Exercises

Bike store sales

Hands on!

In [1]:

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

%matplotlib inline
```

In [2]:

```
sales = pd.read_csv(
    'sales_data.csv',
    parse_dates=['Date'])
```

In [3]:

```
sales.head()
```

Out[3]:

	Date	Day	Month	Year	Customer_Age	Age_Group	Customer_Gender	Country	State	Pr
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	

What's the mean of `Customers_Age`?

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

In [5]:

```
sales['Customer_Age'].mean()
```

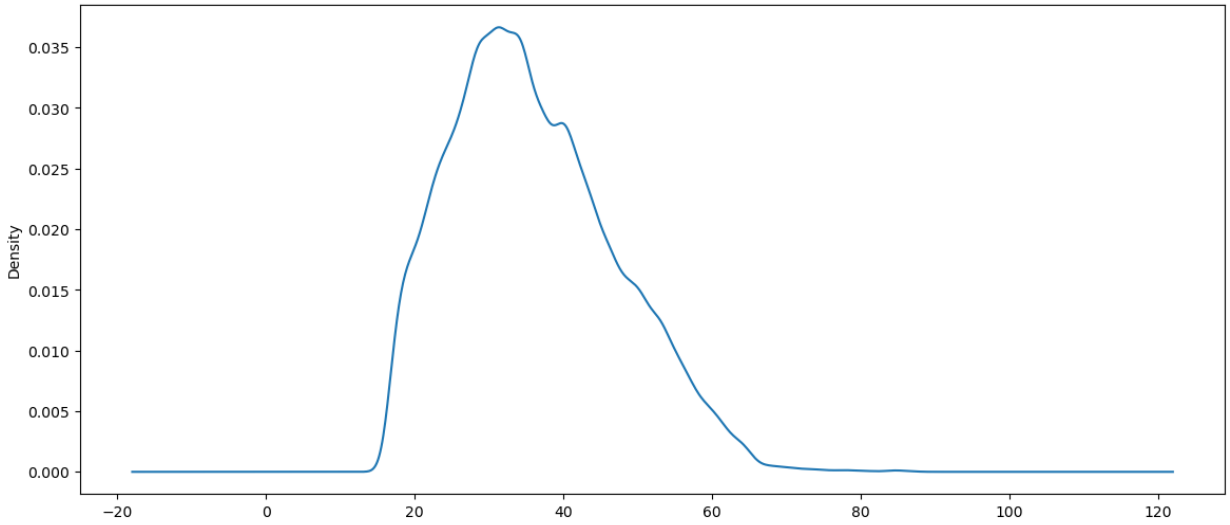
Out[5]:

```
35.92121157861212
```

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

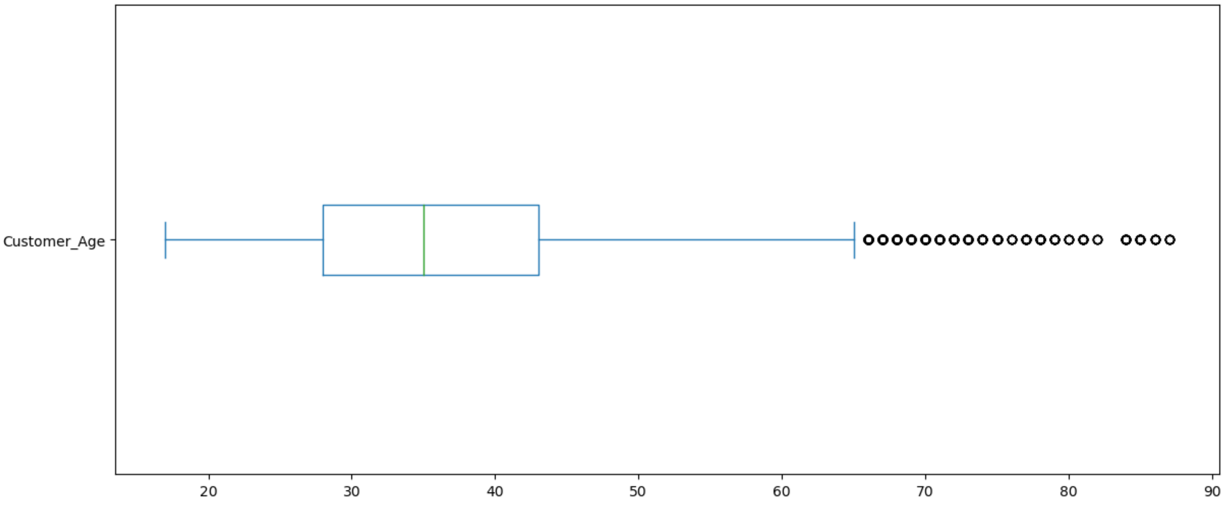
In [7]:
```
sales['Customer_Age'].plot(kind='kde', figsize=(14,6))
```

Out[7]:
```
<Axes: ylabel='Density'>
```



In [8]:
```
sales['Customer_Age'].plot(kind='box', vert=False, figsize=(14,6))
```

Out[8]:
```
<Axes: >
```



What's the mean of `Order_Quantity`?

In [10]:
```
sales['Order_Quantity'].mean()
```

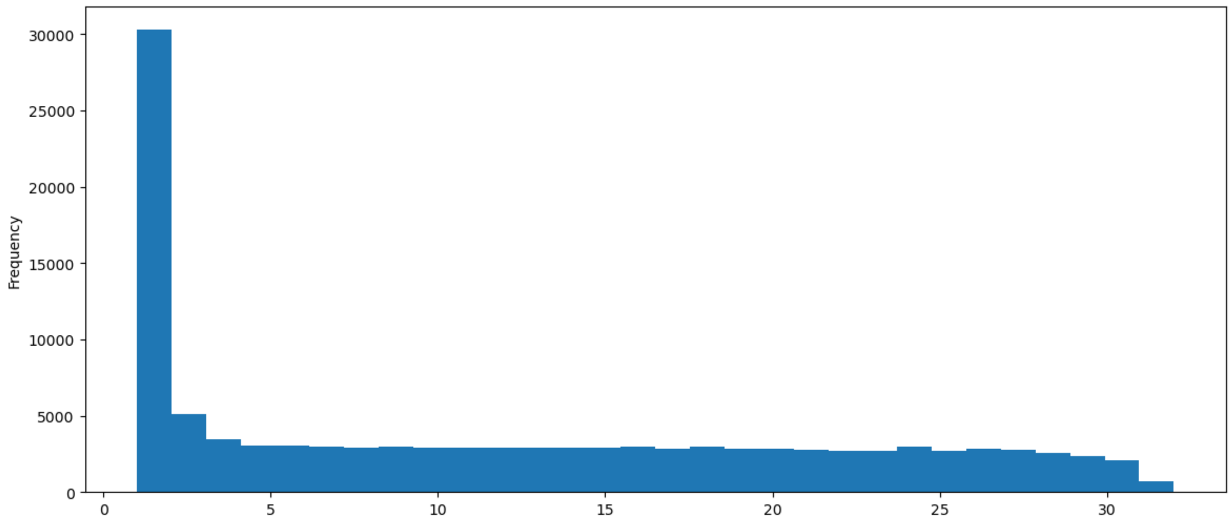
Out[10]:
```
11.901659648253654
```

Show a **histogram** and a **box plot** with the `Order_Quantity` data:

In [12]:
```
sales['Order_Quantity'].plot(kind='hist', bins=30, figsize=(14,6))
```

Out[12]:

```
<Axes: ylabel='Frequency'>
```

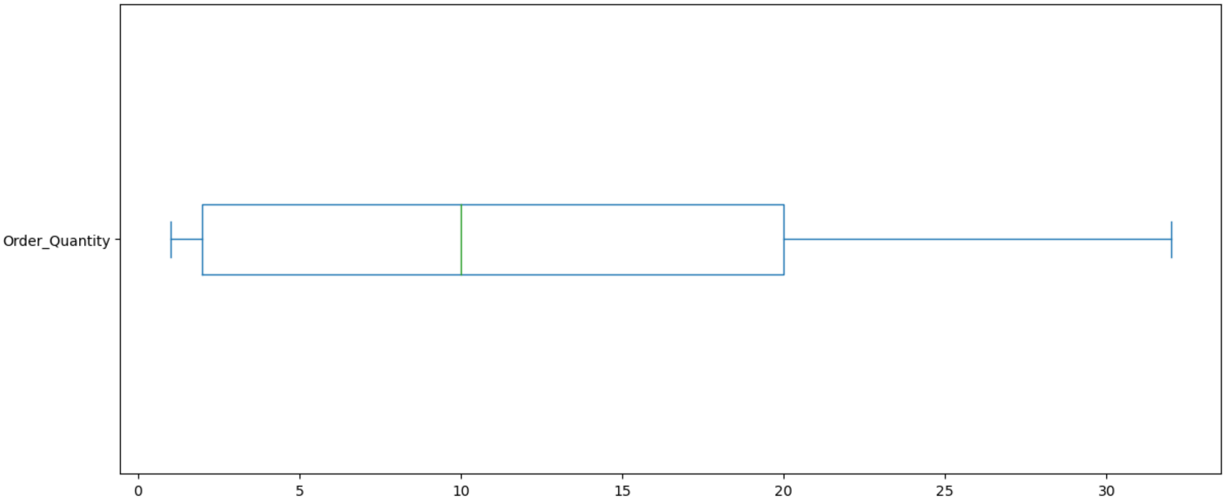


In [13]:

```
sales['Order_Quantity'].plot(kind='box', vert=False, figsize=(14,6))
```

Out[13]:

```
<Axes: >
```



How many sales per year do we have?

In [15]:

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

Out[15]:

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

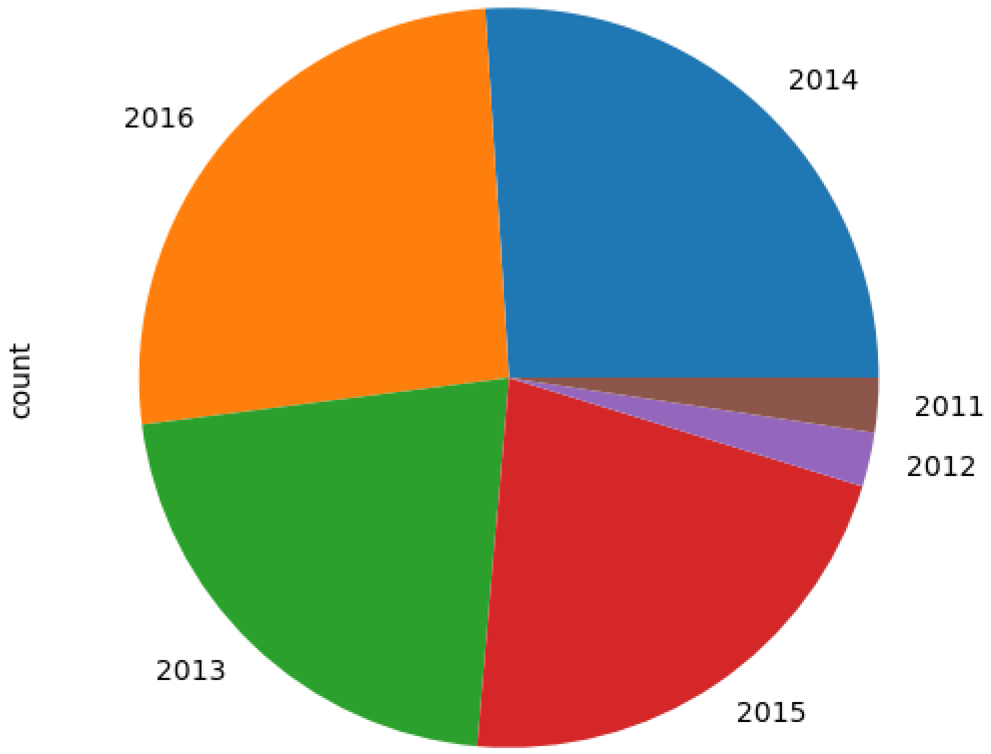
Show a **pie plot** with the previous data:

In [17]:

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

Out[17]:

```
<Axes: ylabel='count'>
```



How many sales per month do we have?

In [19]:

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

Out[19]:

```
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: count, dtype: int64
```

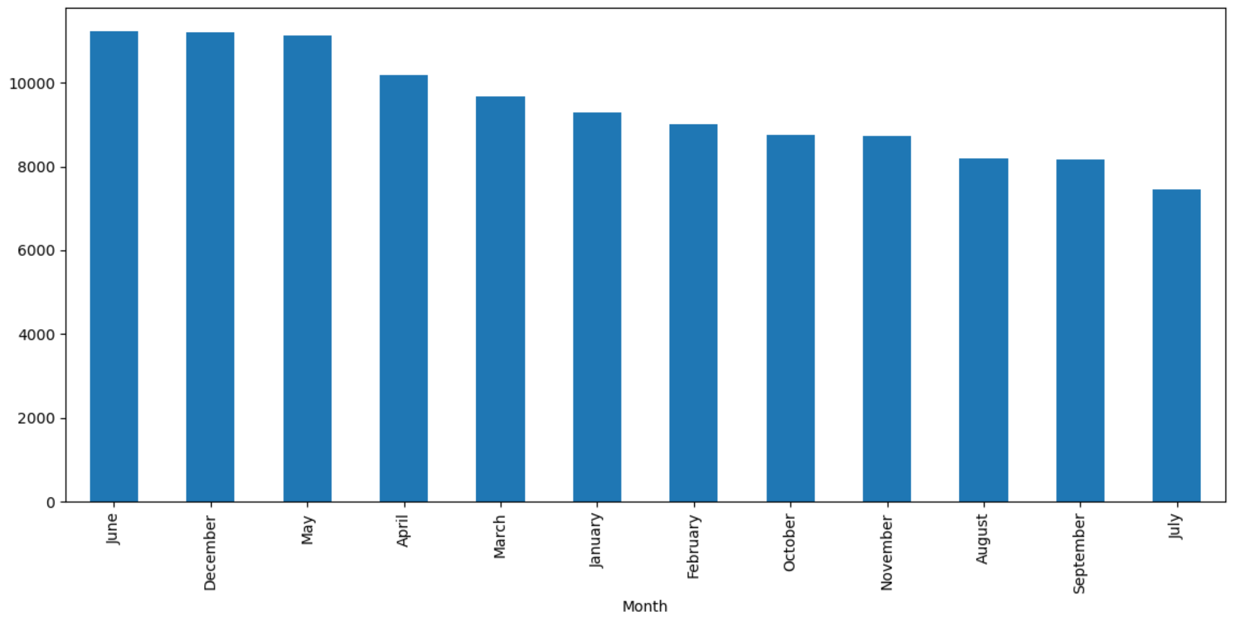
Show a **bar plot** with the previous data:

In [21]:

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

Out[21]:

```
<Axes: xlabel='Month'>
```



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

In [23]:

```
sales['Country'].value_counts().head(1)
```

Out[23]:

```
Country
United States    39206
Name: count, dtype: int64
```

In [24]:

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

Out[24]:

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

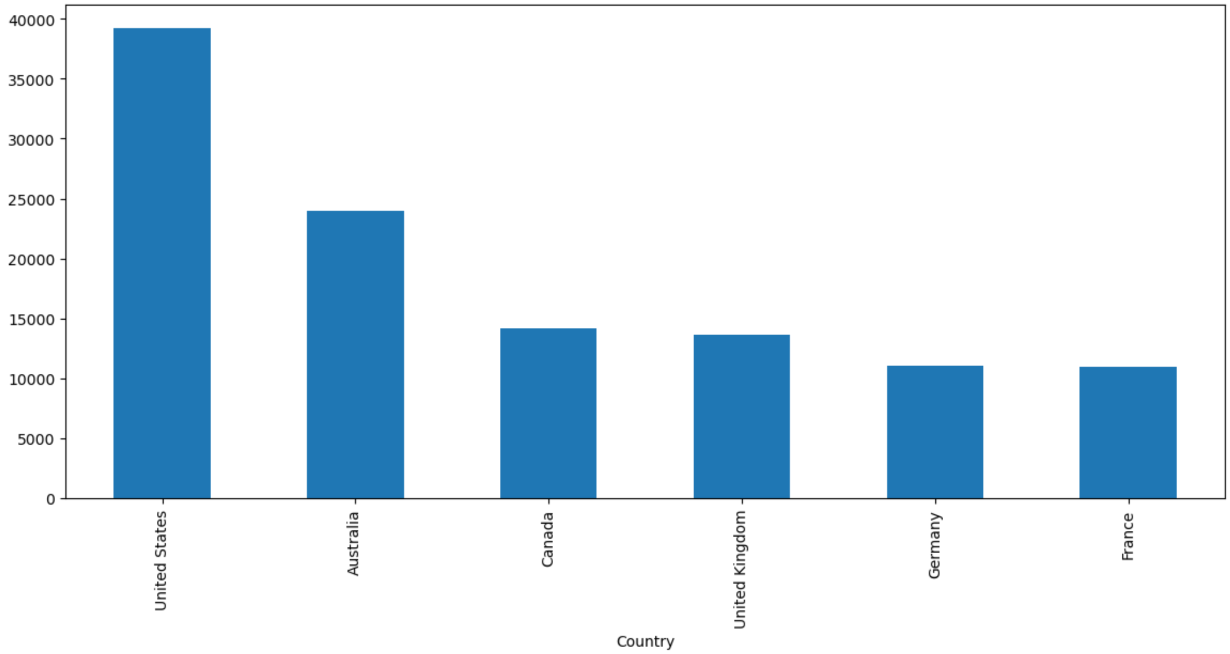
Show a **bar plot** of the sales per country:

In [26]:

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

Out[26]:

```
<Axes: xlabel='Country'>
```



Create a list of every product sold

In [28]:

```
#sales.loc[:, 'Product'].unique()

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

```
Out[28]: 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)
```

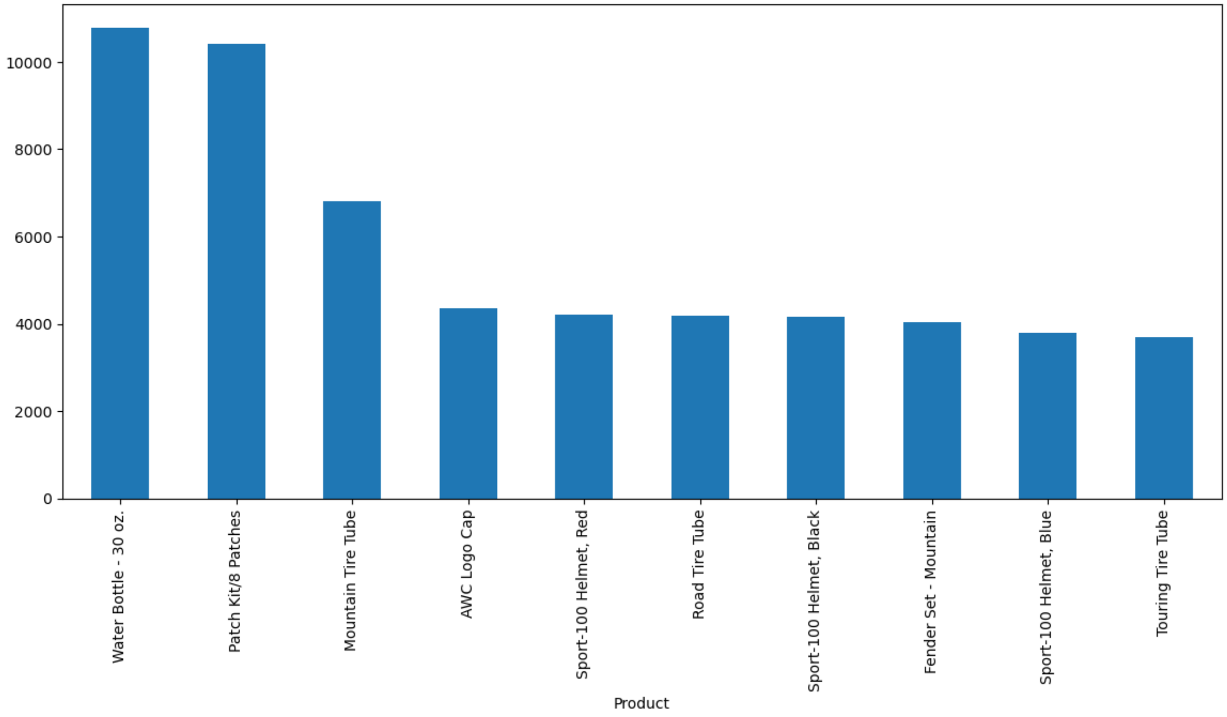
Create a **bar plot** showing the 10 most sold products (best sellers):

In [30]:

```
sales['Product'].value_counts().head(10).plot(kind='bar', figsize=(14,6))
```

Out[30]:

```
<Axes: xlabel='Product'>
```



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

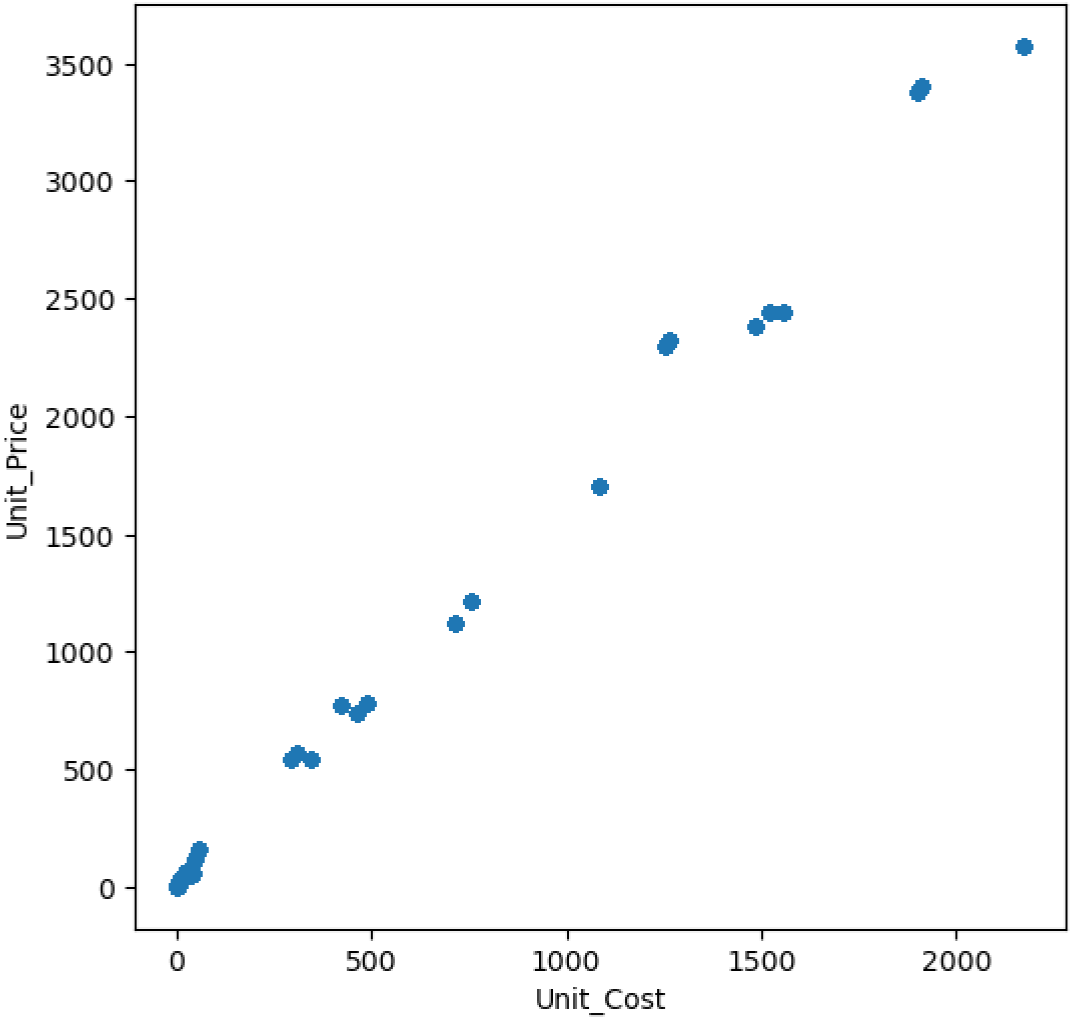
Show a **scatter plot** between both columns.

In [32]:

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

Out[32]:

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

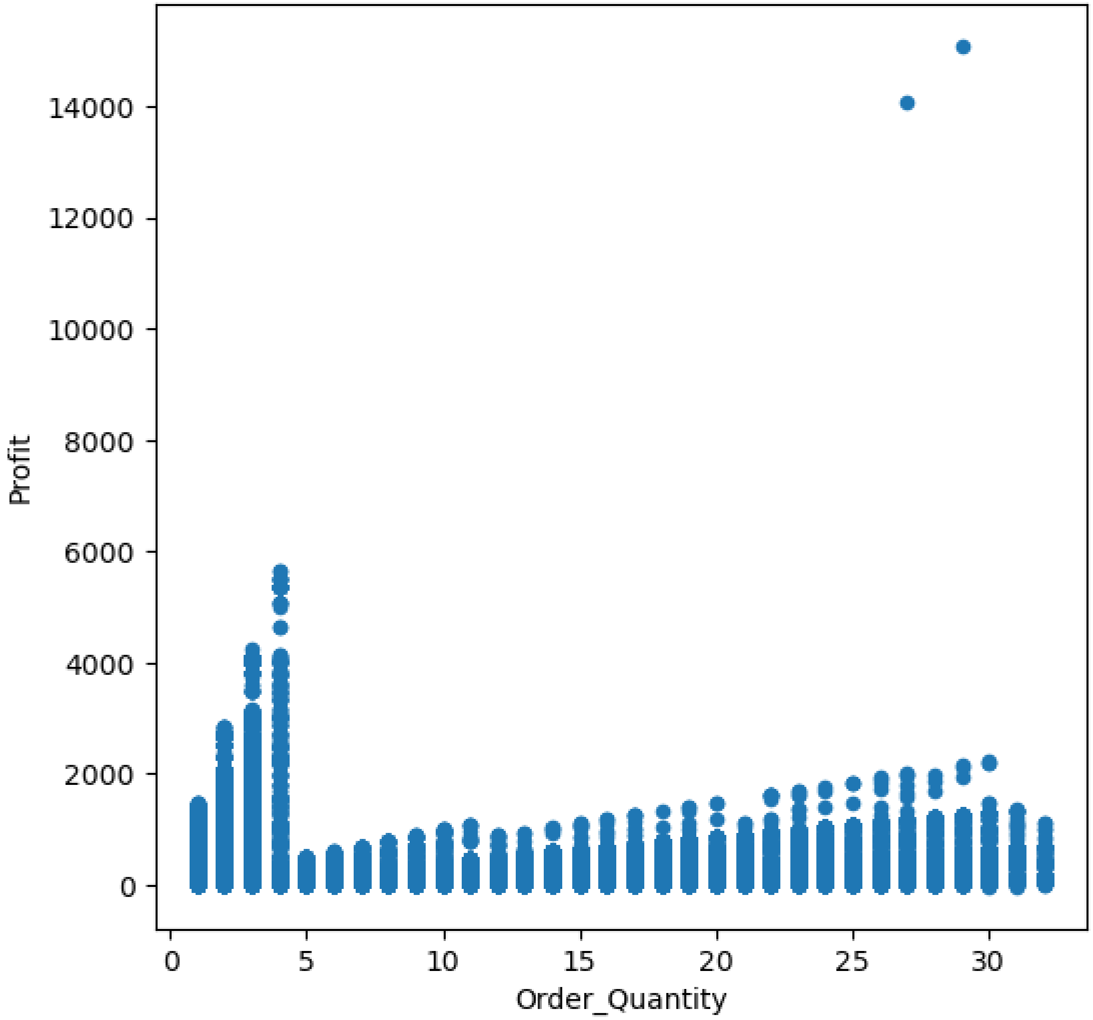


Can you see any relationship between `Order_Quantity` and `Profit`?

Show a **scatter plot** between both columns.

In [34]:
```
sales.plot(kind='scatter', x='Order_Quantity', y='Profit', figsize=(6,6))
```

Out[34]:
```
<Axes: xlabel='Order_Quantity', ylabel='Profit'>
```

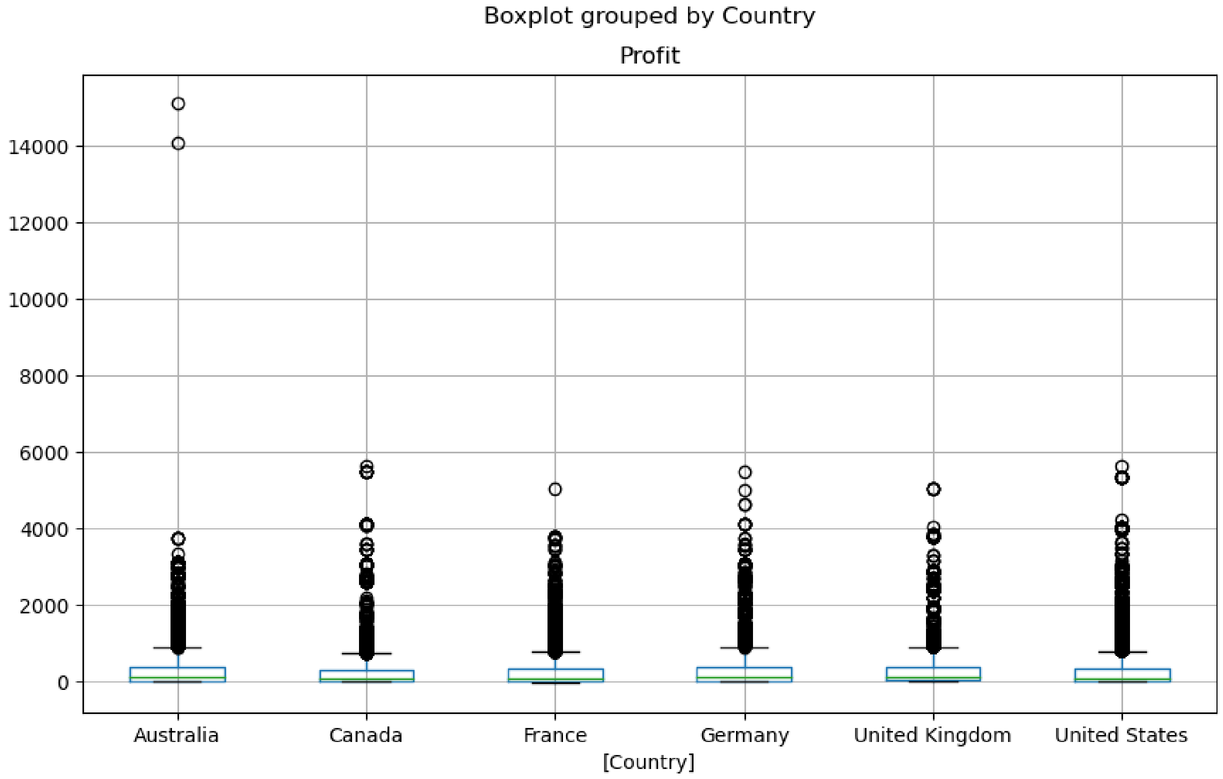


Can you see any relationship between `Profit` per `Country`?

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

```
In [36]: sales[['Profit', 'Country']].boxplot(by='Country', figsize=(10,6))
```

```
Out[36]: <Axes: title={'center': 'Profit'}, xlabel='[Country]'>
```

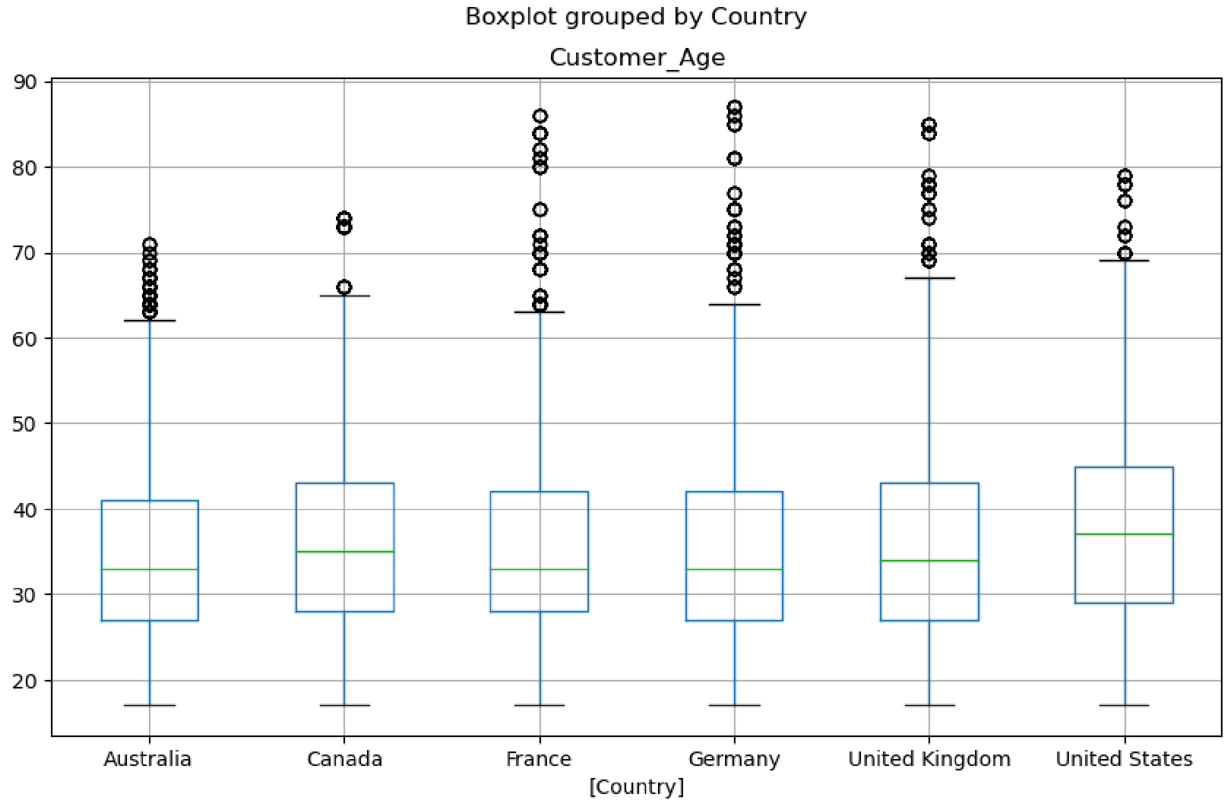


Can you see any relationship between the `Customer_Age` per `Country`?

Show a grouped **box plot** per country with the customer age values.

```
In [38]: sales[['Customer_Age', 'Country']].boxplot(by='Country', figsize=(10,6))
```

```
Out[38]: <Axes: title={'center': 'Customer_Age'}, xlabel='[Country]'>
```



Add and calculate a new `Calculated_Date` column

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

In [40]:
```
sales['Calculated_Date'] = sales[['Year', 'Month', 'Day']].apply(lambda x: '{}-{}-{}'.

sales['Calculated_Date'].head()
```

Out[40]:
```
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 [42]:
```
sales['Calculated_Date'] = pd.to_datetime(sales['Calculated_Date'])

sales['Calculated_Date'].head()
```

Out[42]:

```
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?

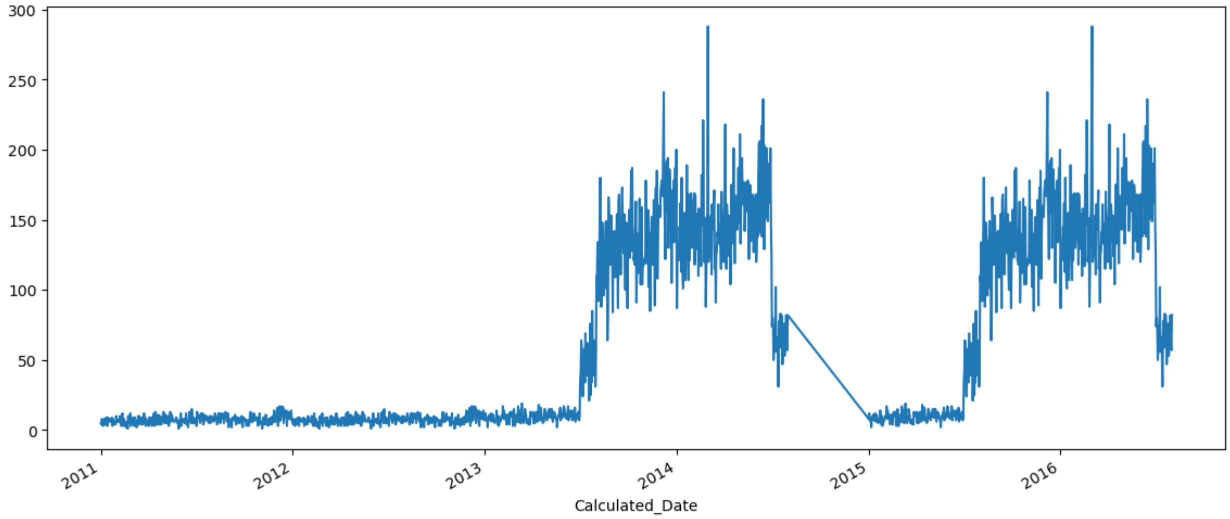
Show a **line plot** using `Calculated_Date` column as the x-axis and the count of sales as the y-axis.

In [44]:

```
sales['Calculated_Date'].value_counts().plot(kind='line', figsize=(14,6))
```

Out[44]:

```
<Axes: xlabel='Calculated_Date'>
```



Increase 50 U$S revenue to every sale

In [46]:

```
#sales['Revenue'] = sales['Revenue'] + 50

sales['Revenue'] += 50
```

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

In [48]:

```
sales.loc[(sales['Country'] == 'Canada') | (sales['Country'] == 'France')].shape[0]
```

Out[48]:

```
25176
```

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

In [50]:
```
sales.loc[(sales['Country'] == 'Canada') & (sales['Sub_Category'] == 'Bike Racks')].sh
```

Out[50]:
```
104
```

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

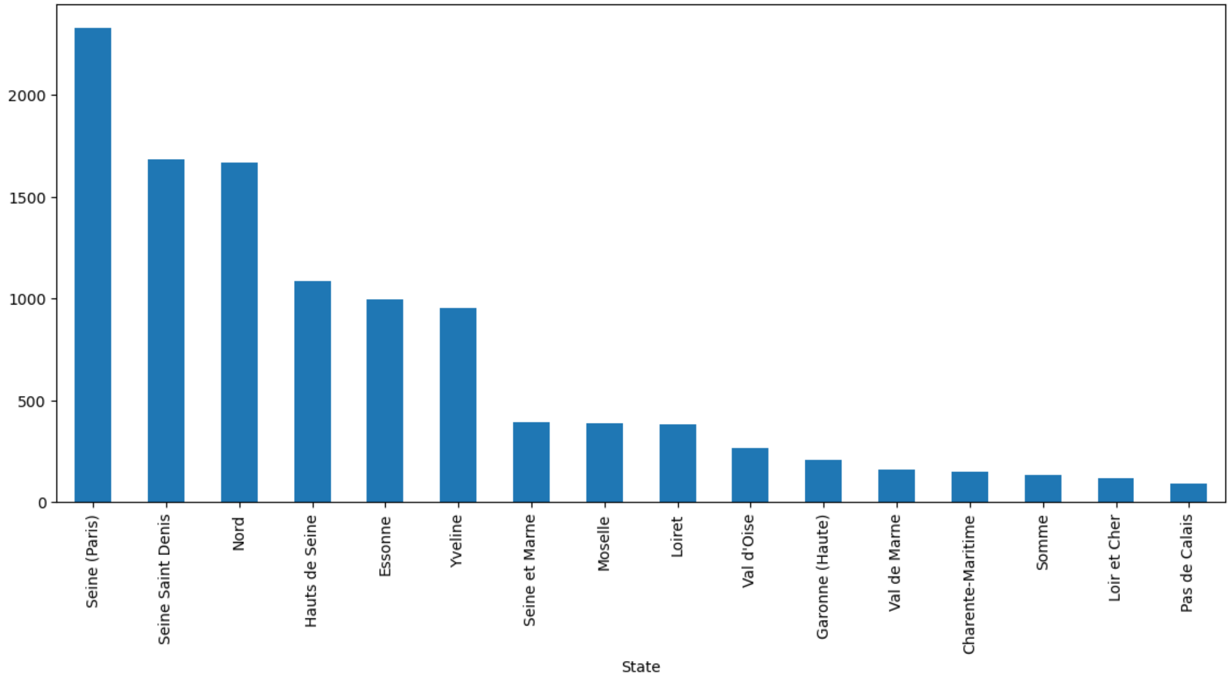
In [52]:
```
france_states = sales.loc[sales['Country'] == 'France', 'State'].value_counts()

france_states
```

Out[52]:
```
State
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: count, dtype: int64
```

Show a **bar plot** with the results:

In [54]:
```
france_states.plot(kind='bar', figsize=(14,6))
```

Out[54]:
```
<Axes: xlabel='State'>
```



How many sales were made per category?

In [56]:

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

Out[56]:

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

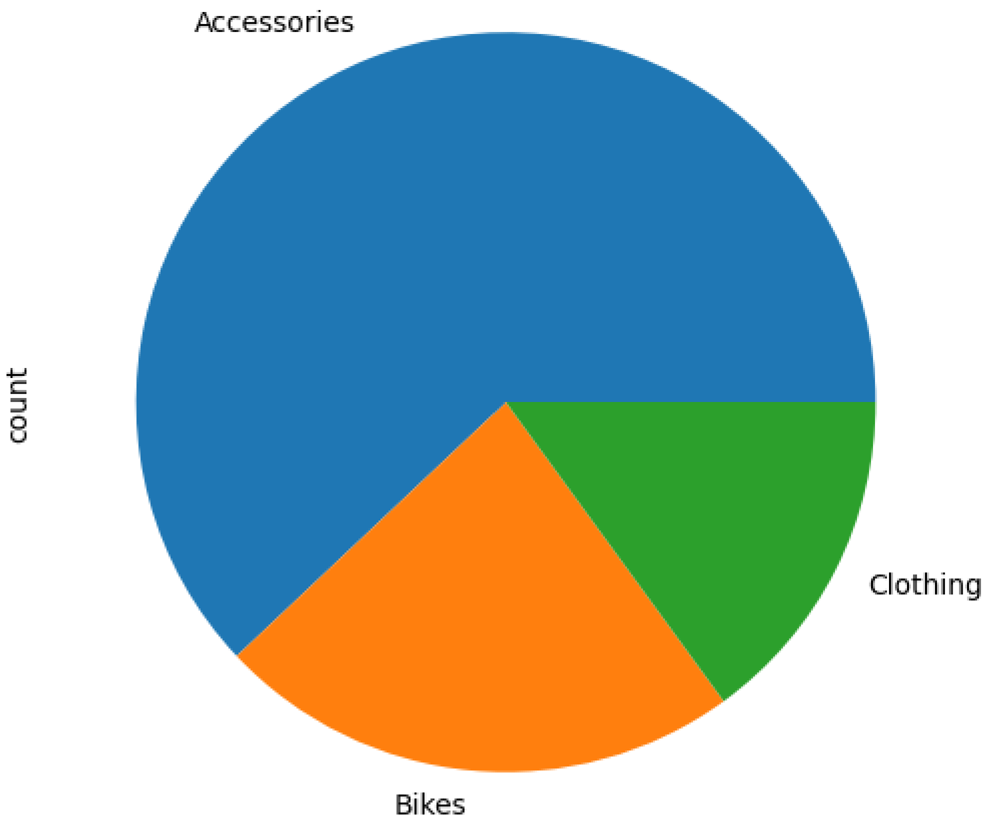
To show a **pie plot** with the results:

In [58]:

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

Out[58]:

```
<Axes: ylabel='count'>
```



How many orders were made per accessory sub-categories?

In [60]:

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

accessories
```

Out[60]:

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

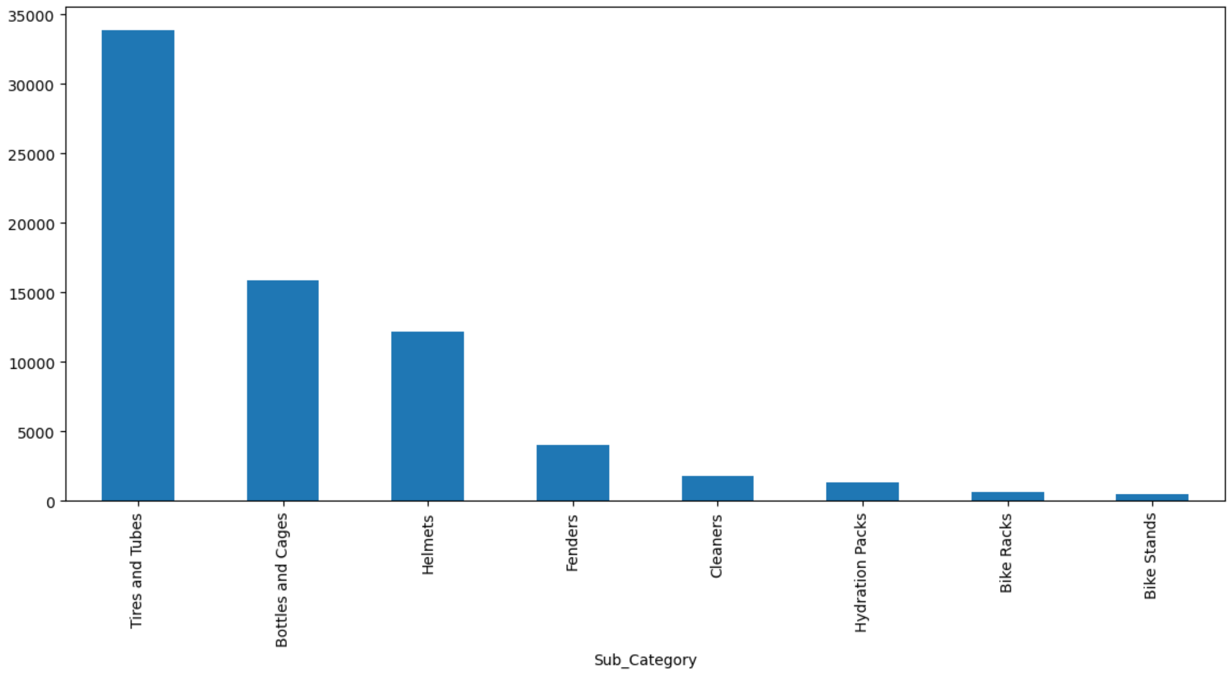
To show a **bar plot** with the results:

In [62]:

```
accessories.plot(kind='bar', figsize=(14,6))
```

Out[62]:

```
<Axes: xlabel='Sub_Category'>
```



How many orders were made per bike sub-categories?

In [64]:

```
bikes = sales.loc[sales['Product_Category'] == 'Bikes', 'Sub_Category'].value_counts()

bikes
```

Out[64]:

```
Sub_Category
Road Bikes         13430
Mountain Bikes      8854
Touring Bikes       3698
Name: count, dtype: int64
```

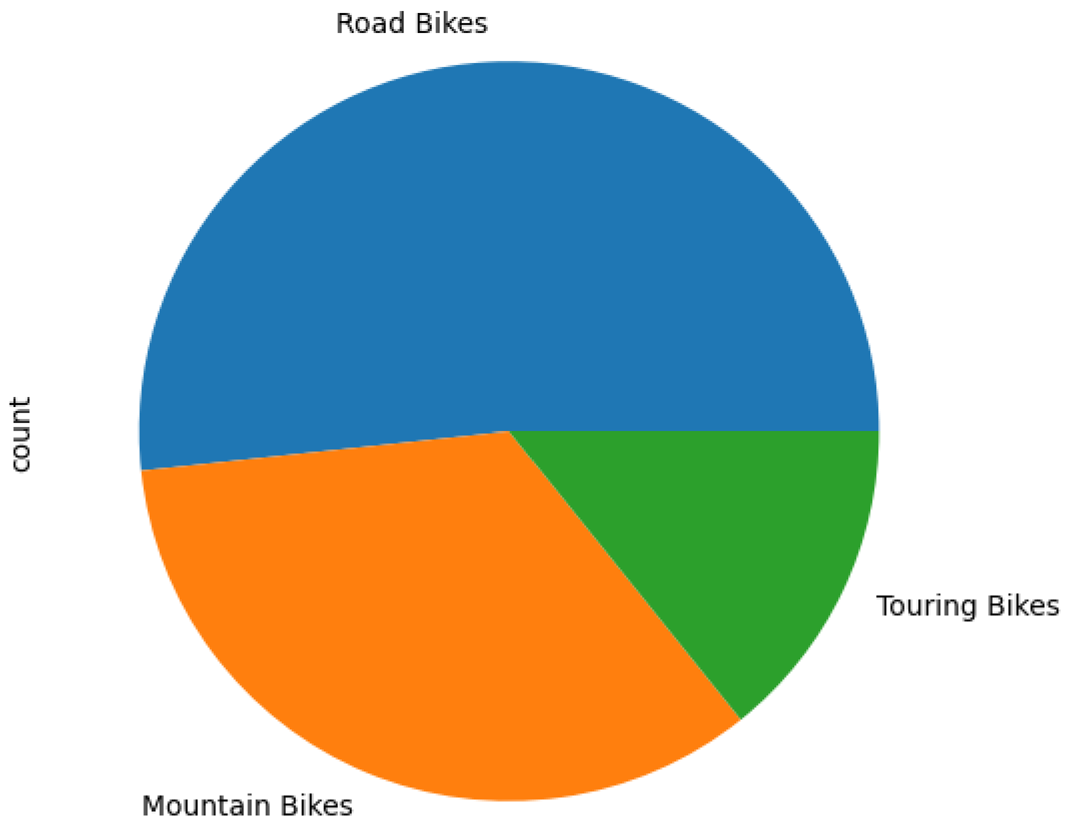
To show a **pie plot** with the results:

In [66]:

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

Out[66]:

```
<Axes: ylabel='count'>
```



Which gender has the most amount of sales?

In [68]:

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

Out[68]:

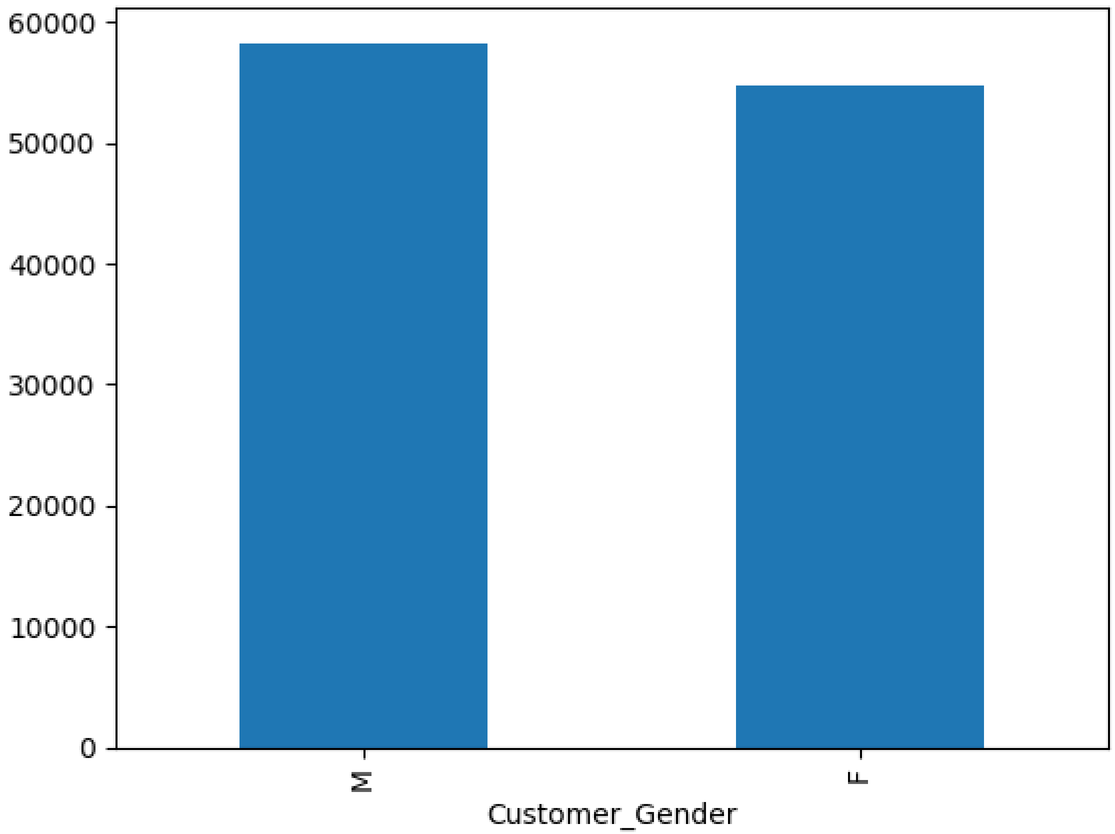
```
Customer_Gender
M    58312
F    54724
Name: count, dtype: int64
```

In [69]:

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

Out[69]:

```
<Axes: xlabel='Customer_Gender'>
```



How many sales with more than 500 in `Revenue` were made by men?

```
In [71]: sales.loc[(sales['Customer_Gender'] == 'M') & (sales['Revenue'] == 500)].shape[0]
```

```
Out[71]: 50
```

Get the top-5 sales with the highest revenue

```
In [73]: sales.sort_values(['Revenue'], ascending=False).head(5)
```

Out[73]:

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

Get the sale with the highest revenue

In [75]:

```
#sales.sort_values(['Revenue'], ascending=False).head(1)

cond = sales['Revenue'] == sales['Revenue'].max()

sales.loc[cond]
```

Out[75]:

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

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

In [77]:

```
cond = sales['Revenue'] > 10_000

sales.loc[cond, 'Order_Quantity'].mean()
```

Out[77]:

```
3.689265536723164
```

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

In [79]:

```
cond = sales['Revenue'] < 10_000

sales.loc[cond, 'Order_Quantity'].mean()
```

Out[79]:

```
11.914539380997528
```

How many orders were made in May of 2016?

In [81]:

```
cond = (sales['Year'] == 2016) & (sales['Month'] == 'May')

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

Out[81]:

```
5015
```

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

In [83]:

```
cond = (sales['Year'] == 2016) & (sales['Month'].isin(['May', 'June', 'July']))

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

Out[83]:

```
12164
```

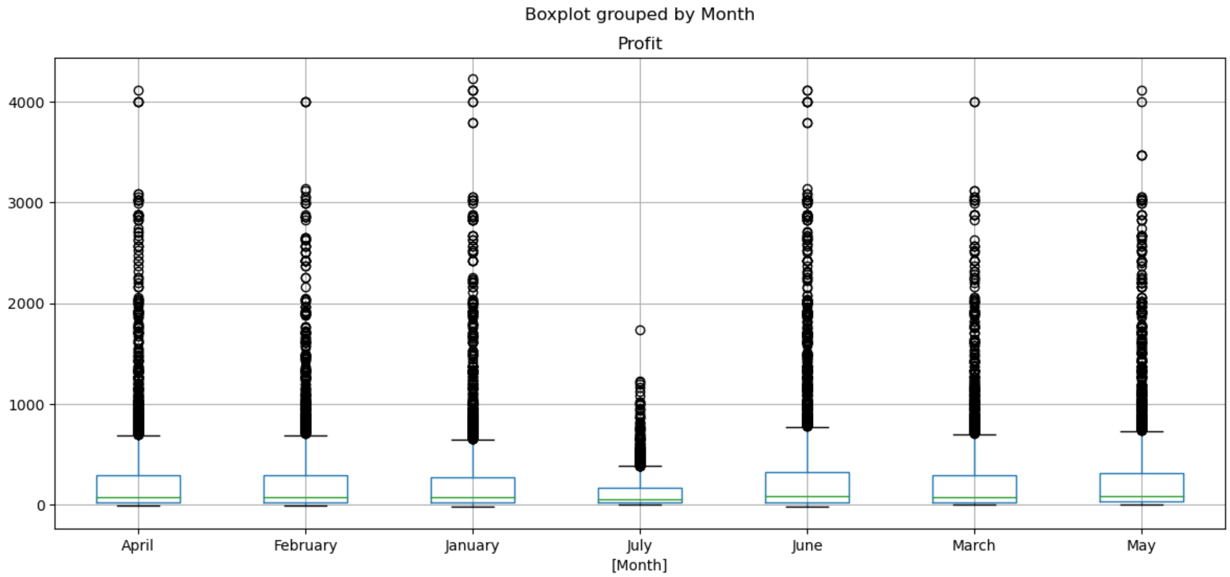
Show a grouped **box plot** per month with the profit values.

In [85]:

```
profit_2016 = sales.loc[sales['Year'] == 2016, ['Profit', 'Month']]

profit_2016.boxplot(by='Month', figsize=(14,6))
```

Out[85]:

```
<Axes: title={'center': 'Profit'}, xlabel='[Month]'>
```



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

In [88]:

```
#sales.loc[sales['Country'] == 'United States', 'Unit_Price'] = sales.loc[sales['Count

sales.loc[sales['Country'] == 'United States', 'Unit_Price'] *= 1.072
```