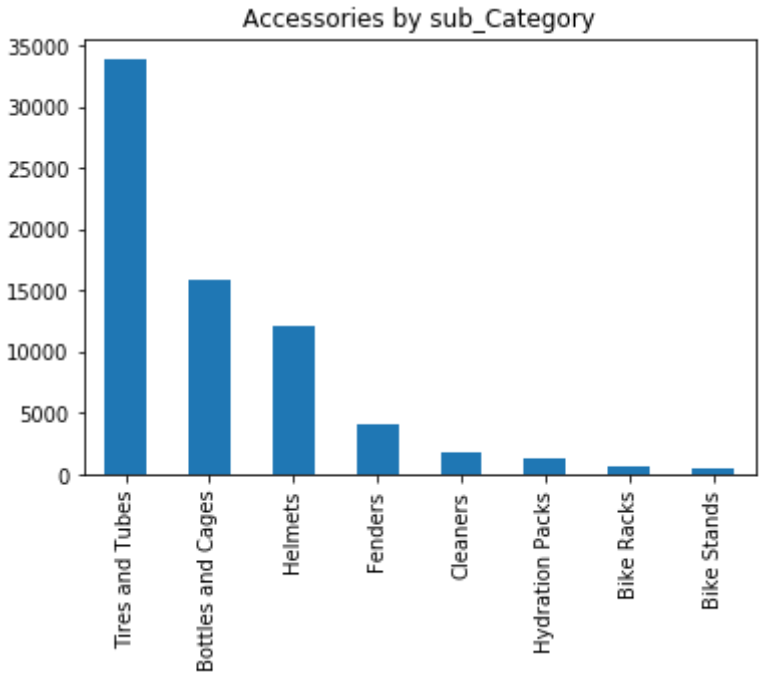
In [ ]:
```

```

Go ahead and show a **bar plot** with the results:

In [133]:
```
# your code goes here
accessories.plot(kind='bar', title='Accessories by sub_Category')
```

Out[133]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f94af4f4c90>
```




In [ ]:
```

```

## How many orders were made per bike sub-categories?

In [135]:
```
# your code goes here
bike=df.loc[df['Product_Category']=='Bikes','Sub_Category'].value_counts
()
bike
```

Out[135]:
```
Road Bikes        13430
Mountain Bikes     8854
Touring Bikes      3698
Name: Sub_Category, dtype: int64
```

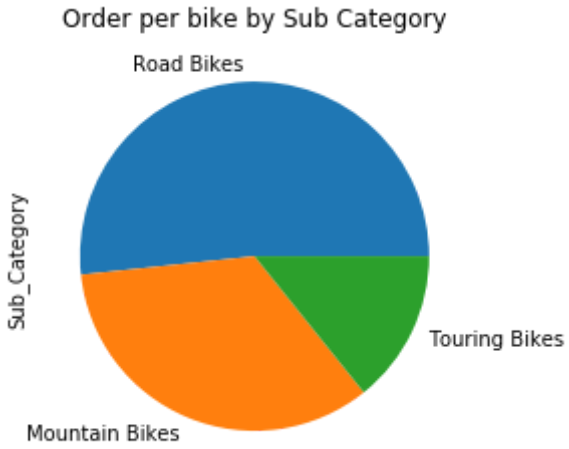
In [ ]:
```

```

Go ahead and show a **pie plot** with the results:

In [137]:
```
# your code goes here
bike.plot(kind='pie', title='Order per bike by Sub Category')
```

Out[137]: <matplotlib.axes._subplots.AxesSubplot at 0x7f94ad6f4b50>




In [ ]:
```
```

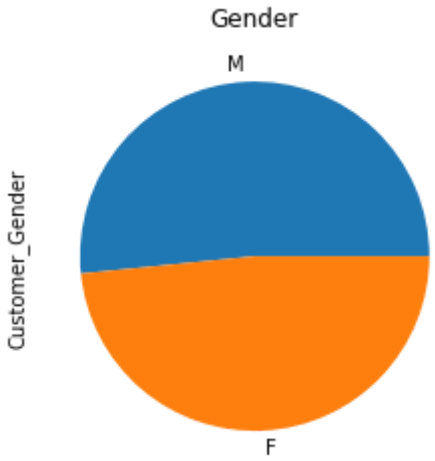
## Which gender has the most amount of sales?

In [140]:
```
# your code goes here
df['Customer_Gender'].value_counts()
```

Out[140]:
```
M    58312
F    54724
Name: Customer_Gender, dtype: int64
```

In [143]:
```
df['Customer_Gender'].value_counts().plot(kind='pie', title='Gender')
```

Out[143]: <matplotlib.axes._subplots.AxesSubplot at 0x7f94b1eb07d0>



In [155]:
```
df.loc[(df['Customer_Gender'] =='M') & (df['Revenue'] == 500)].shape[1]
```

Out[155]: 19

## Get the top-5 sales with the highest revenue

In [156]:
```
df.sort_values(['Revenue'],ascending=False).head(5)
```

Out[156]:

| | Date | Day | Month | Year | Customer_Age | Age_Group | Customer_Gender | Country | |
|---|---|---|---|---|---|---|---|---|---|
| 112073 | 2015-07-24 | 24 | July | 2015 | 52 | Adults (35-64) | M | Australia | Q |
| 112072 | 2013-07-24 | 24 | July | 2013 | 52 | Adults (35-64) | M | Australia | Q |
| 71129 | 2011-07-08 | 8 | July | 2011 | 22 | Youth (<25) | M | Canada | |
| 70307 | 2011-04-30 | 30 | April | 2011 | 44 | Adults (35-64) | M | Canada | |
| 70601 | 2011-09-30 | 30 | September | 2011 | 19 | Youth (<25) | F | Canada | |

In [ ]:
```
```

## Get the sale with the highest revenue

In [165]:
```
df.max()
```

Out[165]:
```
Date                                2016-07-31
Day                                         31
Month                                September
Year                                      2016
Customer_Age                                87
Age_Group                       Youth (<25)
Customer_Gender                             M
Country                         United States
State                                 Yveline
Product_Category                     Clothing
Sub_Category                            Vests
Product             Women's Mountain Shorts, S
Order_Quantity                              32
Unit_Cost                                 2171
Unit_Price                                3578
Profit                                   15096
Cost                                     42978
Revenue                                  58124
Calculated_Date            2016-07-31 00:00:00
dtype: object
```

In [164]:
```
cond = df['Revenue']== df['Revenue'].max()
df.loc[cond]
```

Out[164]:

| | Date | Day | Month | Year | Customer_Age | Age_Group | Customer_Gender | Country | |
|---|---|---|---|---|---|---|---|---|---|
| 112073 | 2015-07-24 | 24 | July | 2015 | 52 | Adults (35-64) | M | Australia | Queer |

## What is the mean `Order_Quantity` of orders with more than 10K in revenue?

In [166]:
```
# your code goes here
cond=df['Revenue']>10000
df.loc[cond,'Order_Quantity'].mean()
```

Out[166]:
```
3.689265536723164
```

## What is the mean `Order_Quantity` of orders with less than 10K in revenue?

In [167]:
```
# your code goes here
cond =df['Revenue']<10000
df.loc[cond,"Order_Quantity"].mean()
```

Out[167]:
```
11.914539380997528
```

## How many orders were made in May of 2016?

In [174]:
```
cond=(df['Year']==2016)&(df['Month']=='May')
df.loc[cond].shape[0]
```

Out[174]:
```
5015
```

## How many orders were made between May and July of 2016?

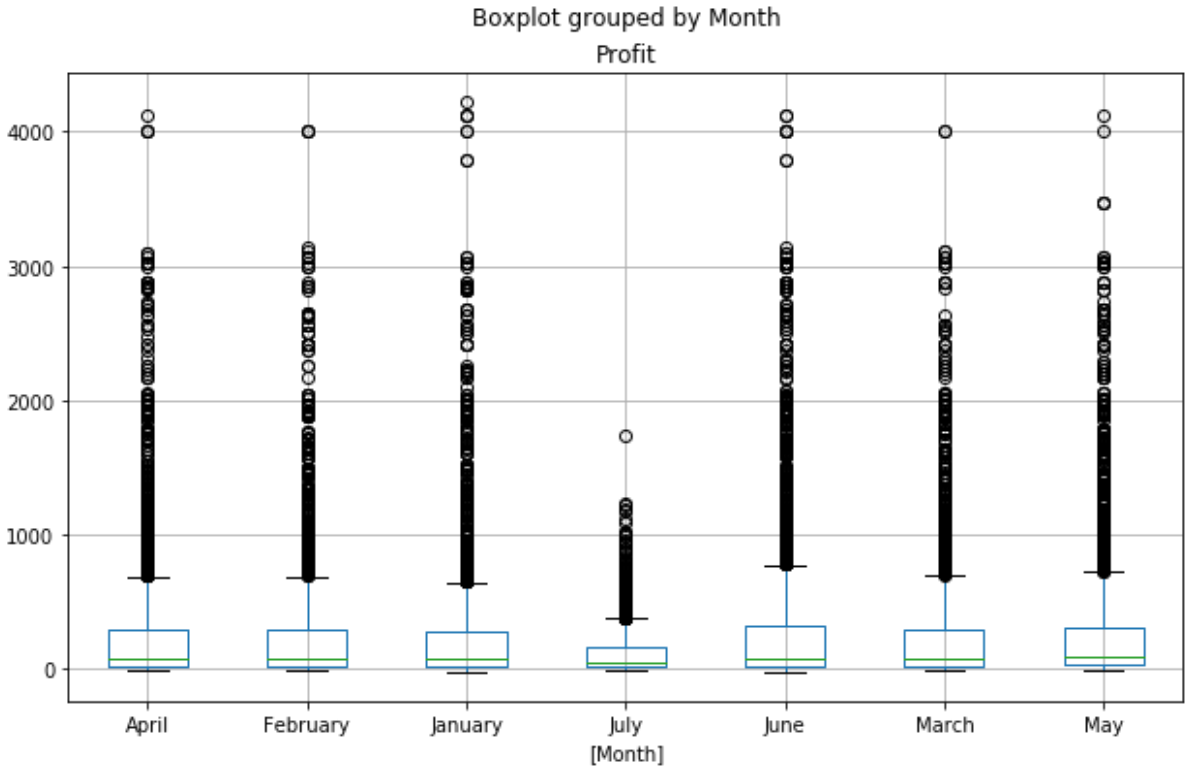
In [176]:
```
cond1=(df["Year"]== 2016) & (df["Month"].isin(['May','June','July']))
df.loc[cond1].shape[0]
```

Out[176]:
```
12164
```

Show a grouped **box plot** per month with the profit values.

In [181]:
```
profit= df.loc[df['Year'] == 2016,['Profit','Month']]
profit.boxplot(by="Month", figsize=(10,6))
```

Out[181]: <matplotlib.axes._subplots.AxesSubplot at 0x7f94b2116190>




## Add 7.2% TAX on every sale `Unit_Price` within United States

In [185]:
```
df.loc[df['Country']=='United States','Unit_Price']*= 1.072
```