Sales Dataset ¶

In this notebook, I will use sales data from an international store that specializes in bikes and bike accessories. First, I will explore the data Second, check if it needs cleaning Then, will answer various questions using filters and search.

In [ ]:

In [194]:
```
import pandas as pd
```

In [60]:
```
import matplotlib.pyplot as plt
```

In [2]:
```
pwd
```

Out[2]:
```
'C:\\Users\\maria\\Documents\\python for github\\FreeCode'
```

In [8]:
```
sales = pd.read_csv('sales_data.csv')
```

In [23]:
```
sales.shape
```

Out[23]:
```
(113036, 18)
```

In [9]:
```
sales.head()
```

Out[9]:

	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

In [11]:
```
sales.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 113036 entries, 0 to 113035
Data columns (total 18 columns):
Date                113036 non-null object
Day                 113036 non-null int64
Month               113036 non-null object
Year                113036 non-null int64
Customer_Age        113036 non-null int64
Age_Group           113036 non-null object
Customer_Gender     113036 non-null object
Country             113036 non-null object
State               113036 non-null object
Product_Category    113036 non-null object
Sub_Category        113036 non-null object
Product             113036 non-null object
Order_Quantity      113036 non-null int64
Unit_Cost           113036 non-null int64
Unit_Price          113036 non-null int64
Profit              113036 non-null int64
Cost                113036 non-null int64
Revenue             113036 non-null int64
dtypes: int64(9), object(9)
memory usage: 15.5+ MB
```

In [13]:
```
sales.describe()
```

Out[13]:

	Day	Year	Customer_Age	Order_Quantity	Unit_Cost	Unit_Pric
count	113036.000000	113036.000000	113036.000000	113036.000000	113036.000000	113036.00000
mean	15.665753	2014.401739	35.919212	11.901660	267.296366	452.93842
std	8.781567	1.272510	11.021936	9.561857	549.835483	922.07121
min	1.000000	2011.000000	17.000000	1.000000	1.000000	2.00000
25%	8.000000	2013.000000	28.000000	2.000000	2.000000	5.00000
50%	16.000000	2014.000000	35.000000	10.000000	9.000000	24.00000
75%	23.000000	2016.000000	43.000000	20.000000	42.000000	70.00000
max	31.000000	2016.000000	87.000000	32.000000	2171.000000	3578.00000

In [195]:
```
product_types=sales['Product'].value_counts()
```

In [16]:
```
product_types.shape
```

Out[16]:
```
(130,)
```

I'm interested in seeing what are the specific products that this store sells:

In [17]:
```
sales['Sub_Category'].value_counts()
```

Out[17]:
```
Tires and Tubes      33870
Bottles and Cages    15876
Road Bikes           13430
Helmets              12158
Mountain Bikes        8854
Jerseys               6010
Caps                  4358
Fenders               4032
Touring Bikes         3698
Gloves                2686
Cleaners              1802
Shorts                1794
Hydration Packs       1334
Socks                 1122
Vests                  964
Bike Racks             592
Bike Stands            456
Name: Sub_Category, dtype: int64
```

In [18]:
```
sales['Sub_Category'].value_counts().shape
```

Out[18]:
```
(17,)
```

In [19]:
```
sales['Product_Category'].value_counts()
```

Out[19]:
```
Accessories    70120
Bikes          25982
Clothing       16934
Name: Product_Category, dtype: int64
```

In [20]:
```
sales.isnull().any() #checks each column
```

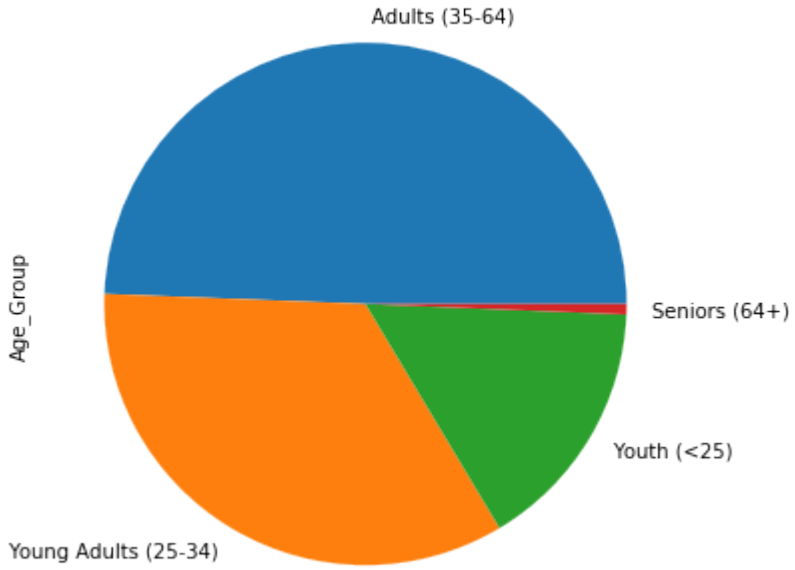
Out[20]:
```
Date                False
Day                 False
Month               False
Year                False
Customer_Age        False
Age_Group           False
Customer_Gender     False
Country             False
State               False
Product_Category    False
Sub_Category        False
Product             False
Order_Quantity      False
Unit_Cost           False
Unit_Price          False
Profit              False
Cost                False
Revenue             False
dtype: bool
```

In [ ]:
```
#if there were null values, we could use .dropna() if we decide to drop them
#but there are other things we can do with null values
```

In [22]:
```
#sales.isnull()  gives for each entry
```

In [25]:
```
sales['Age_Group'].value_counts().plot(kind='pie', figsize=(6,6))
```

Out[25]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1d8bccec888>
```



In [27]:
```
sales['Age_Group'].value_counts()
```

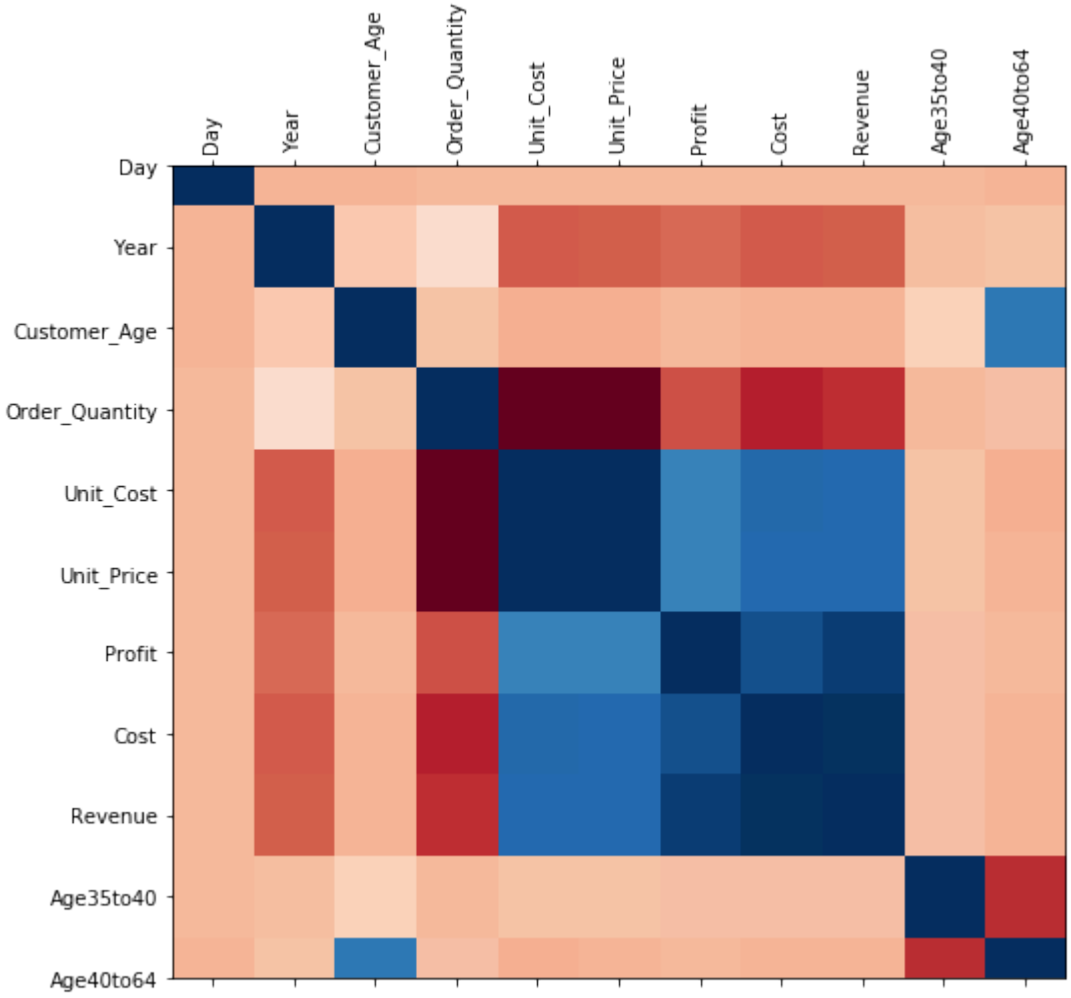
Out[27]:
```
Adults (35-64)          55824
Young Adults (25-34)    38654
Youth (<25)             17828
Seniors (64+)             730
Name: Age_Group, dtype: int64
```

In [56]:
```
corr = sales.corr()
```

In [57]:
```
fig = plt.figure(figsize=(8,8))
plt.matshow(corr, cmap='RdBu', fignum=fig.number)
plt.xticks(range(len(corr.columns)), corr.columns, rotation='vertical');
plt.yticks(range(len(corr.columns)), corr.columns);
```

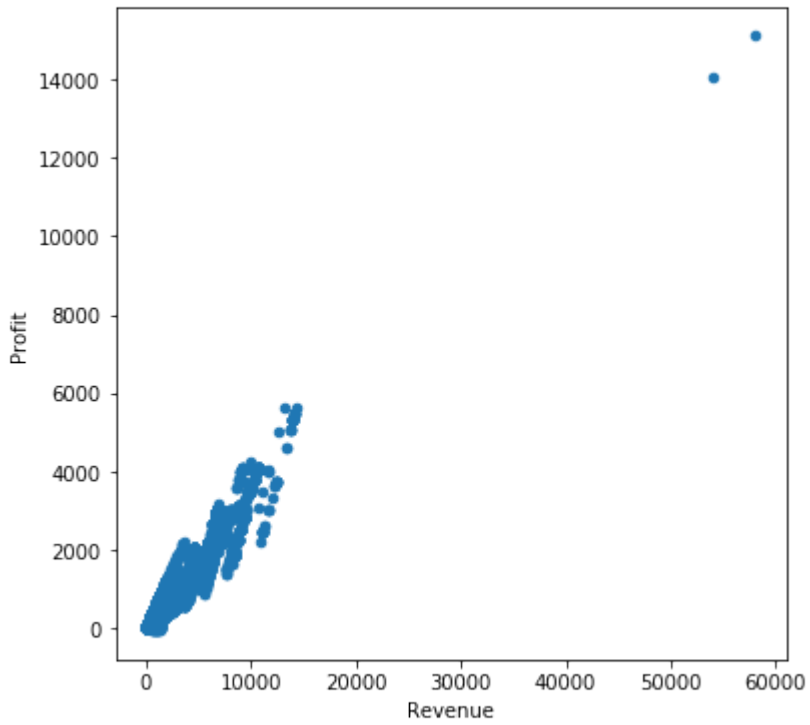
```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-57-ccb5beba7442> in <module>
----> 1 fig = plt.figure(figsize=(8,8))
      2 plt.matshow(corr, cmap='RdBu', fignum=fig.number)
      3 plt.xticks(range(len(corr.columns)), corr.columns, rotation='vertica
l');
      4 plt.yticks(range(len(corr.columns)), corr.columns);

NameError: name 'plt' is not defined
```

In [61]:
```
fig = plt.figure(figsize=(8,8))
plt.matshow(corr, cmap='RdBu', fignum=fig.number)
plt.xticks(range(len(corr.columns)), corr.columns, rotation='vertical');
plt.yticks(range(len(corr.columns)), corr.columns);
```



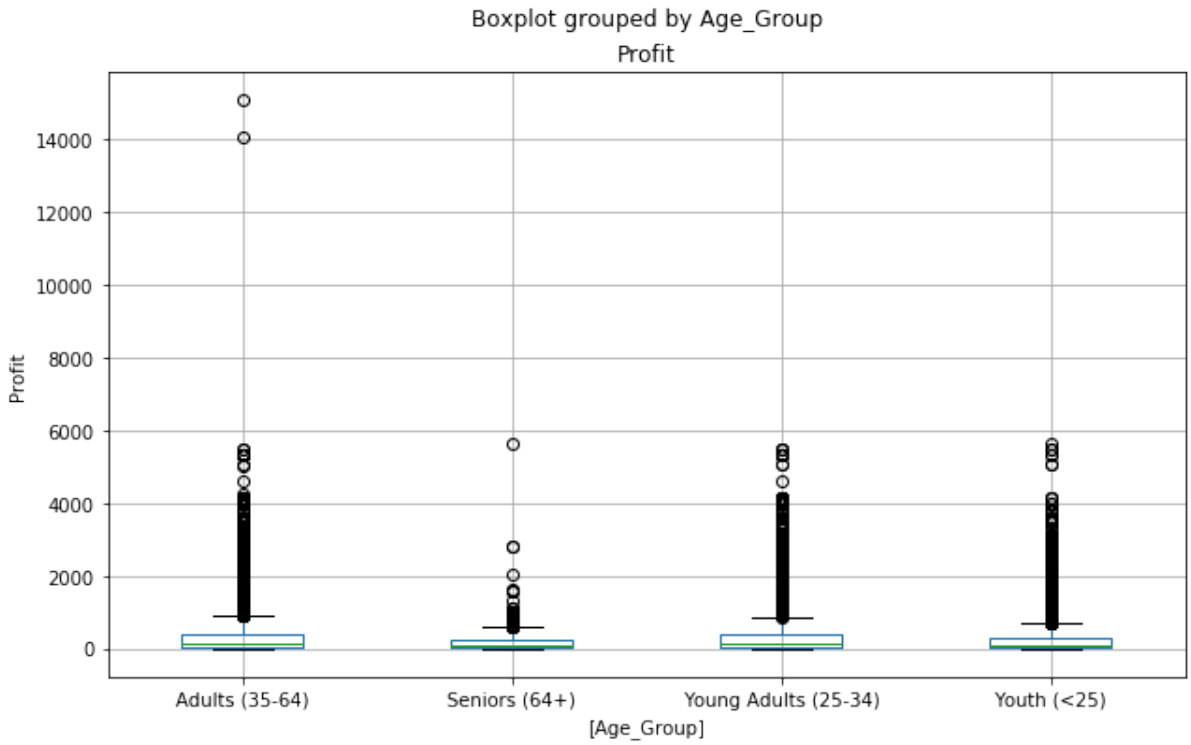
In [63]:
```
sales.plot(kind='scatter', x='Revenue', y='Profit', figsize=(6,6))
```

Out[63]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1d8bd201948>
```



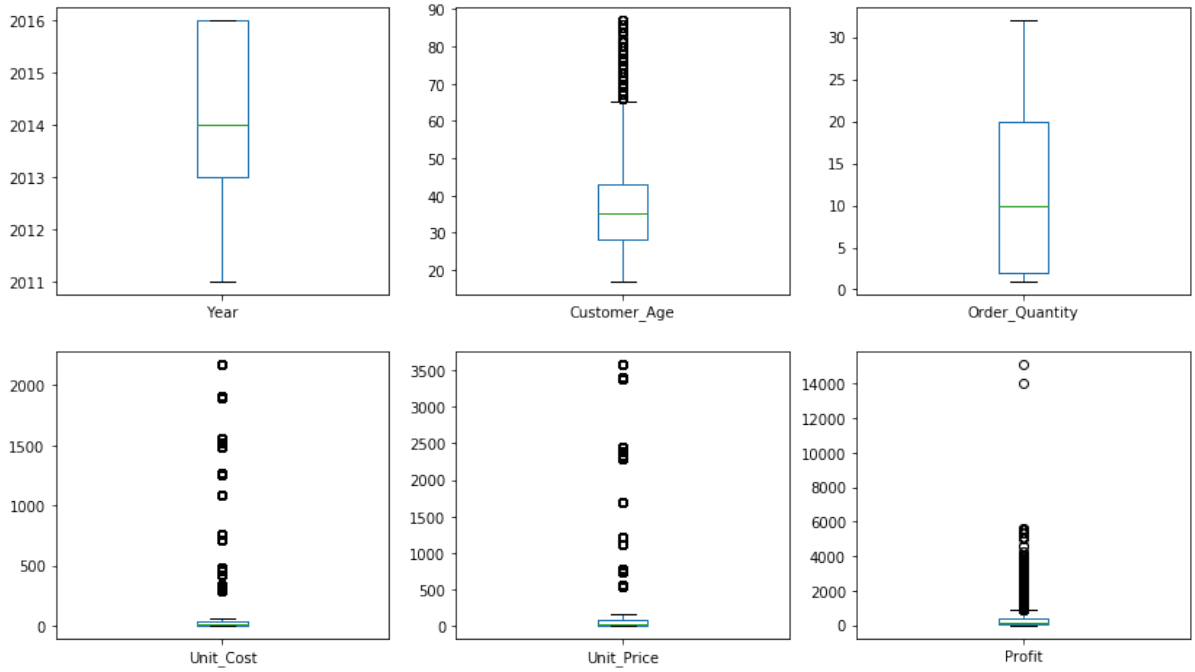
In [65]:
```
ax = sales[['Profit', 'Age_Group']].boxplot(by='Age_Group', figsize=(10,6))
ax.set_ylabel('Profit')
```

Out[65]:
```
Text(0, 0.5, 'Profit')
```



In [74]:
```
boxplot_cols = ['Year', 'Customer_Age', 'Order_Quantity', 'Unit_Cost', 'Unit_Price', 'Profit']
sales[boxplot_cols].plot(kind='box', subplots=True, layout=(2,3), figsize=(14, 8))
```

Out[74]:
```
Year                AxesSubplot(0.125,0.536818;0.227941x0.343182)
Customer_Age      AxesSubplot(0.398529,0.536818;0.227941x0.343182)
Order_Quantity    AxesSubplot(0.672059,0.536818;0.227941x0.343182)
Unit_Cost            AxesSubplot(0.125,0.125;0.227941x0.343182)
Unit_Price        AxesSubplot(0.398529,0.125;0.227941x0.343182)
Profit            AxesSubplot(0.672059,0.125;0.227941x0.343182)
dtype: object
```



In [98]:
```
sales['Country'].unique()
```

Out[98]:
```
array(['Canada', 'Australia', 'United States', 'Germany', 'France',
       'United Kingdom'], dtype=object)
```

In [96]:
```
sales.loc[sales['Age_Group'] == 'Adults (35-64)', 'Revenue'].mean()
```

Out[96]:
```
762.8287654055604
```

In [100]:
```
sales.loc[(sales['Age_Group'] == 'Adults (35-64)') & (sales['Country'] == 'Australia'), 'Revenue'].mean()
```

Out[100]:
```
894.8311525880315
```

In [101]:
```
sales.loc[(sales['Age_Group'] == 'Adults (35-64)') & (sales['Country'] == 'United States'), 'Revenue'].mean()
```

Out[101]:
```
726.7260473588342
```

In [104]:
```
sales.loc[(sales['Age_Group'] == 'Adults (35-64)') & (sales['Country'] == 'Canada'), 'Revenue'].mean()
```

Out[104]: 616.0251466890193

In [105]:
```
sales.loc[(sales['Age_Group'] == 'Adults (35-64)') & (sales['Country'] == 'Germany'), 'Revenue'].mean()
```

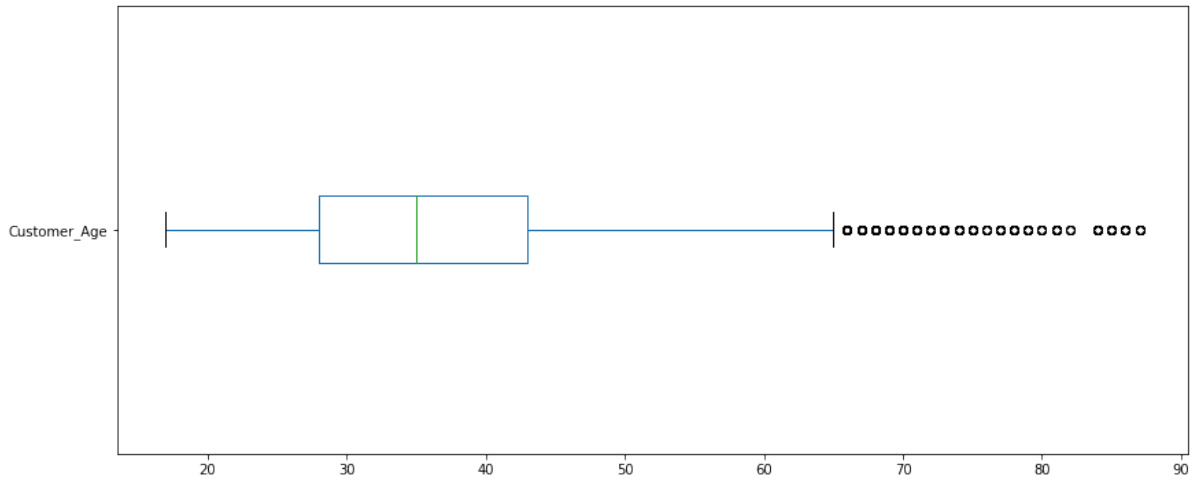
Out[105]: 839.0201314708299

In [110]:
```
sales['Customer_Age'].mean()
```

Out[110]: 35.91921157861212

In [113]:
```
sales['Customer_Age'].plot(kind='box',vert=False, figsize=(14,6))
```

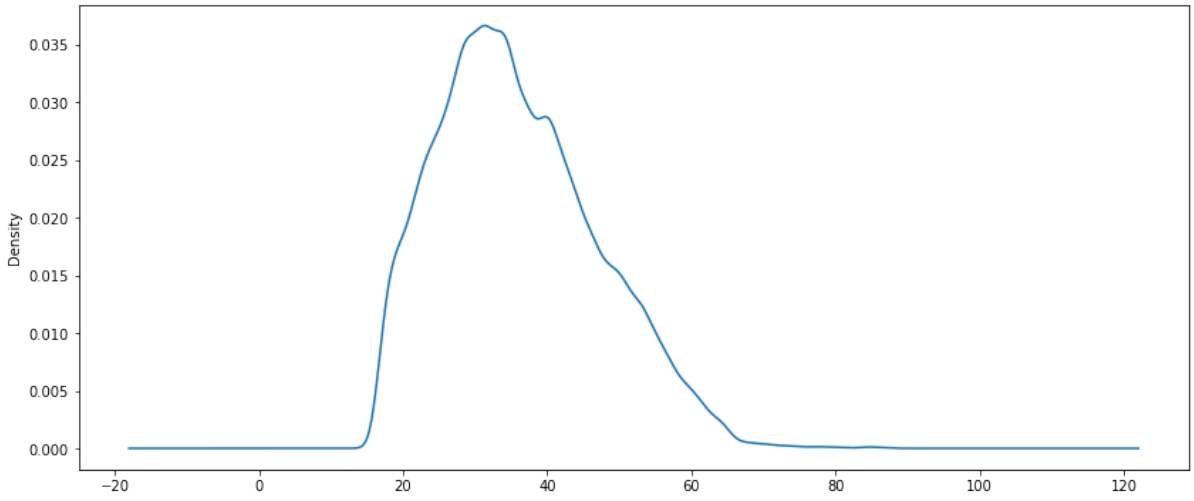
Out[113]: <matplotlib.axes._subplots.AxesSubplot at 0x1d8bd4e34c8>



In [ ]:

In [114]:
```
sales['Customer_Age'].plot(kind='kde', figsize=(14,6))
```

Out[114]: <matplotlib.axes._subplots.AxesSubplot at 0x1d8bd559d88>

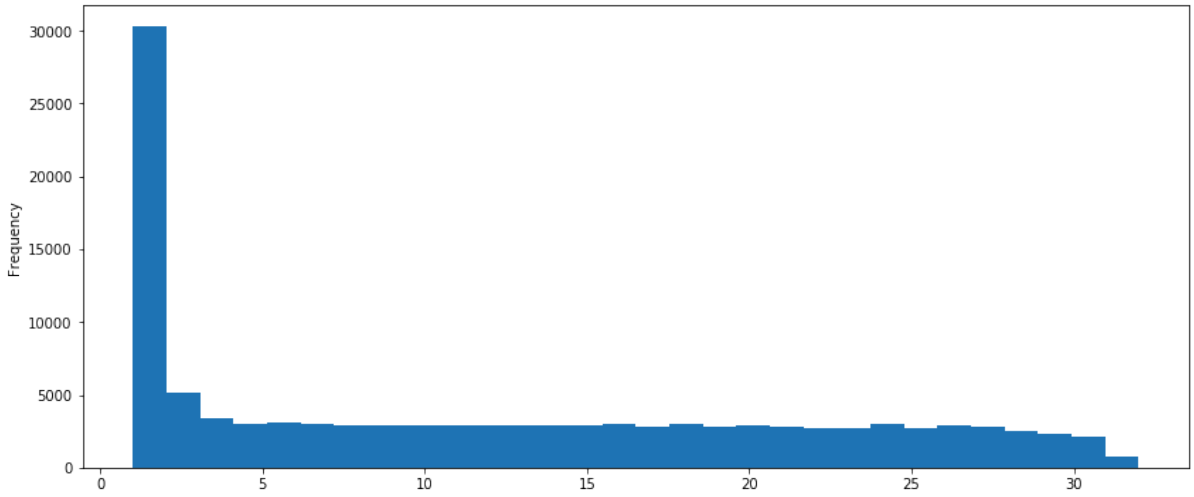


In [115]:
```
sales['Order_Quantity'].mean()
```

Out[115]:
```
11.901659648253654
```

In [116]:
```
sales['Order_Quantity'].plot(kind='hist', bins=30, figsize=(14,6))
```

Out[116]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1d8becb6408>
```



In [119]:
```
sales.columns
```

Out[119]:
```
Index(['Date', 'Day', 'Month', 'Year', 'Customer_Age', 'Age_Group',
       'Customer_Gender', 'Country', 'State', 'Product_Category',
       'Sub_Category', 'Product', 'Order_Quantity', 'Unit_Cost', 'Unit_Pric
e',
       'Profit', 'Cost', 'Revenue'],
      dtype='object')
```

In [121]:
```
sales['Year'].unique()
```

Out[121]:
```
array([2013, 2015, 2014, 2016, 2012, 2011], dtype=int64)
```

In [122]:
```
sales['Year'].value_counts()
```

Out[122]:
```
2016    29398
2014    29398
2015    24443
2013    24443
2012     2677
2011     2677
Name: Year, dtype: int64
```

In [131]:
```
print('Total Revenue in 2013')
print(sales.loc[(sales['Year'] == 2013), 'Revenue'].sum())
print('Total Revenue in 2014')
print(sales.loc[(sales['Year'] == 2014), 'Revenue'].sum())
print('Total Revenue in 2015')
print(sales.loc[(sales['Year'] == 2015), 'Revenue'].sum())
print('Total Revenue in 2016')
print(sales.loc[(sales['Year'] == 2016), 'Revenue'].sum())
print('  ')
print("Total Revenue")
print(sales['Revenue'].sum())
```

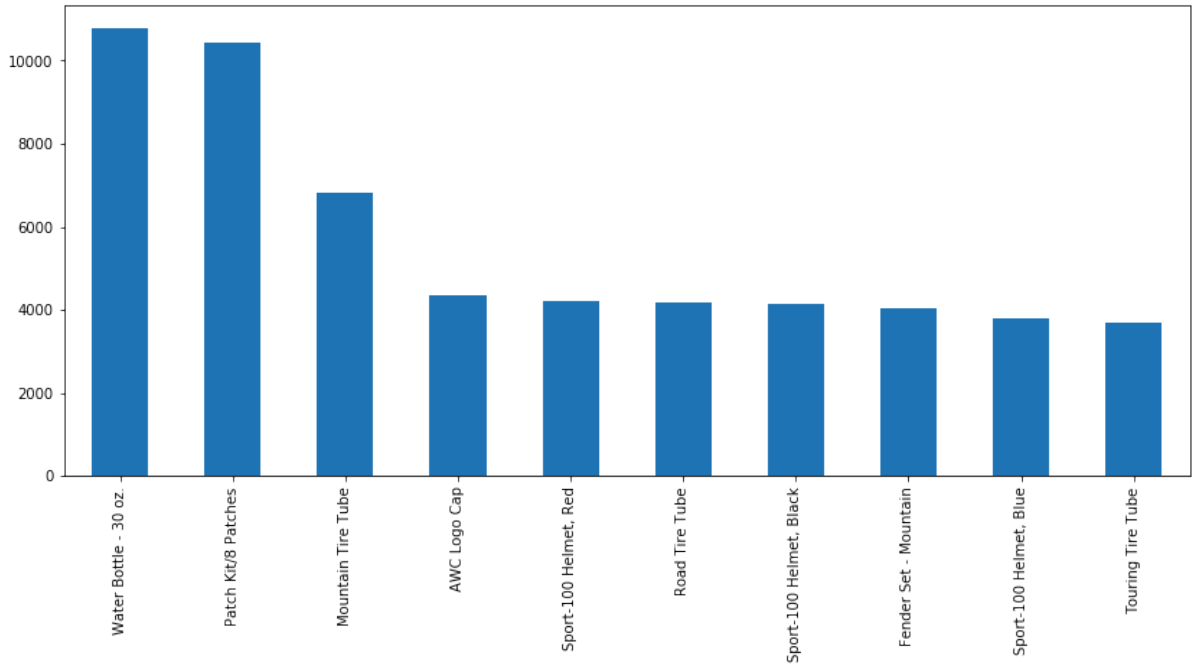
```
Total Revenue in 2013
15240037
Total Revenue in 2014
14152724
Total Revenue in 2015
20023991
Total Revenue in 2016
17713385

Total Revenue
85271008
```

In [132]:
```
sales['Country'].value_counts()
```

Out[132]:
```
United States     39206
Australia         23936
Canada            14178
United Kingdom    13620
Germany           11098
France            10998
Name: Country, dtype: int64
```

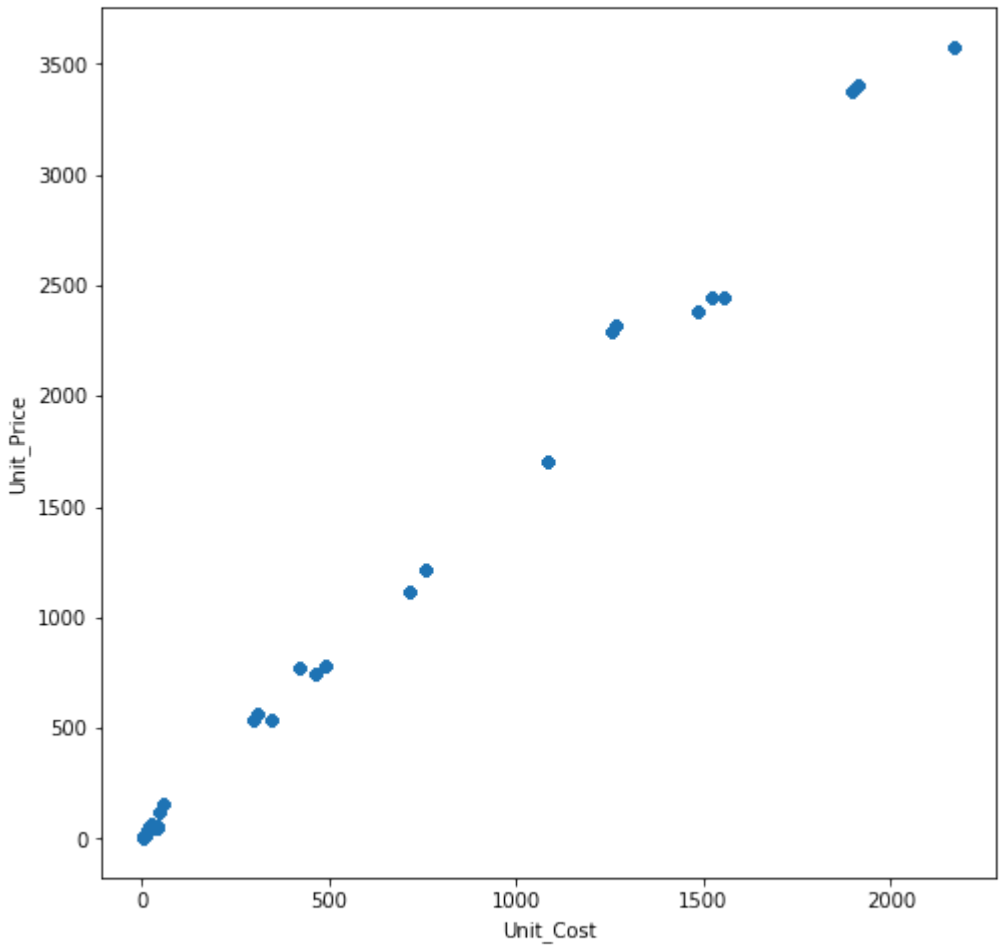
In [133]:
```
sales['Product'].value_counts().head(10).plot(kind='bar', figsize=(14,6))
```

Out[133]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1d8beb63288>
```



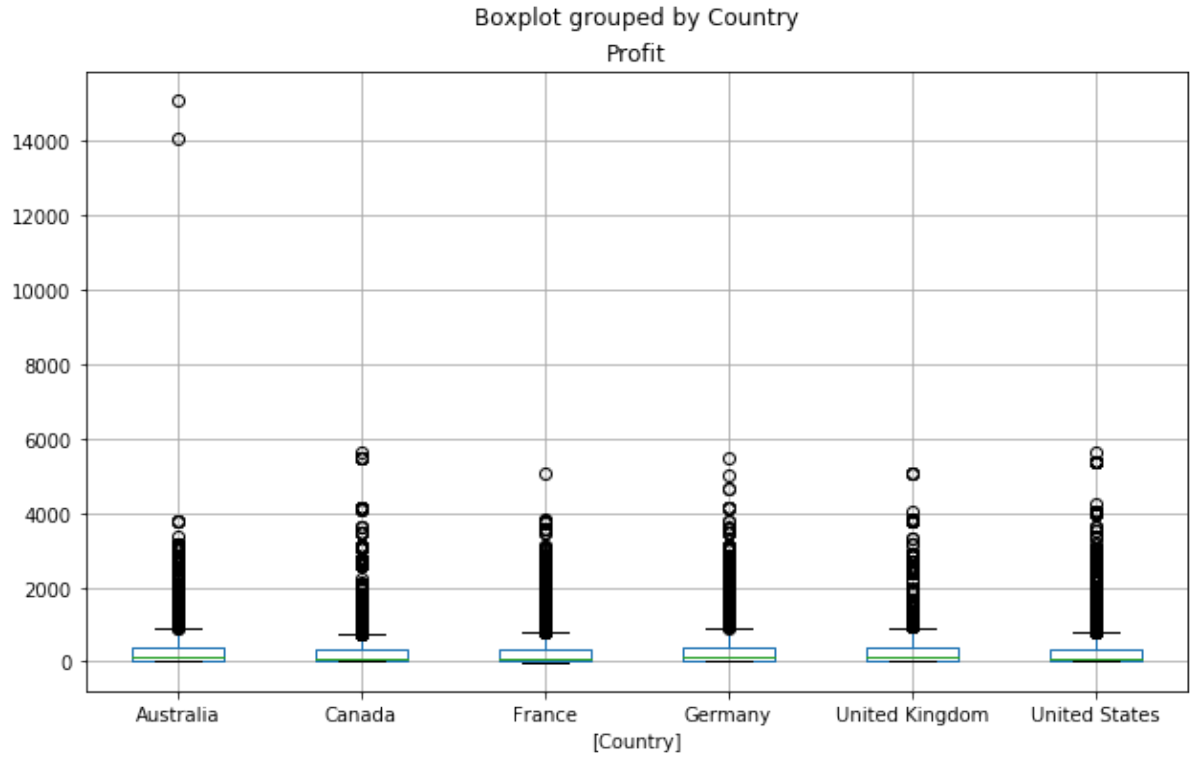
In [135]:
```
sales.plot(kind='scatter', x='Unit_Cost', y='Unit_Price', figsize=(8,8))
```

Out[135]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1d8bebce148>
```



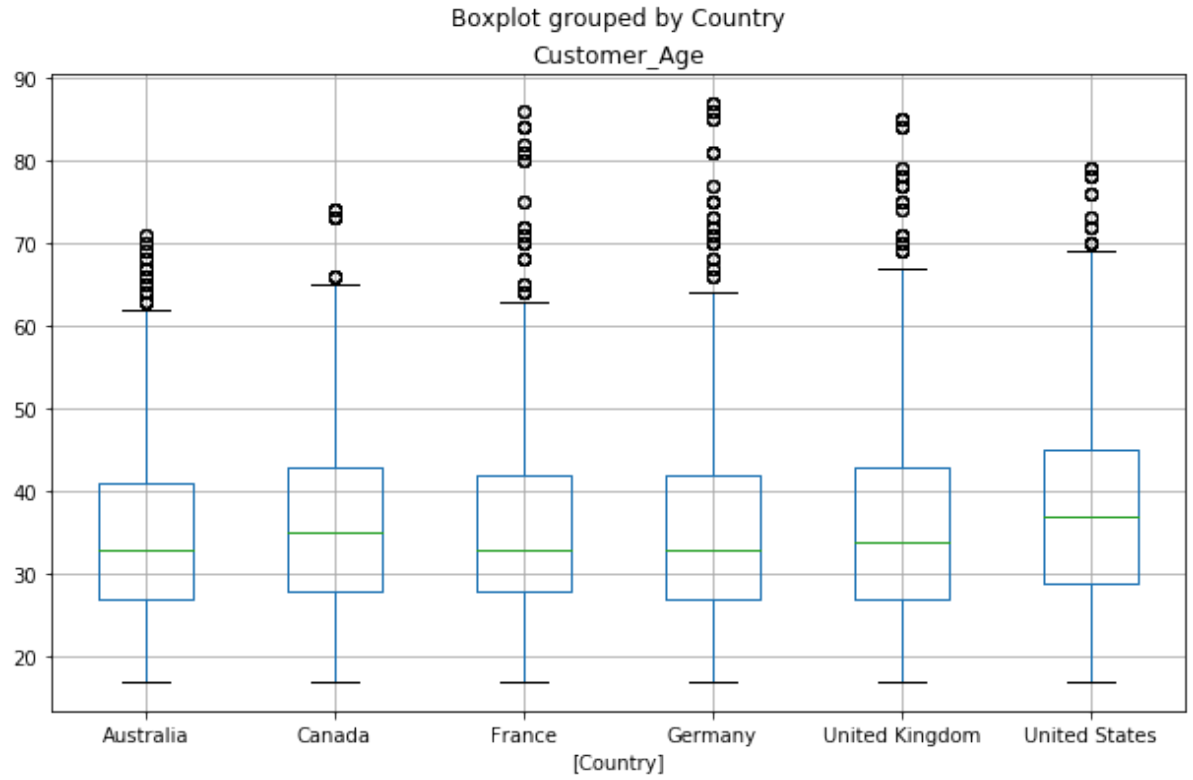
In [139]:
```
sales[['Profit', 'Country']].boxplot(by='Country', figsize=(10,6))
```

Out[139]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1d8bec13688>
```



In [140]:
```
sales[['Customer_Age', 'Country']].boxplot(by='Country', figsize=(10,6))
```

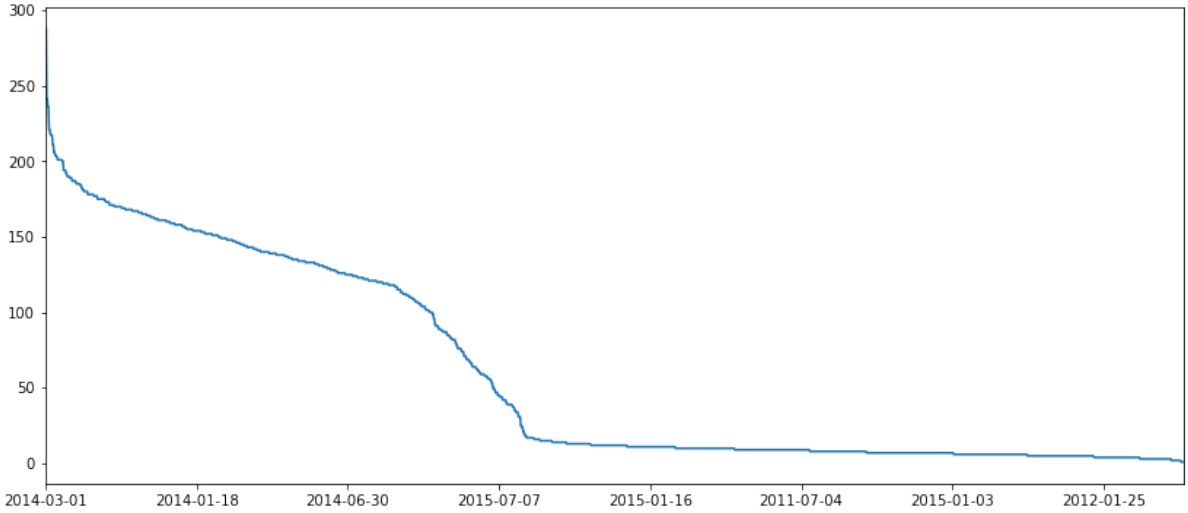
Out[140]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1d8be8f0308>
```



How did sales evolved throghh out the year

In [141]:
```
sales['Date'].value_counts().plot(kind='line', figsize=(14,6))
```

Out[141]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1d8c0375a88>
```



How many Bike Racks orders were made from Canada

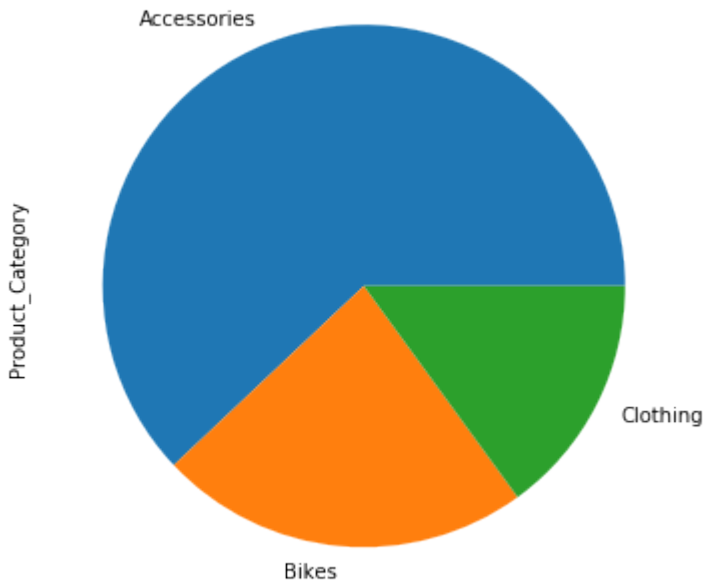
In [142]:
```
sales.loc[(sales['Country'] == 'Canada') & (sales['Sub_Category'] == 'Bike Racks')].shape[0]
```

Out[142]:
```
104
```

How much sales was made for each product category

In [145]:
```
sales['Product_Category'].value_counts().plot(kind='pie', figsize=(6,6))
```

Out[145]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1d8c03f2f48>
```



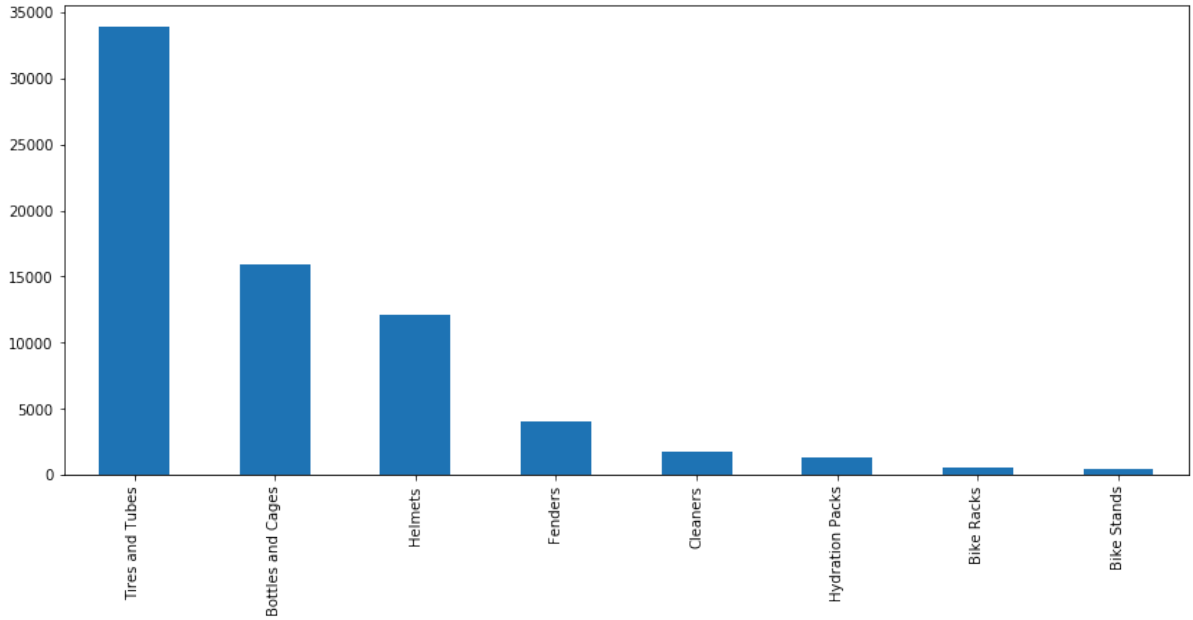
How many orders were made per accessory sub-categories?

In [148]:
```
Accessories = sales.loc[sales['Product_Category'] == 'Accessories', 'Sub_Categ
ory'].value_counts()


Accessories
```

Out[148]:
```
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 [149]:
```
Accessories.plot(kind='bar', figsize=(14,6))
```

Out[149]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1d8c0535588>
```



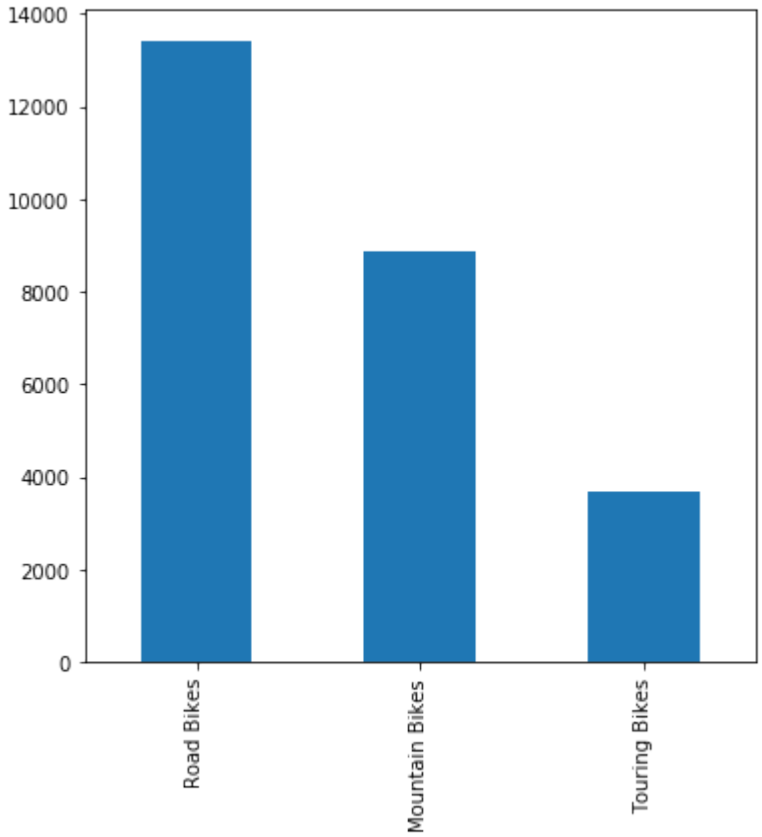
How many orders were made per bike sub_category?

In [150]:
```
Bike_sub = sales.loc[sales['Product_Category']== 'Bikes', 'Sub_Category'].valu
e_counts()
Bike_sub
```

Out[150]:
```
Road Bikes        13430
Mountain Bikes     8854
Touring Bikes      3698
Name: Sub_Category, dtype: int64
```

In [152]:
```
Bike_sub.plot(kind='bar', figsize=(6,6))
```

Out[152]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1d8c06143c8>
```



Which gender has the most amount of sales?

In [153]:
```
sales['Customer_Gender'].value_counts()
```

Out[153]:
```
M    58312
F    54724
Name: Customer_Gender, dtype: int64
```

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

In [156]:
```
sales.loc[(sales['Customer_Gender'] == 'M') & (sales['Revenue'] == 500)].shape
[0]
```

Out[156]:
```
8
```

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

In [158]:
```
sales.loc[(sales['Customer_Gender'] == 'F') & (sales['Revenue'] == 500)].shape
[0]
```

Out[158]:
```
25
```

Get the top highest revenue sales

In [159]:
```
sales.sort_values(['Revenue'], ascending=False).head(5)
```

Out[159]:

	Date	Day	Month	Year	Customer_Age	Age_Group	Customer_Gender	Country	
112073	2015-07-24	24	July	2015	52	Adults (35-64)	M	Australia	Qu
112072	2013-07-24	24	July	2013	52	Adults (35-64)	M	Australia	Qu
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	

Get the sale with the highest revenue

In [166]:
```
sales.sort_values(['Revenue'], ascending=False).head(1)
```

Out[166]:

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

In [167]:
```
cond = sales['Revenue'] == sales['Revenue'].max()

sales.loc[cond]
```

Out[167]:

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

What is the mean order quantity of orders wtih more than 10K in revenue?

In [171]:
```
over10k= sales['Revenue'] > 10000
```

In [172]:
```
sales.loc[over10k, 'Order_Quantity'].mean()
```

Out[172]: 3.7218934911242605

In [175]:
```
sales.loc[sales['Revenue'] > 10000, 'Order_Quantity'].mean()
```

Out[175]: 3.7218934911242605

How many orders were made in May 2016

In [177]:
```
# get the indeces where these two conditionsn are met
May2016 = (sales['Month'] == 'May') & (sales['Year'] == 2016)
```

In [183]:
```
#filter the onees with True
```

In [182]:
```
sales.loc[May2016].shape[0]
```

Out[182]: 5015

How many orders were made between May and July 2016

use insin(["May", 'June', 'July'])

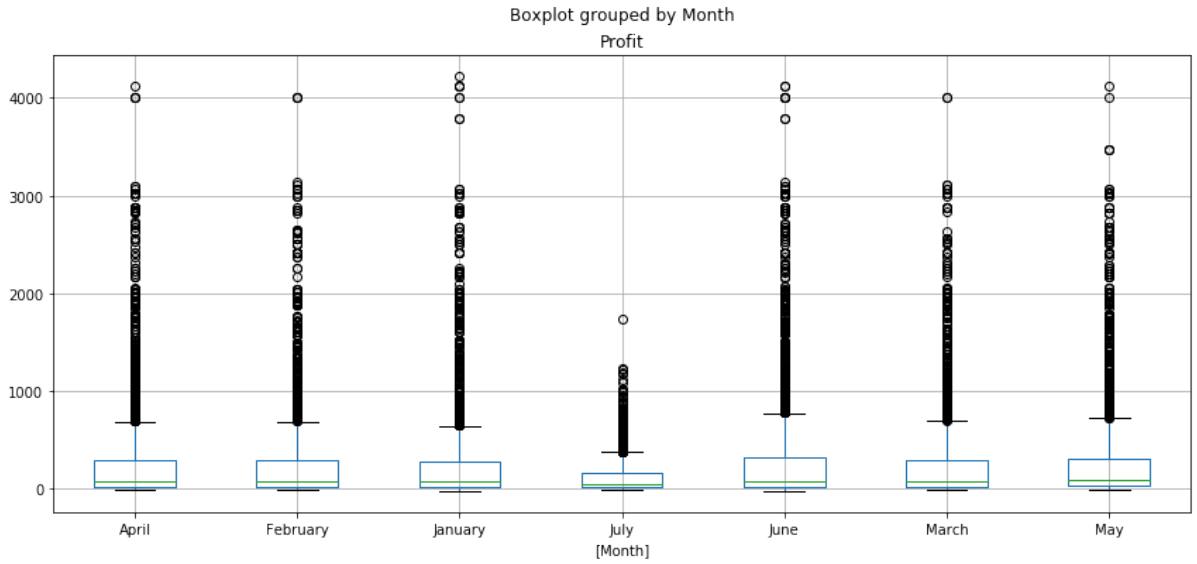
In [188]:
```
selected = (sales['Year'] == 2016) & (sales['Month'].isin(['May', 'June', 'Jul
y']))
```

In [190]:
```
sales.loc[selected].shape[0]
```

Out[190]: 12164

In [193]:
```
sales.loc[sales['Year']== 2016, ['Profit', 'Month']].boxplot(by='Month', figsize=(14,6))
```

Out[193]: <matplotlib.axes._subplots.AxesSubplot at 0x1d8c2bc2fc8>



In [ ]: