In [3]:

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

%matplotlib inline
```

In [4]:

```
sales = pd.read_csv(r'C:\Users\Dator\Desktop\Data Analysis With Python\sales_data.cs
```

In [5]:

```
print("Bike store sales Data")
sales.head()
```

```
Bike store sales Data
```

Out[5]:

	Date	Day	Month	Year	Customer_Age	Age_Group	Customer_Gender	Country	State
0	2013-11-26	26	November	2013	19	Youth (<25)	M	Canada	British Columbia
1	2015-11-26	26	November	2015	19	Youth (<25)	M	Canada	British Columbia
2	2014-03-23	23	March	2014	49	Adults (35-64)	M	Australia	New South Wales
3	2016-03-23	23	March	2016	49	Adults (35-64)	M	Australia	New South Wales
4	2014-05-15	15	May	2014	47	Adults (35-64)	F	Australia	New South Wales

Customers_Average_Age

In [12]:

```
print("Averageof Customers_Age is :")
sales['Customer_Age'].mean()
```

```
Averageof Customers_Age is :
```

Out[12]:

```
35.91921157861212
```

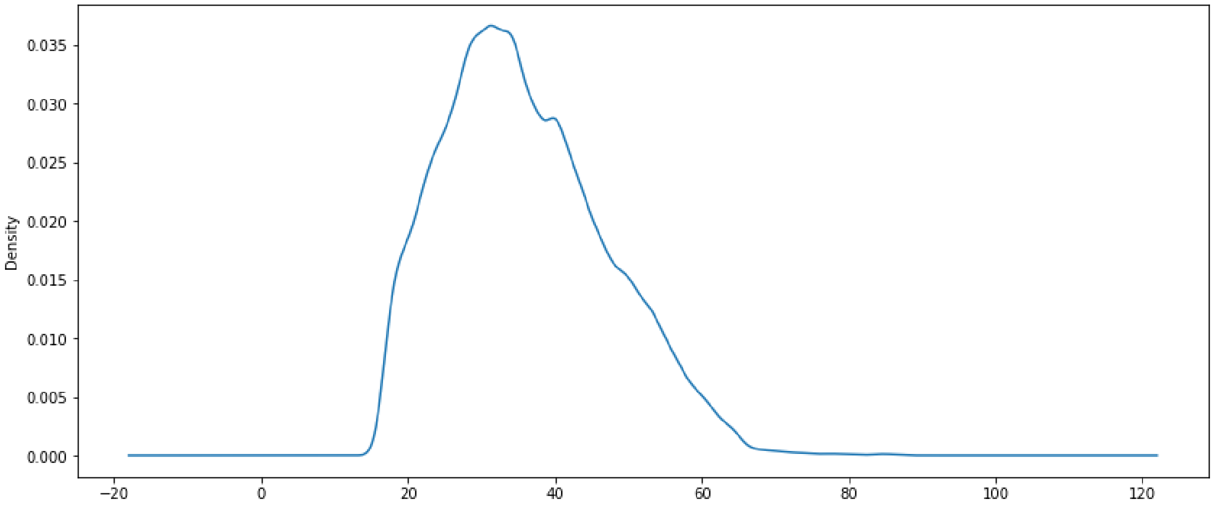
Density (KDE) and a box plot with the Customer_Age data:

In [8]:

```
sales['Customer_Age'].plot(kind='kde', figsize=(14,6))
```

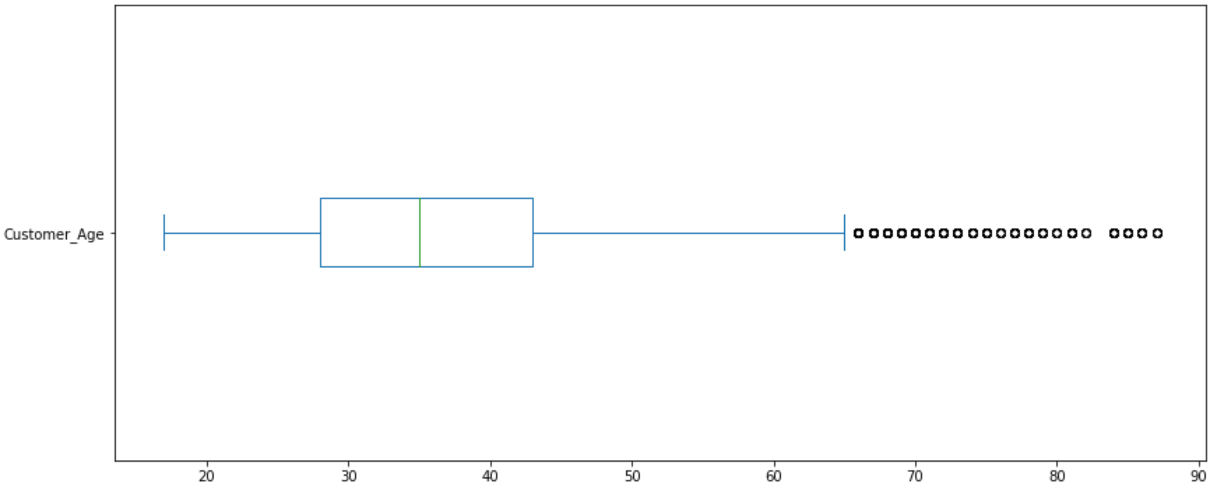
Out[8]:

```
<AxesSubplot:ylabel='Density'>
```



```
In [9]: sales['Customer_Age'].plot(kind='box', vert=False, figsize=(14,6))
```

```
Out[9]: <AxesSubplot:>
```



Average of Order_Quantity

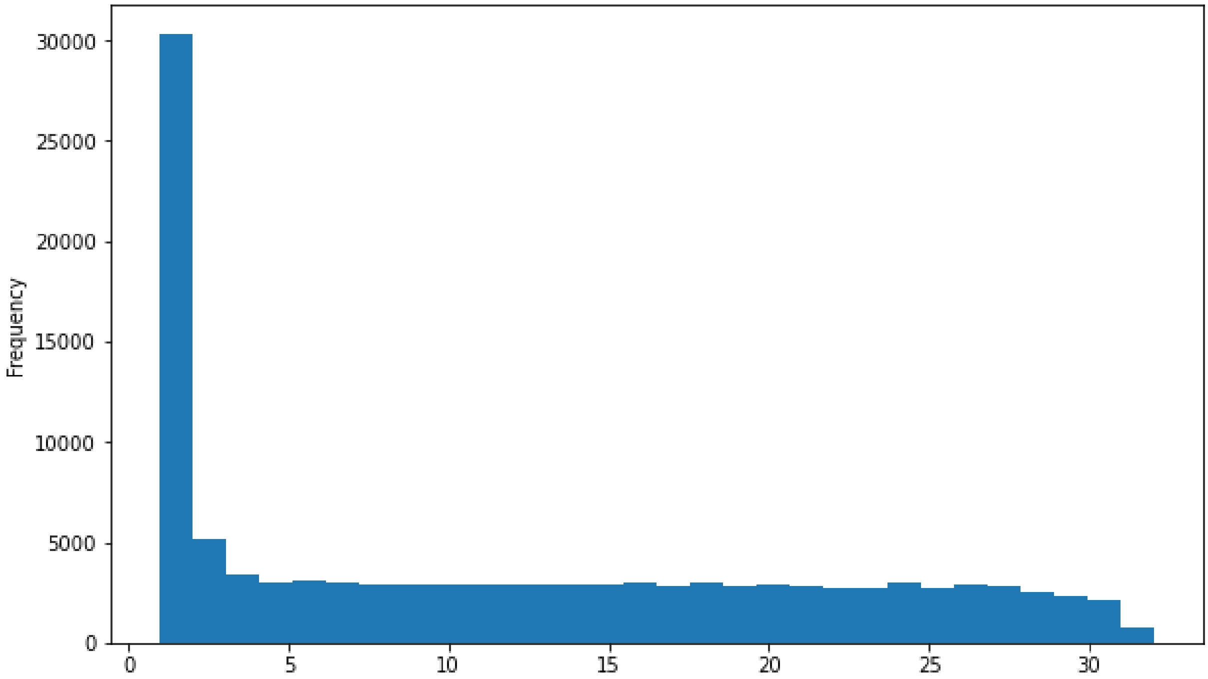
```
In [13]: print("Average of Order_Quantity is")
         sales['Order_Quantity'].mean()
```

```
Average of Order_Quantity is
Out[13]: 11.901659648253654
```

Histogram and box plot with the Order_Quantity data:

```
In [14]: sales['Order_Quantity'].plot(kind='hist', bins=30, figsize=(10,6))
```

```
Out[14]: <AxesSubplot:ylabel='Frequency'>
```

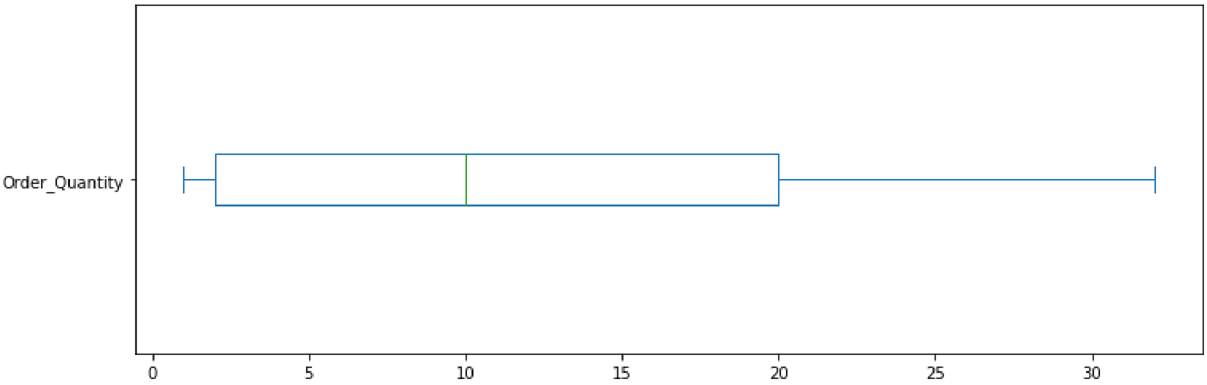


In [20]:

```
sales['Order_Quantity'].plot(kind='box', vert=False, figsize=(12,4))
```

Out[20]:

```
<AxesSubplot:>
```



Sales per year

In [20]:

```
print("Sales per year")

sales['Year'].value_counts(ascending = False)
```

Out[20]:

```
Sales per year
2014    29398
2016    29398
2013    24443
2015    24443
2012     2677
2011     2677
Name: Year, dtype: int64
```

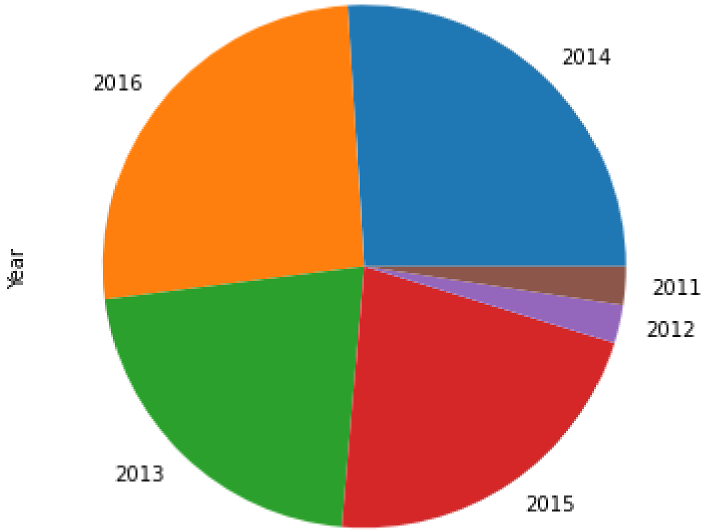
Pie plot with the Sales per Year:

In [21]:

```
sales['Year'].value_counts().plot(kind='pie', figsize=(6,6))
```

Out[21]:

```
<AxesSubplot:ylabel='Year'>
```



Sales per Month

In [22]:

```
print("Sales per Month")
sales['Month'].value_counts()
```

Out[22]:

```
Sales per Month
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
```

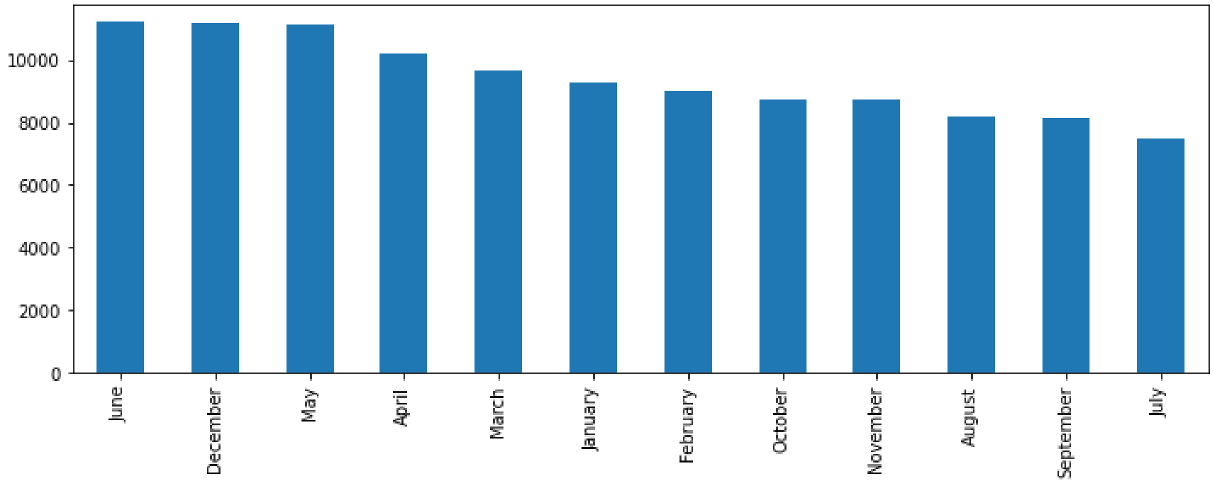
Bar plot with the Sales per Month:

In [19]:

```
sales['Month'].value_counts().plot(kind='bar', figsize=(12,4))
```

Out[19]:

```
<AxesSubplot:>
```



Country has the most sales quantity of sales

In [23]:
```
print("Most sales quantity country")
sales['Country'].value_counts().head(1)
```

Out[23]:
```
Most sales quantity country
United States    39206
Name: Country, dtype: int64
```

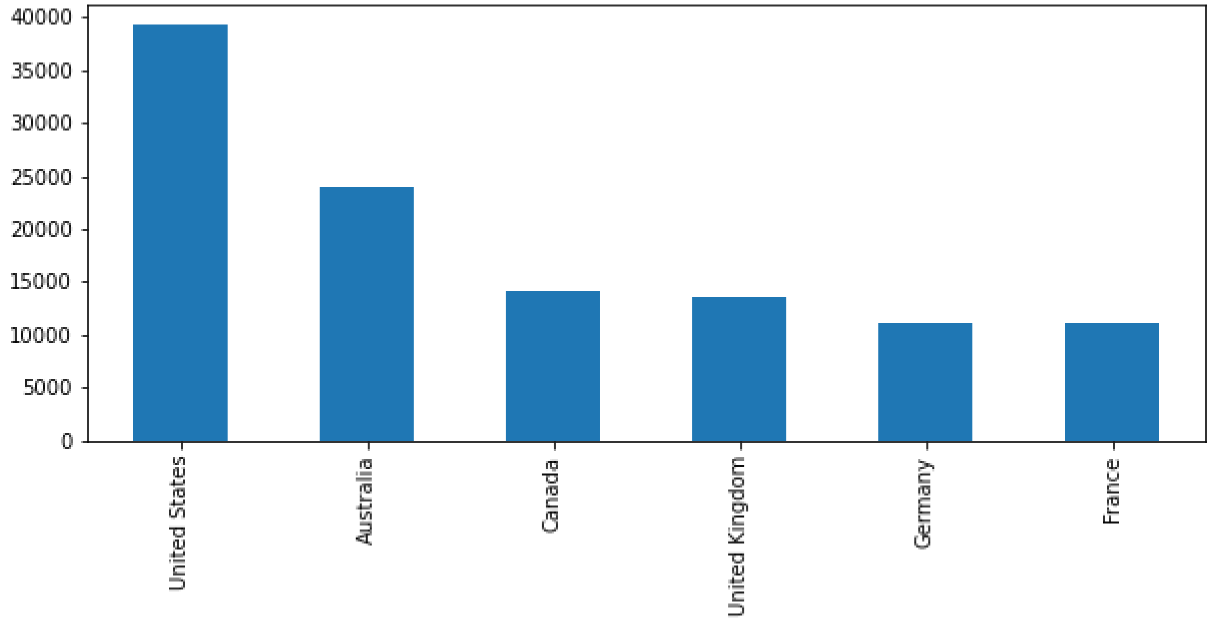
In [20]:
```
print("Country wise sales")
sales['Country'].value_counts()
```

Out[20]:
```
Country wise sales
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 [28]:
```
sales['Country'].value_counts().plot(kind='bar', figsize=(10,4))
```

Out[28]:
```
<AxesSubplot:>
```



List of every product sold

In [25]:

```
print("List of every product sold:")

sales['Product'].unique()
```

```
        List of every product sold:
Out[25]: 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, 62',
               'Road-750 Black, 48', 'Road-750 Black, 58', 'Road-750 Black, 52',
               '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, 58',
               '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, 52',
               'Road-650 Black, 44', 'Road-650 Red, 62', 'Road-650 Red, 58',
               'Road-650 Black, 60', 'Road-650 Black, 58', 'Road-650 Black, 52',
               '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 Patches',
               'HL Mountain Tire', 'LL Mountain Tire', 'Road Tire Tube',
               'LL Road Tire', 'Touring Tire', 'ML Mountain Tire', 'HL Road Tire',
               'ML Road Tire', 'Touring-1000 Yellow, 50', 'Touring-1000 Blue, 46',
               '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)
```

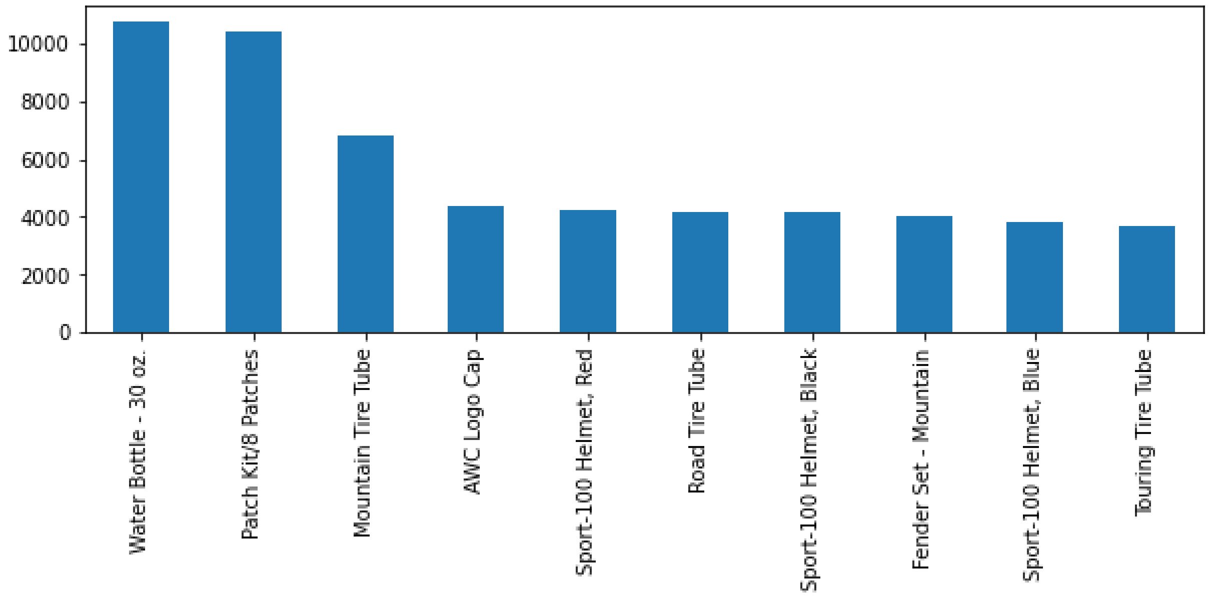
Bar plot showing the 10 most sold products (best sellers):

In [29]:

```
sales['Product'].value_counts().head(10).plot(kind='bar', figsize=(10,3))
```

Out[29]:

```
<AxesSubplot:>
```



Relationship between Unit_Cost and Unit_Price

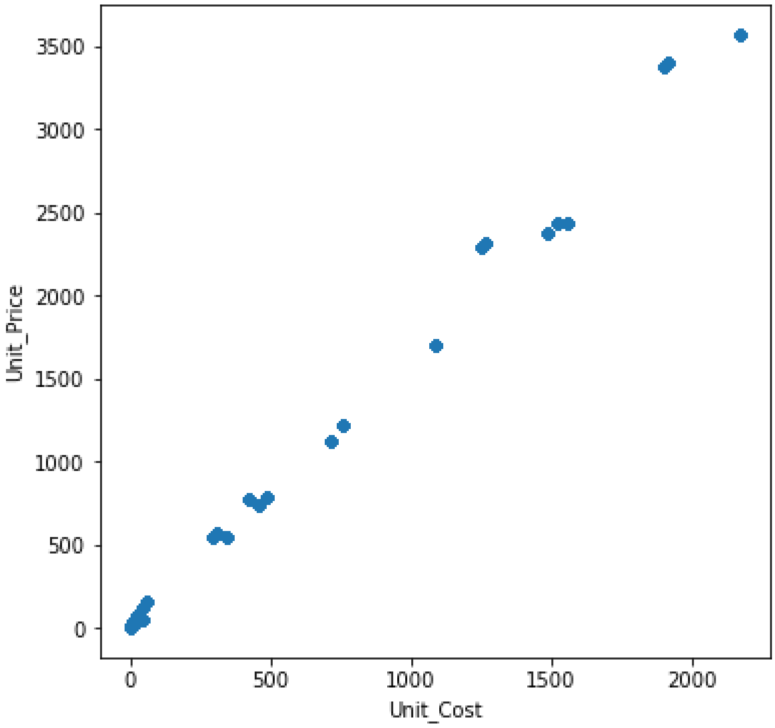
- Scatter plot between Unit_Cost and Unit_price.

In [30]:

```
sales.plot(kind='scatter', x='Unit_Cost', y='Unit_Price', figsize=(6,6))
```

Out[30]:

```
<AxesSubplot:xlabel='Unit_Cost', ylabel='Unit_Price'>
```

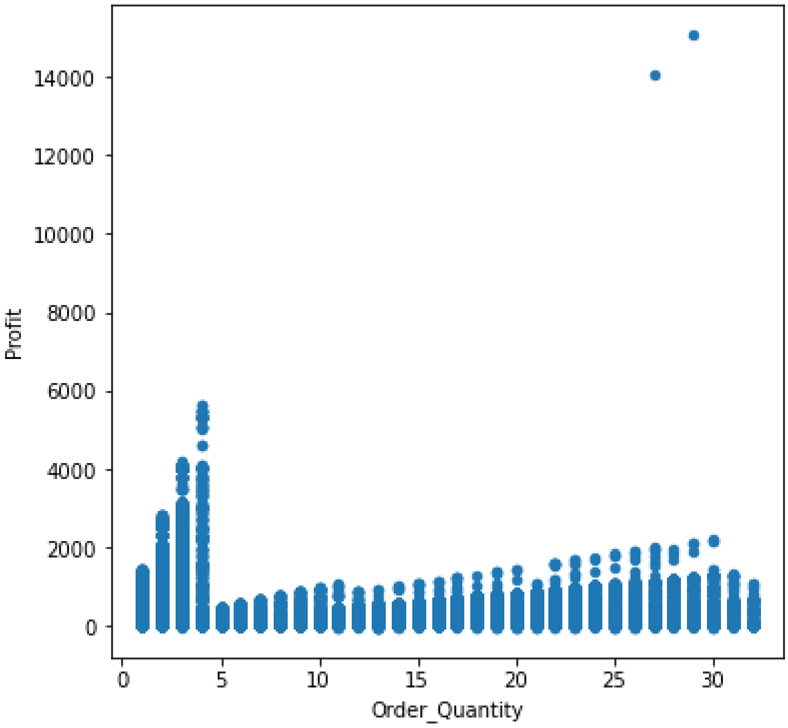


Relationship between Order_Quantity and Profit?

- Scatter plot between Order_Quautity and Profit.

In [31]:
```
sales.plot(kind='scatter', x='Order_Quantity', y='Profit', figsize=(6,6))
```

Out[31]:
```
<AxesSubplot:xlabel='Order_Quantity', ylabel='Profit'>
```



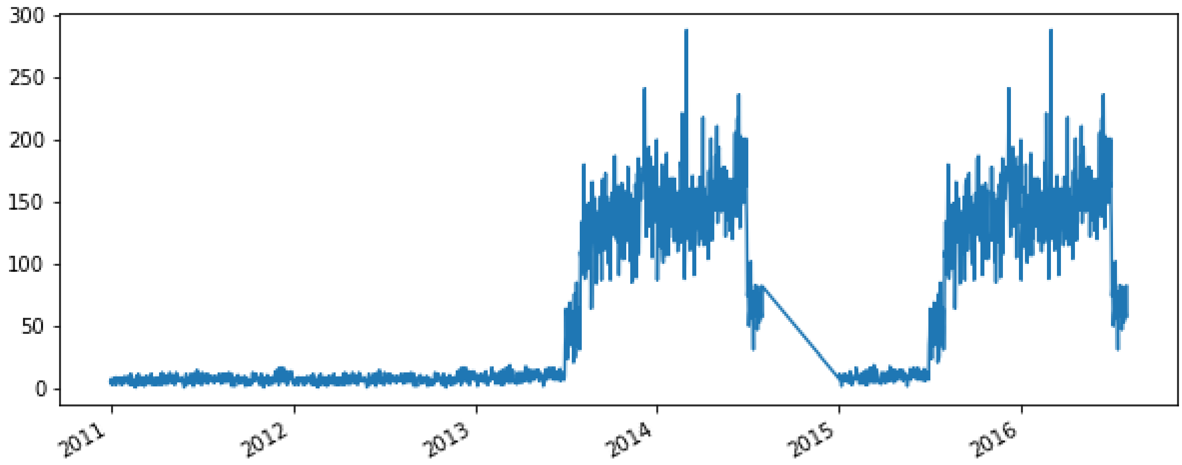
Sales evolve through the years?

- Line plot using Calculated_Date column as the x-axis and the count of sales as the y-axis.

In [40]:
```
sales['Calculated_Date'].value_counts().plot(kind='line', figsize=(10,4))
```

Out[40]:

```
<AxesSubplot:>
```



Orders were made in Canada and France

In [23]:

```
print("Order were made in Canada and France:")
sales.loc[(sales['Country'] == 'Canada') | (sales['Country'] == 'France')].shape[0]
```

```
Order were made in Canada and France:
```

Out[23]:

```
25176
```

Bike Racks orders were made from Canada

In [24]:

```
print ("Total number of Bike Racks orders were made from Canada :")
sales.loc[(sales['Country'] == 'Canada') & (sales['Sub_Category'] == 'Bike Racks')].
```

```
Total number of Bike Racks orders were made from Canada :
```

Out[24]:

```
104
```

Sales were made per category

In [27]:

```
print("Number of sales were made per categoty:")
sales['Product_Category'].value_counts()
```

```
Number of sales were made per categoty:
```

Out[27]:

```
Accessories    70120
Bikes          25982
Clothing       16934
Name: Product_Category, dtype: int64
```

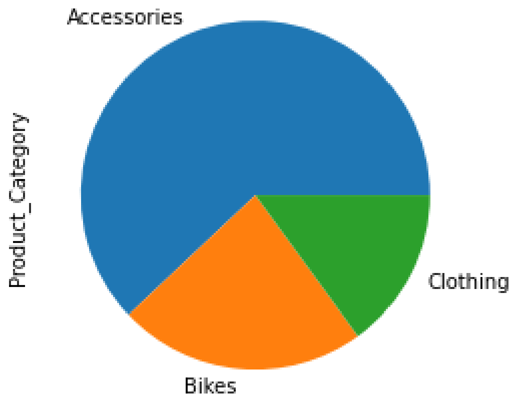
Pie plot with the results:

In [50]:

```
sales['Product_Category'].value_counts().plot(kind='pie', figsize=(4,4))
```

Out[50]:

```
<AxesSubplot:ylabel='Product_Category'>
```



Orders were made per accessory sub-categories

In [51]:

```
accessories = sales.loc[sales['Product_Category'] == 'Accessories', 'Sub_Category'].

accessories
```

Out[51]:

```
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
```

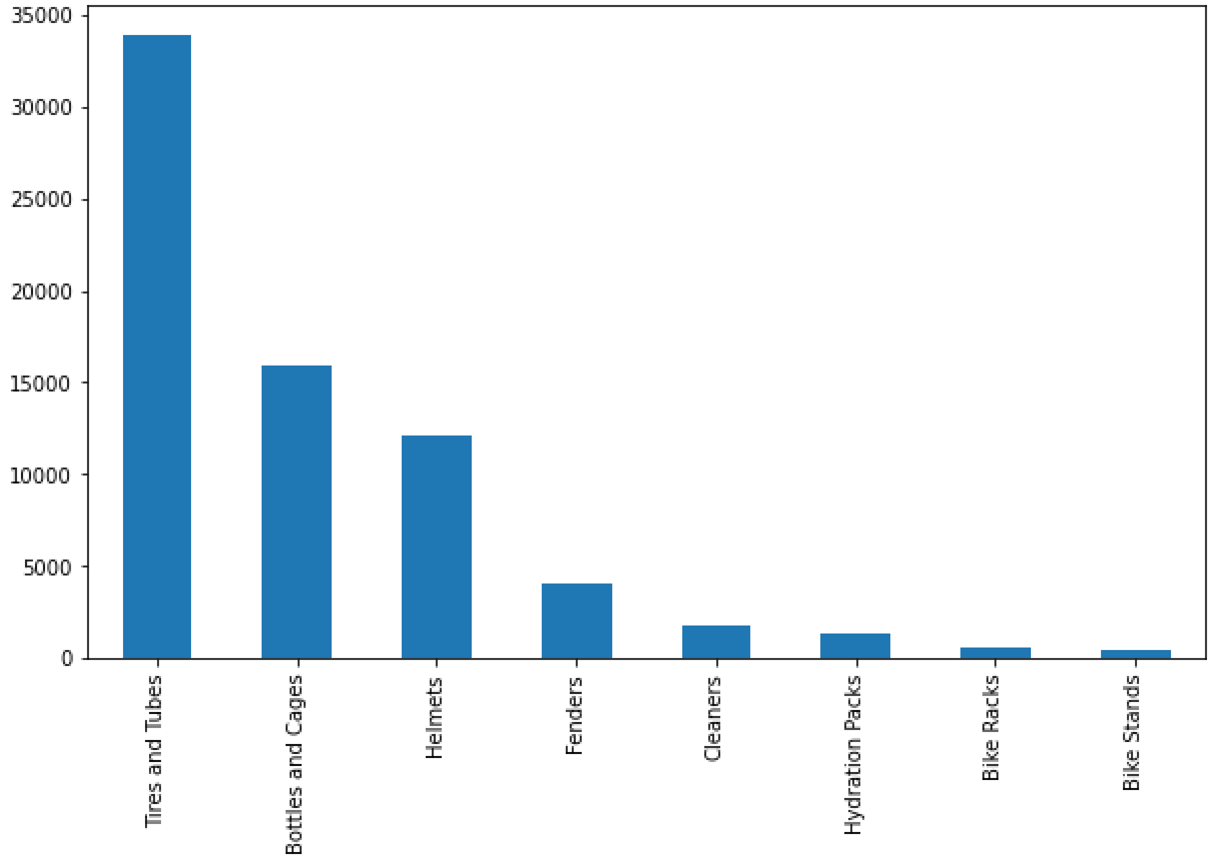
Bar plot with the results:

In [52]:

```
accessories.plot(kind='bar', figsize=(10,6))
```

Out[52]:

```
<AxesSubplot:>
```



Orders were made per bike sub-categories

In [28]:

```
print("Total number of orders were made per bike sub-categories:")
bikes = sales.loc[sales['Product_Category'] == 'Bikes', 'Sub_Category'].value_counts

bikes
```

Out[28]:

```
Total number of orders were made per bike sub-categories:
Road Bikes        13430
Mountain Bikes     8854
Touring Bikes      3698
Name: Sub_Category, dtype: int64
```

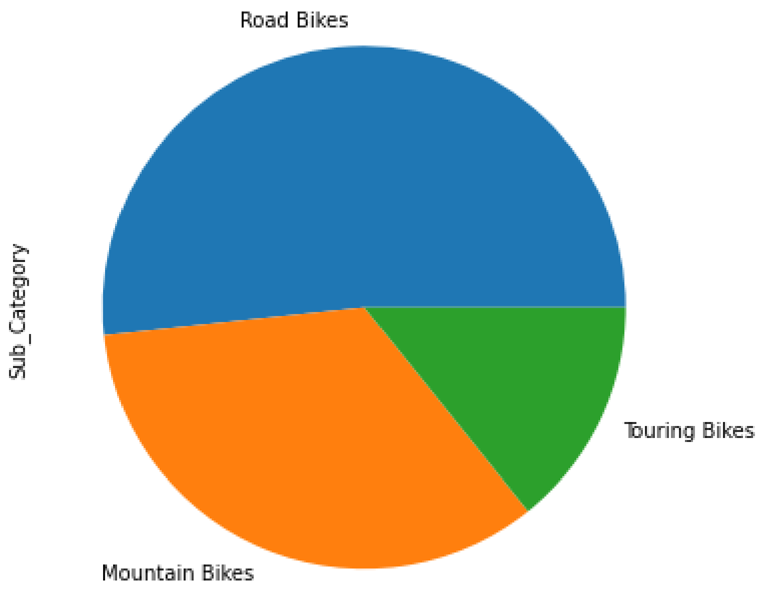
Pie plot with the results:

In [29]:

```
bikes.plot(kind='pie', figsize=(6,6))
```

Out[29]:

```
<AxesSubplot:ylabel='Sub_Category'>
```



Gender has the most amount of sales

In [34]:

```
sales['Customer_Gender'].value_counts()
```

Out[34]:

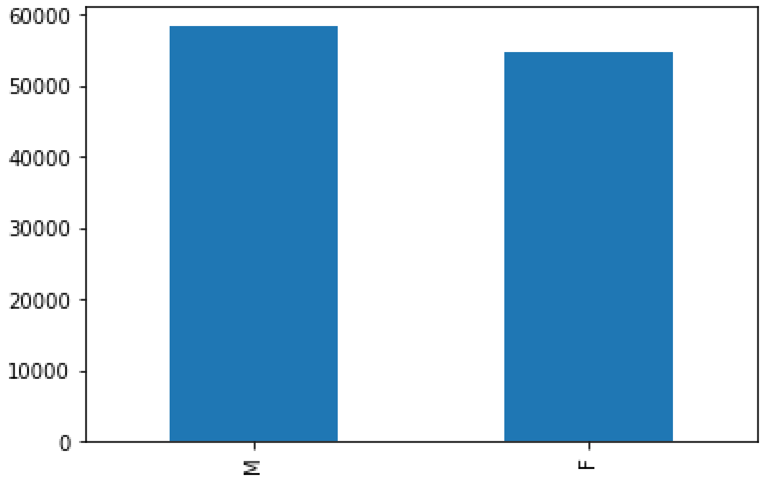
```
M    58312
F    54724
Name: Customer_Gender, dtype: int64
```

In [56]:

```
sales['Customer_Gender'].value_counts().plot(kind='bar')
```

Out[56]:

```
<AxesSubplot:>
```



Top-5 sales with the highest revenue

In [32]:

```
print("Top-5 sales:")
sales.sort_values(['Revenue'], ascending=False).head(5)
```

```
Top-5 sales:
```

Out[32]:

	Date	Day	Month	Year	Customer_Age	Age_Group	Customer_Gender	Country

	Date	Day	Month	Year	Customer_Age	Age_Group	Customer_Gender	Country	
112073	2015-07-24	24	July	2015	52	Adults (35-64)	M	Australia	Quee
112072	2013-07-24	24	July	2013	52	Adults (35-64)	M	Australia	Quee
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	Cc
70601	2011-09-30	30	September	2011	19	Youth (<25)	F	Canada	Cc

Sale with the highest revenue

In [30]:

```
print("Sale with the highest revenue:")
cond = sales['Revenue'] == sales['Revenue'].max()

sales.loc[cond]
```

```
Sale with the highest revenue:
```

Out[30]:

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

Orders were made in May of 2016

In [35]:

```
print("Total number of orders were made in May of 2016:")
cond = (sales['Year'] == 2016) & (sales['Month'] == 'May')

sales.loc[cond].shape[0]
```

```
Total number of orders were made in May of 2016:
```

Out[35]:

```
5015
```

In [ ]: