Product Analytics Project

Project Goals

Project Goals

- Analyze the overall time-series trends in revenue and numbers of purchases
- Analyze the purchase patterns of repeat purchase customers
- Analyze the trends in products being sold

Dataset description:

- This is a transactional data set which contains all the transactions occurring between 2010 and 2011 for a UK based and registered non-store online retail. The company mainly sells unique all-occasion gifts. Many customers of the company are wholesalers

Attribute Information

- InvoiceNo: Invoice number. Nominal, a 6-digit integral number uniquely assigned to each transaction. If this code starts with letter 'c', it indicates a cancellation.
- StockCode: Product (item) code. Nominal, a 5-digit integral number uniquely assigned to each distinct product.
- Description: Product (item) name. Nominal.
- Quantity: The quantities of each product (item) per transaction. Numeric.
- InvoiceDate: Invice Date and time. Numeric, the day and time when each transaction was generated.
- UnitPrice: Unit price. Numeric, Product price per unit in sterling.
- CustomerID: Customer number. Nominal, a 5-digit integral number uniquely assigned to each customer.
- Country: Country name. Nominal, the name of the country where each customer resides.

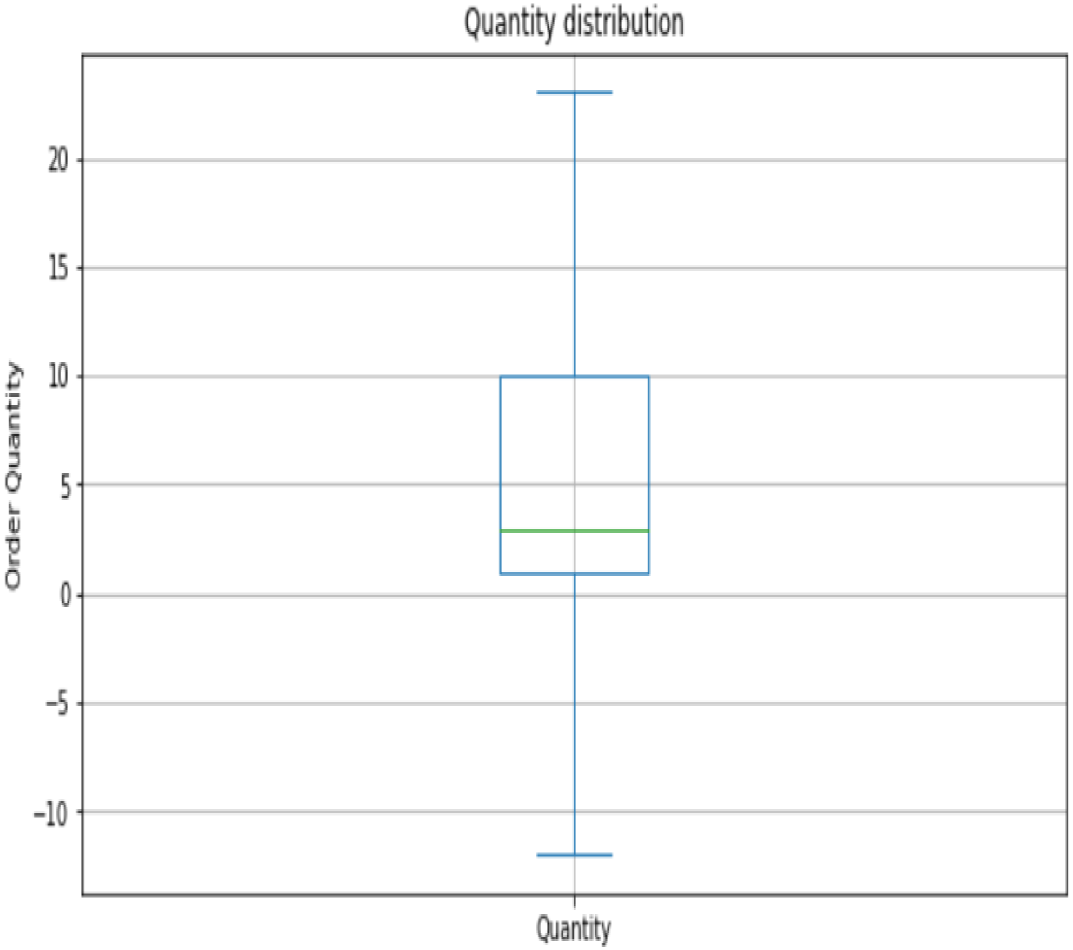
Cleaning Data

- Use the pd.read_excel() function to read the Online Retail.xlsx file into dataframe called df
- Use the DataFrame.info() and DataFrame.head() methods to print information about both dataframes, as well as the first few rows
- Check duplicated rows
- Which attribute determines other attributes?

Cleaning Data

- Visualize the distribution of *Quantity* attribute using pandas DataFrame’s plot.box
- What anomalies have you observed?
- Make a hypothesis about these anomalies and check this hypothesis.
- Filter out all cancelled orders

Cleaning Data



```
df['OrderCancel'] = df['InvoiceNo'].str.startswith('C')

df[df['Quantity']<0]['OrderCancel'].value_counts()
```

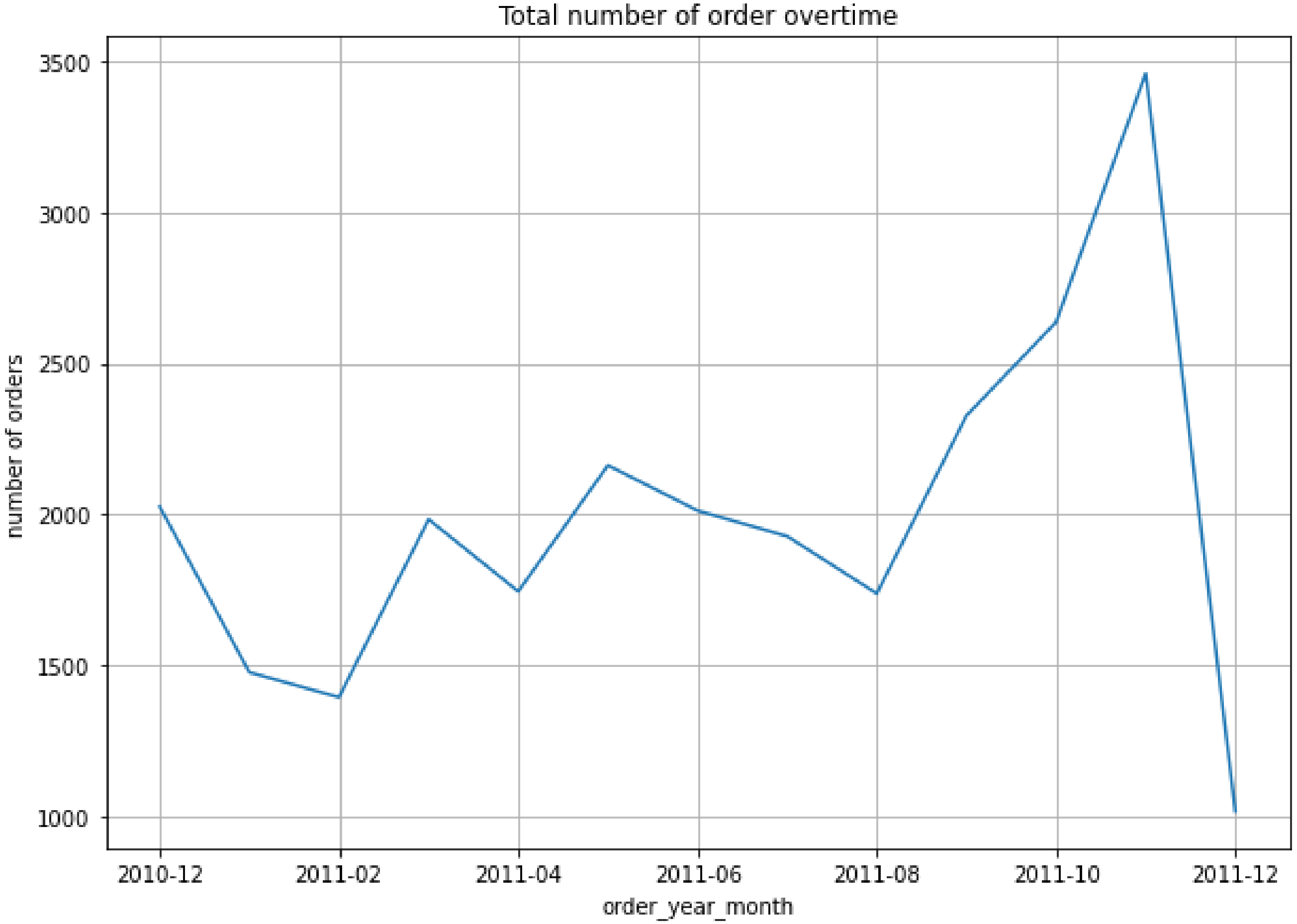
Analyst Time Series Trends

- Before we look at product-level data, as a marketer for an e-commerce business, it will be beneficial to have a better understanding of the overall time series trends in the revenue and the numbers of orders.
- It will help us understand whether the business is growing in terms of both the overall revenue and the numbers of orders we received over time
- Calculate the numbers of orders received per month

Analyst Time Series Trends

- Calculate the numbers of orders received by month
  - Create new column named *year-month* to extract year and month from *InvoiceDate* column using strftime
  - Groupby year-month column and count the number of unique InvoiceNo for each month
  - Visualize monthly time-series data using line charts

Analyst Time Series Trends



- What do you observe here?

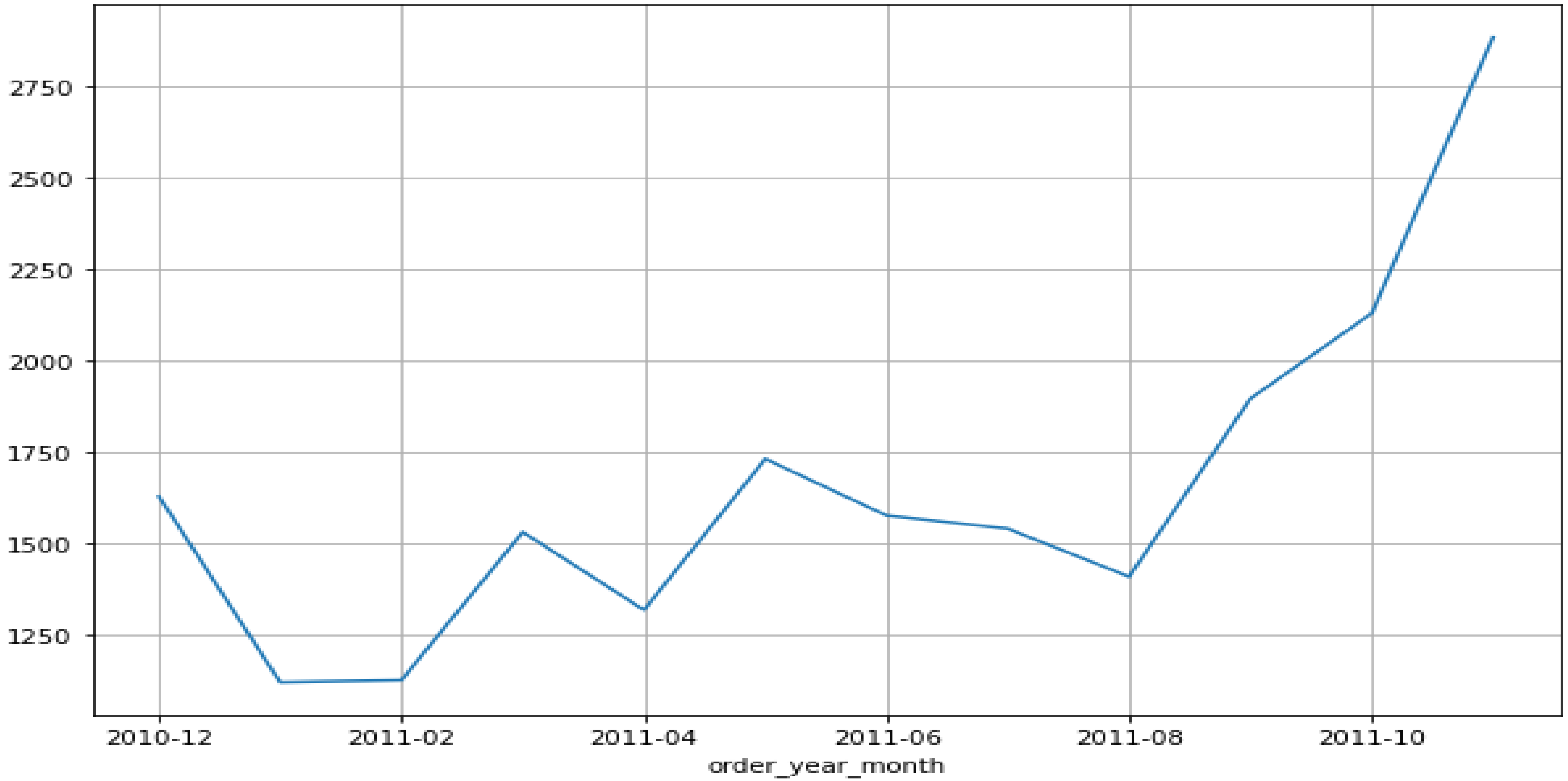
```
df['order_year_month'] = df['InvoiceDate'].dt.strftime('%Y-%m')
df.groupby('order_year_month').InvoiceNo.nunique()
ax = df.groupby('order_year_month').InvoiceNo.nunique().plot(figsize=(10,7),grid=True)
ax.set_title('Total number of order overtime')
ax.set_ylabel('number of orders')
```

```
invoice_dates = df.loc[df['order_year_month']>='2011-12','InvoiceDate']
print('Min date: {x}\nMax date{y}'.format(x=invoice_dates.min(),y=invoice_dates.max()))
```

Analyst Time Series Trends

- Filter out incomplete data for December 2011
- Redraw the line chart
- Draw conclusions about this chart

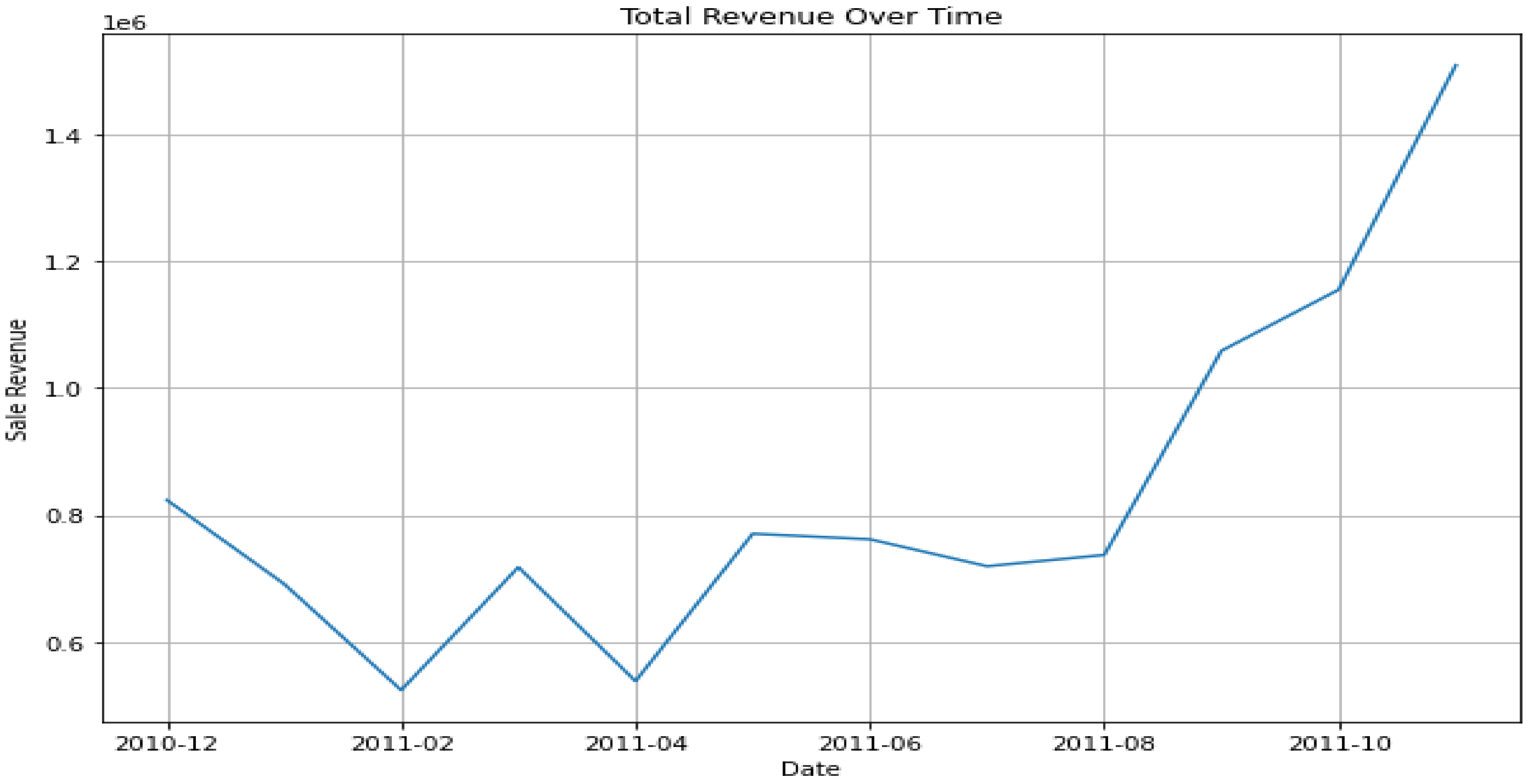
Analyst Time Series Trends



Analyst Time Series Trends

- Similar calculate monthly revenue data
  - Create new column named ‘*Sales’* as product of two columns *Quantity* and *UnitPrice*
  - Aggregate by *year-month* column, using sum as the aggregate function, we can get the monthly sales revenue data
  - Visualize this time-series data
  - Draw conclusions
- df['Revenue'] = df['Quantity']*df['UnitPrice']
- ax = df.groupby('order_year_month').Revenue.sum().plot(figsize=(10,7),grid=True)

Analyze Time-Series Trends



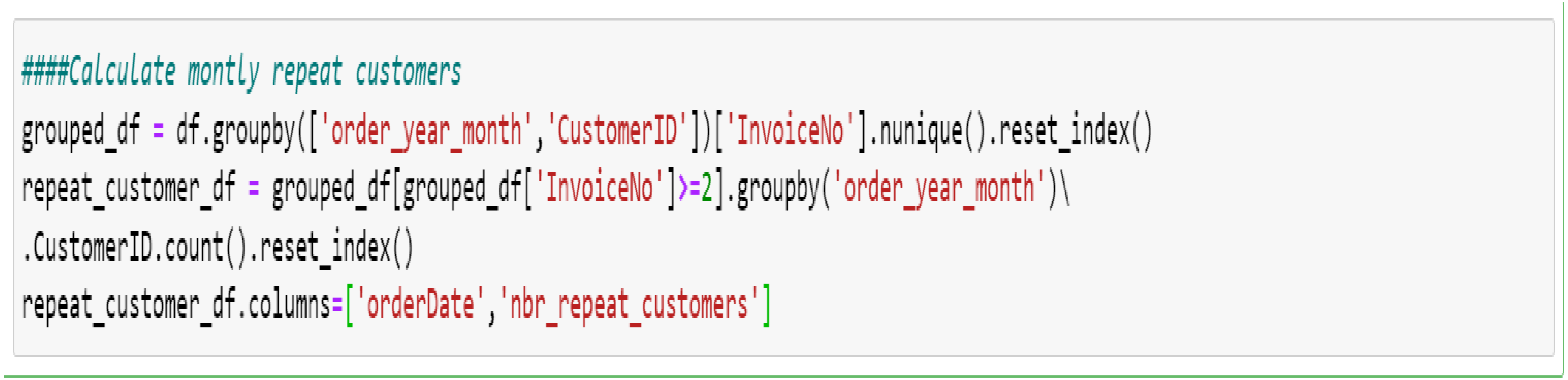
Repeat Customers

- Another important factor of a successful business is how well it is retaining customers and how many repeat purchases and customers it has.
- A typical strong and stable business has a steady stream of sales from existing customers
- We are going to analyze the number of monthly repeat customers and how much of the monthly revenue contributes to these repeat purchases customers
- *Repeat customer: a customer makes more than one order within a given month*
- **Task**: **Calculate the number of repeat customers per month**

Repeat Customers

- Groupby *year_month* and *CustomerID* column, count unique InvoiceNo for each group. In this dataframe, each record gives us information of how many purchases each customer makes for a given month
- Filter out customers who make only 1 purchase for a given month
- Groupby *year_month* and count the number of *CustomerID* for each month

Repeat Customers

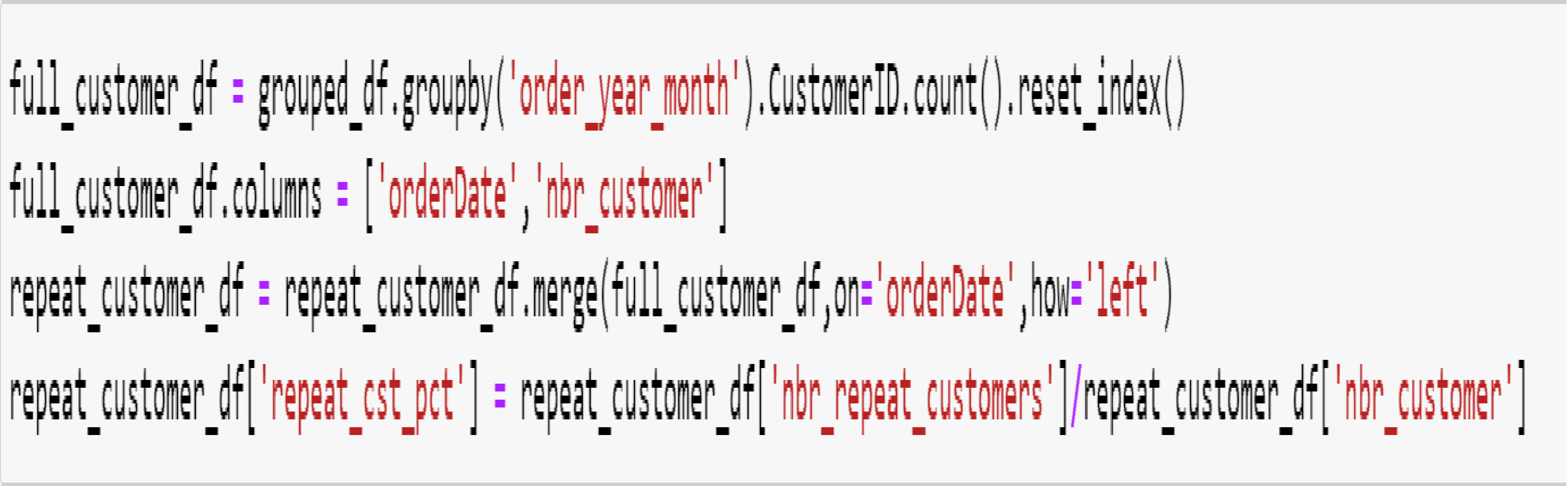


```
####Calculate montly repeat customers
grouped_df = df.groupby(['order_year_month','CustomerID'])['InvoiceNo'].nunique().reset_index()
repeat_customer_df = grouped_df[grouped_df['InvoiceNo']>=2].groupby('order_year_month')\
.CustomerID.count().reset_index()
repeat_customer_df.columns=['orderDate','nbr_repeat_customers']
```

Repeat Customers

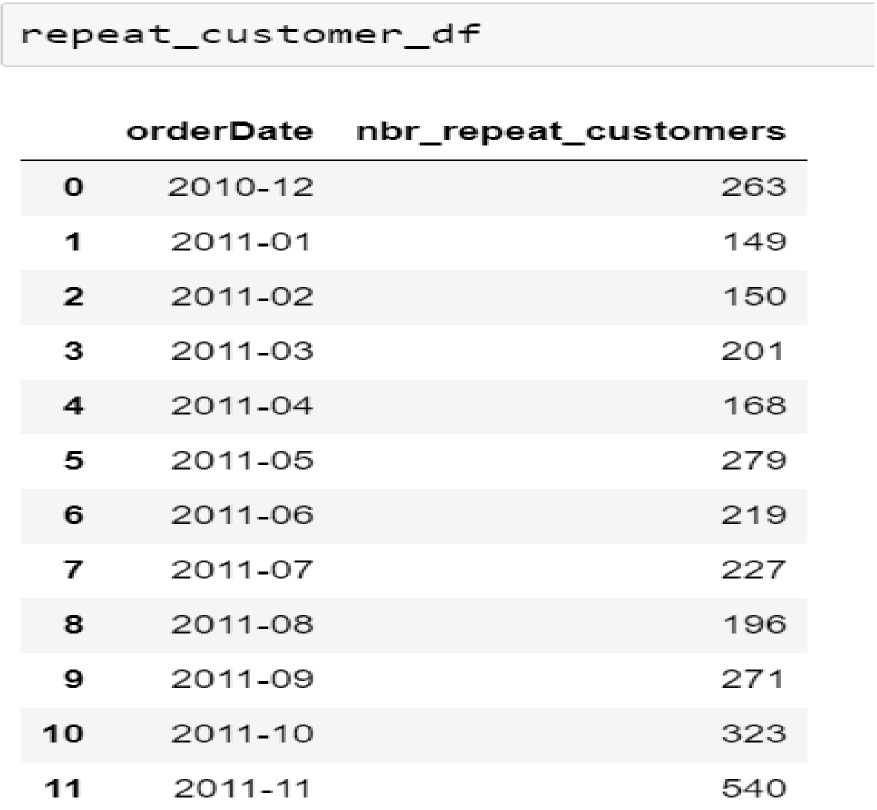
- Calculate the total number of customers monthly
- Calculate the percentage of repeat customers for each month
- Visualize all of this data in one chart

Repeat Customers



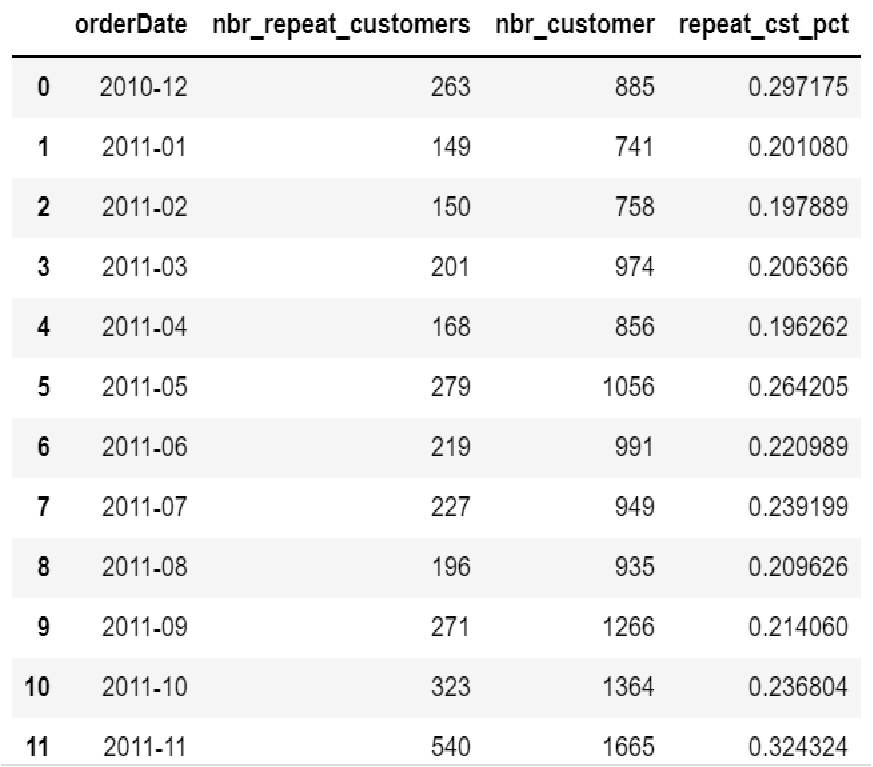
```
full_customer_df = grouped_df.groupby('order_year_month').CustomerID.count().reset_index()
full_customer_df.columns = ['orderDate','nbr_customer']
repeat_customer_df = repeat_customer_df.merge(full_customer_df,on='orderDate',how='left')
repeat_customer_df['repeat_cst_pct'] = repeat_customer_df['nbr_repeat_customers']/repeat_customer_df['nbr_customer']
```

Repeat Customers



```
repeat_customer_df
```

	orderDate	nbr_repeat_customers
0	2010-12	263
1	2011-01	149
2	2011-02	150
3	2011-03	201
4	2011-04	168
5	2011-05	279
6	2011-06	219
7	2011-07	227
8	2011-08	196
9	2011-09	271
10	2011-10	323
11	2011-11	540



	orderDate	nbr_repeat_customers	nbr_customer	repeat_cst_pct
0	2010-12	263	885	0.297175
1	2011-01	149	741	0.201080
2	2011-02	150	758	0.197889
3	2011-03	201	974	0.206366
4	2011-04	168	856	0.196262
5	2011-05	279	1056	0.264205
6	2011-06	219	991	0.220989
7	2011-07	227	949	0.239199
8	2011-08	196	935	0.209626
9	2011-09	271	1266	0.214060
10	2011-10	323	1364	0.236804
11	2011-11	540	1665	0.324324

Repeat Customers

- Analyze how much of the monthly revenue comes from these Repeat Customers
  - Groupby year_month, CustomerID using aggregation function: nunique for InvoiceNo and sum for Revenue
  - Filter out customers who make only one purchase
  - Groupby year_month , using aggregation function sum for Revenue column
  - Calculate total revenue each month
  - Calculate the percentage of revenues coming from repeat customers for each month

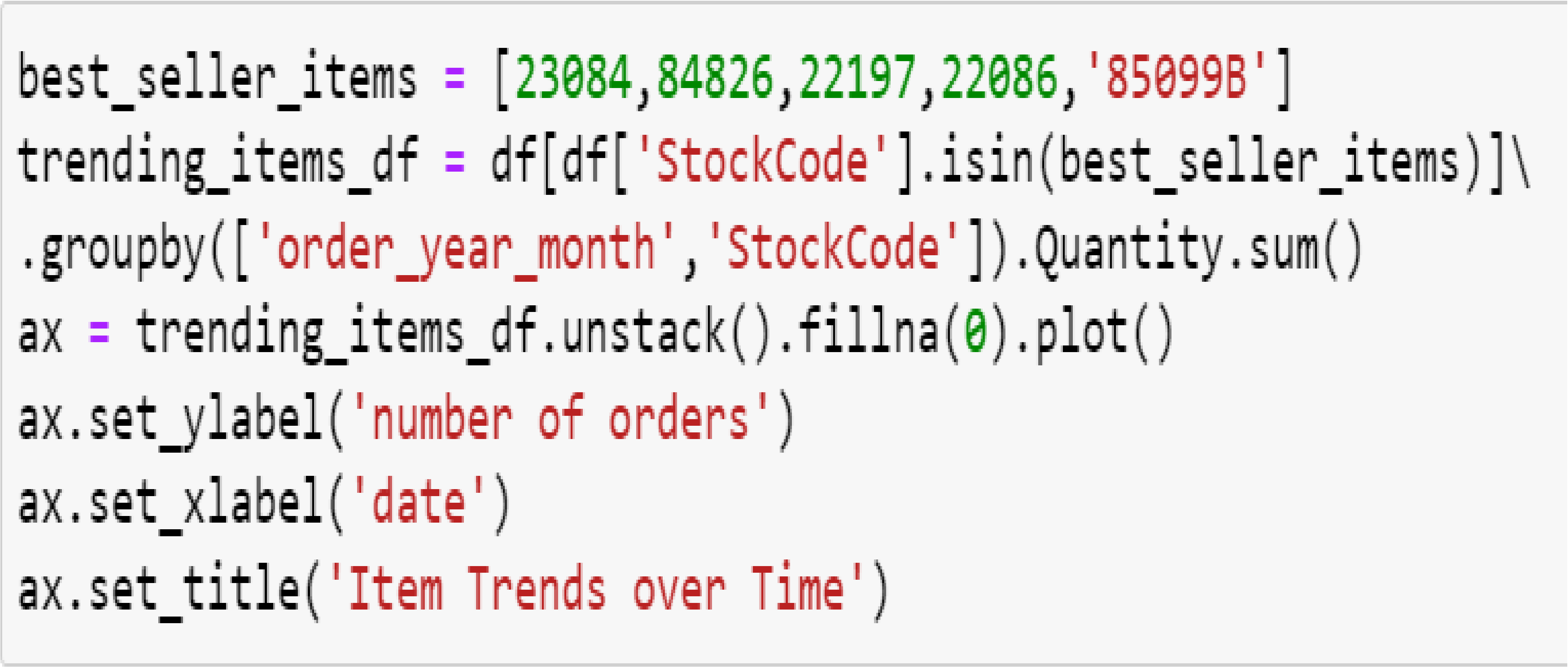
Trending Items Over Time

- So far, we have analyzed the overall time-series patterns and how customers engage with the overall business, but not how customers engage with individual products. In this section, we are going to explore and analyze how customers interact with individual products that have been sold. More specifically, we will take a look at the trends of the top five bestsellers over time.
- Let's count the number of items sold for each product for each period
  - Groupby year_month and StockCode column, using aggregation function sum for Quantity column
- What items were sold the most on November 30, 2011 ?

Trending Items Over Time

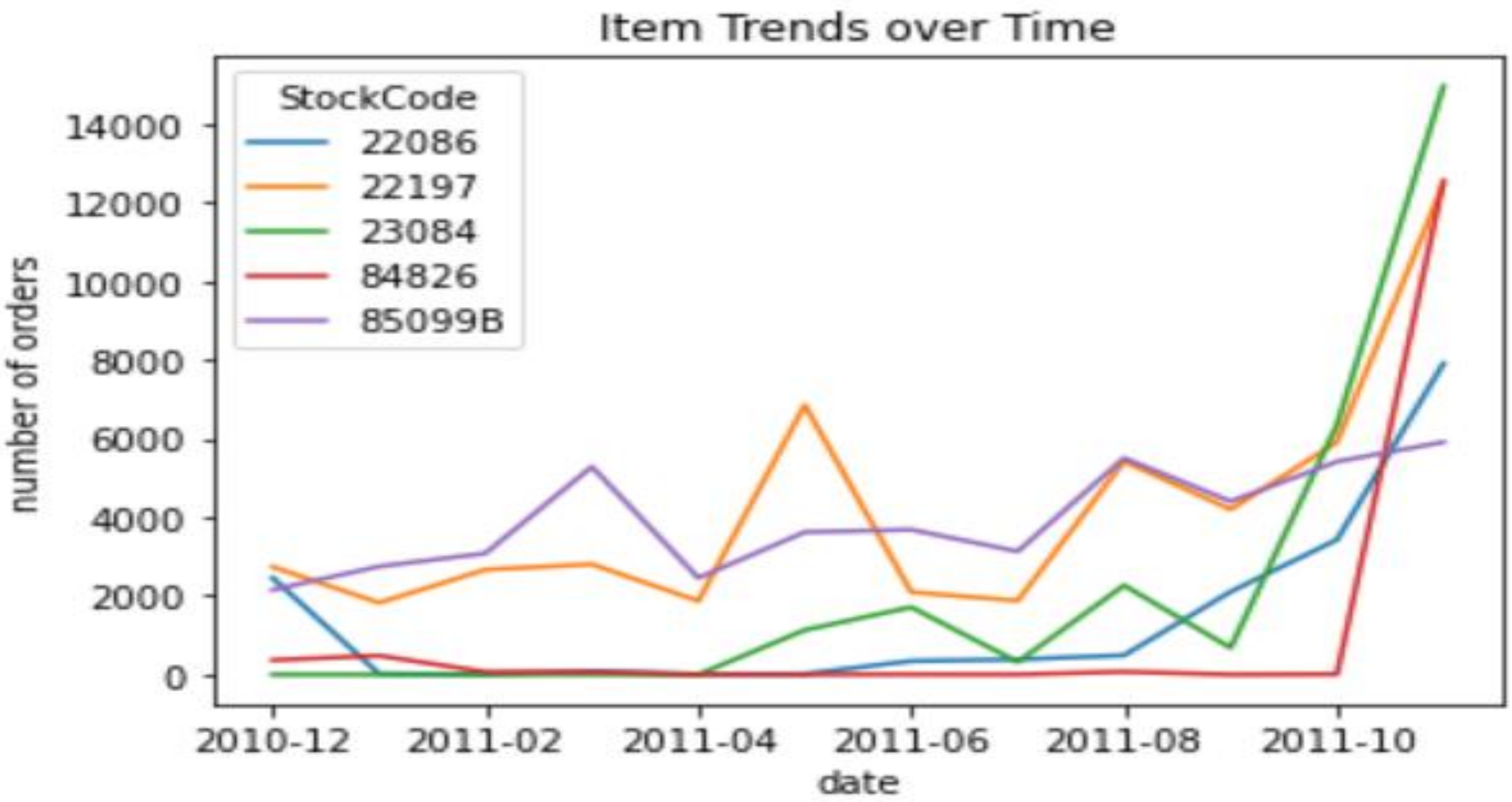
- As you can see from this result, the products with the codes 23084, 84826, 22197, 22086, and 85099B were the top five best-sellers in the month of November 2011.
- Calculate the monthly sales data for these five products again
  - Keep records with the StockCode that matches with the top five best-sellers’ item codes from the transaction dataset
  - Groupby year_month and StockCode, sum by Quantity
  - Visualize the trends of each item over time

Trending Items Over Time

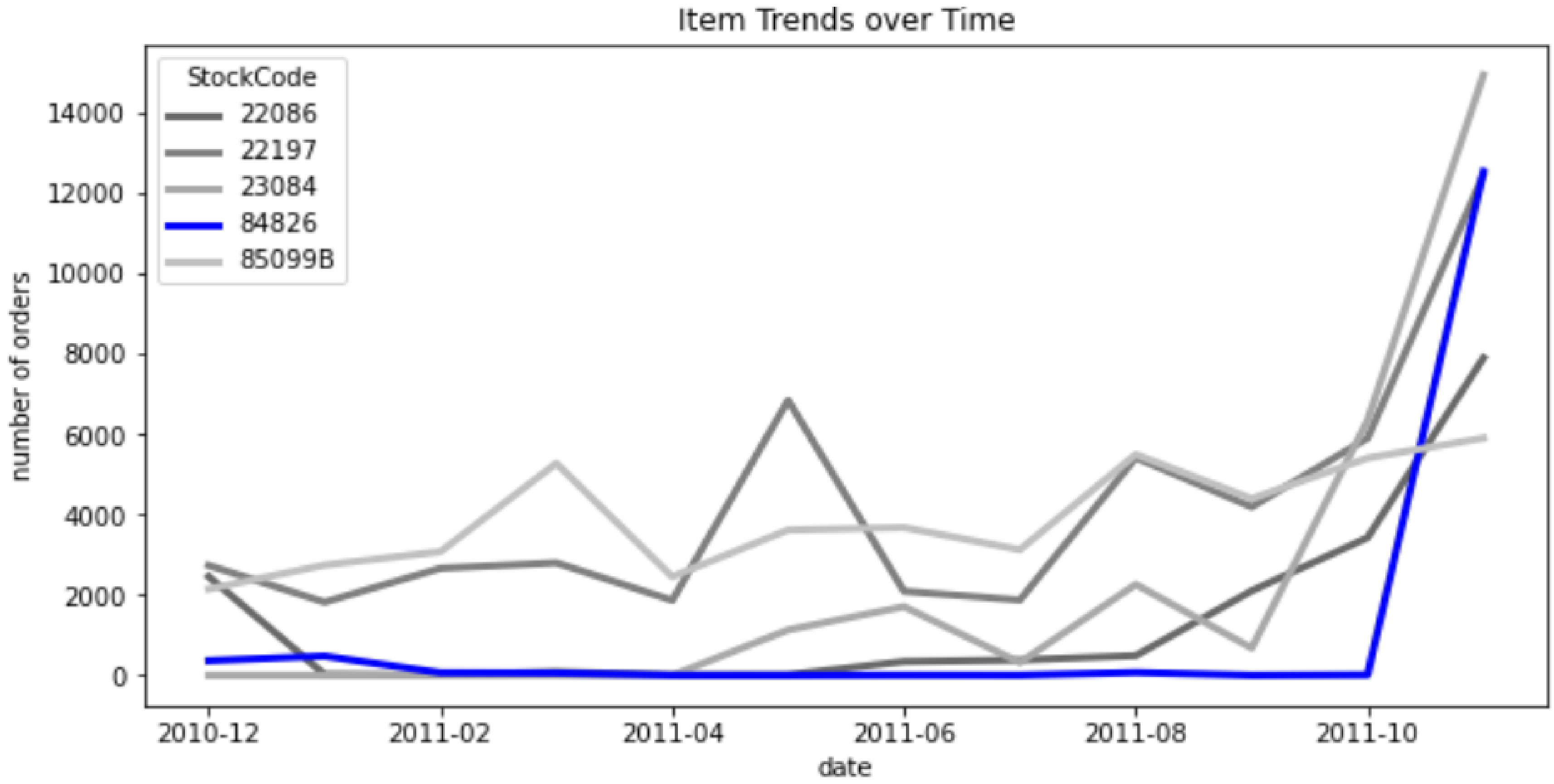


```
best_seller_items = [23084,84826,22197,22086,'85099B']
trending_items_df = df[df['StockCode'].isin(best_seller_items)]\
.groupby(['order_year_month','StockCode']).Quantity.sum()
ax = trending_items_df.unstack().fillna(0).plot()
ax.set_ylabel('number of orders')
ax.set_xlabel('date')
ax.set_title('Item Trends over Time')
```

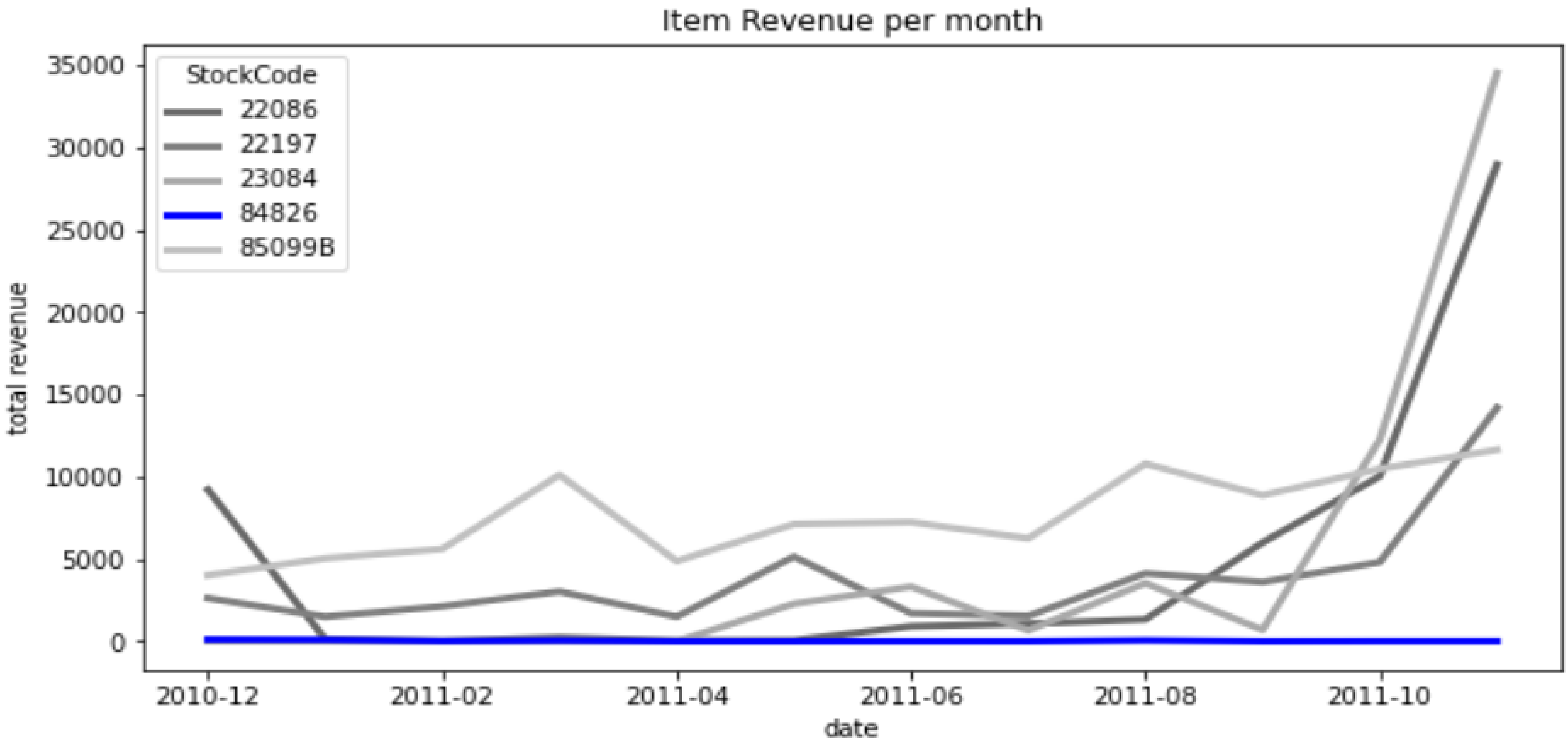
Trending Items Over Time

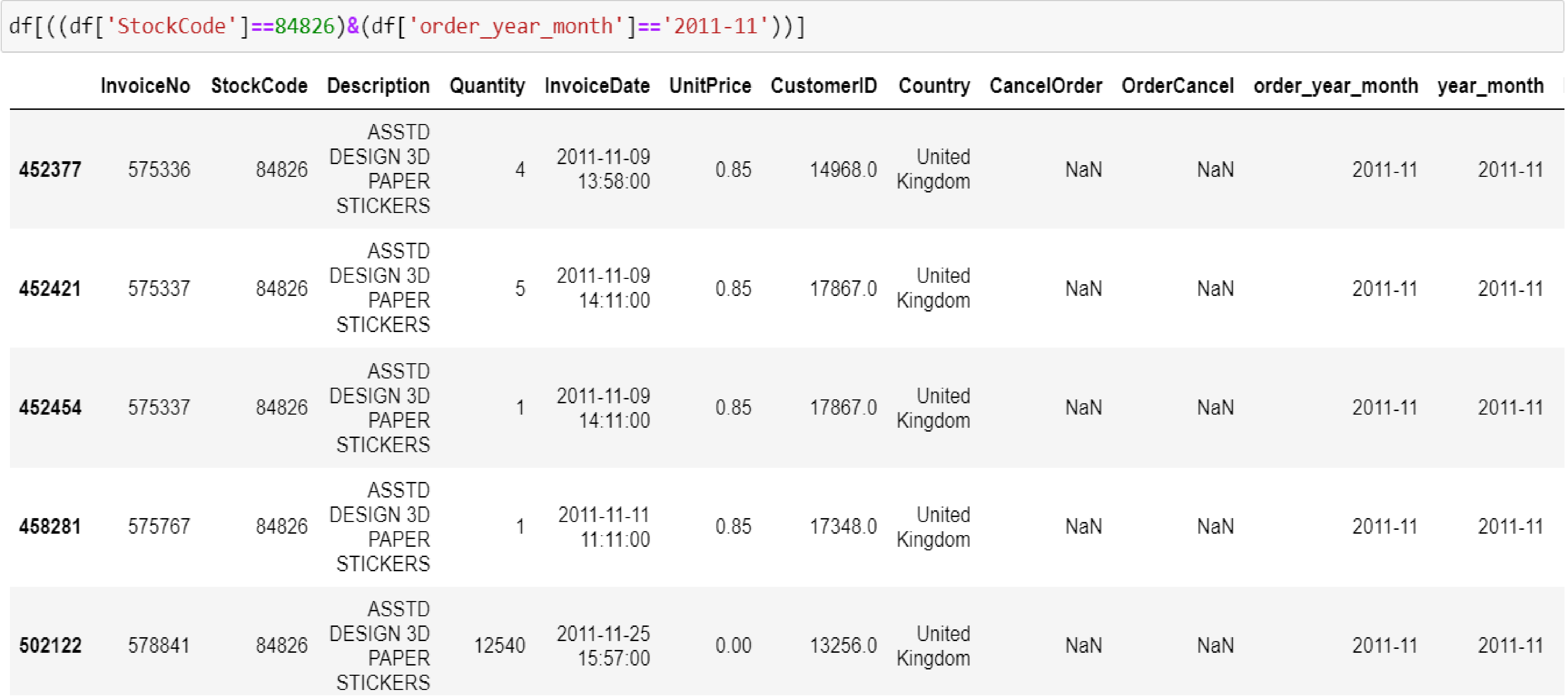


Trending Items Over Time



Trending Items Over Time

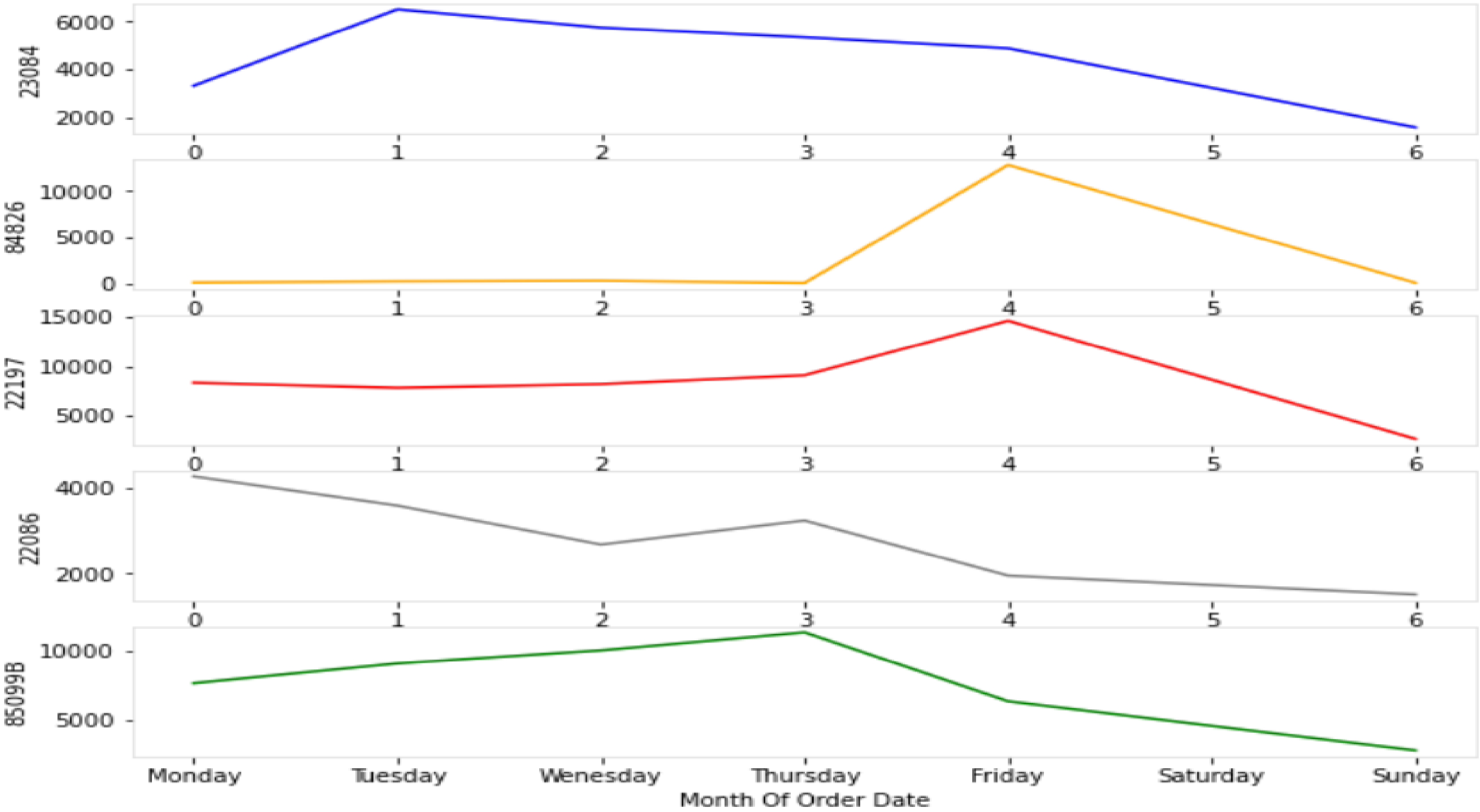


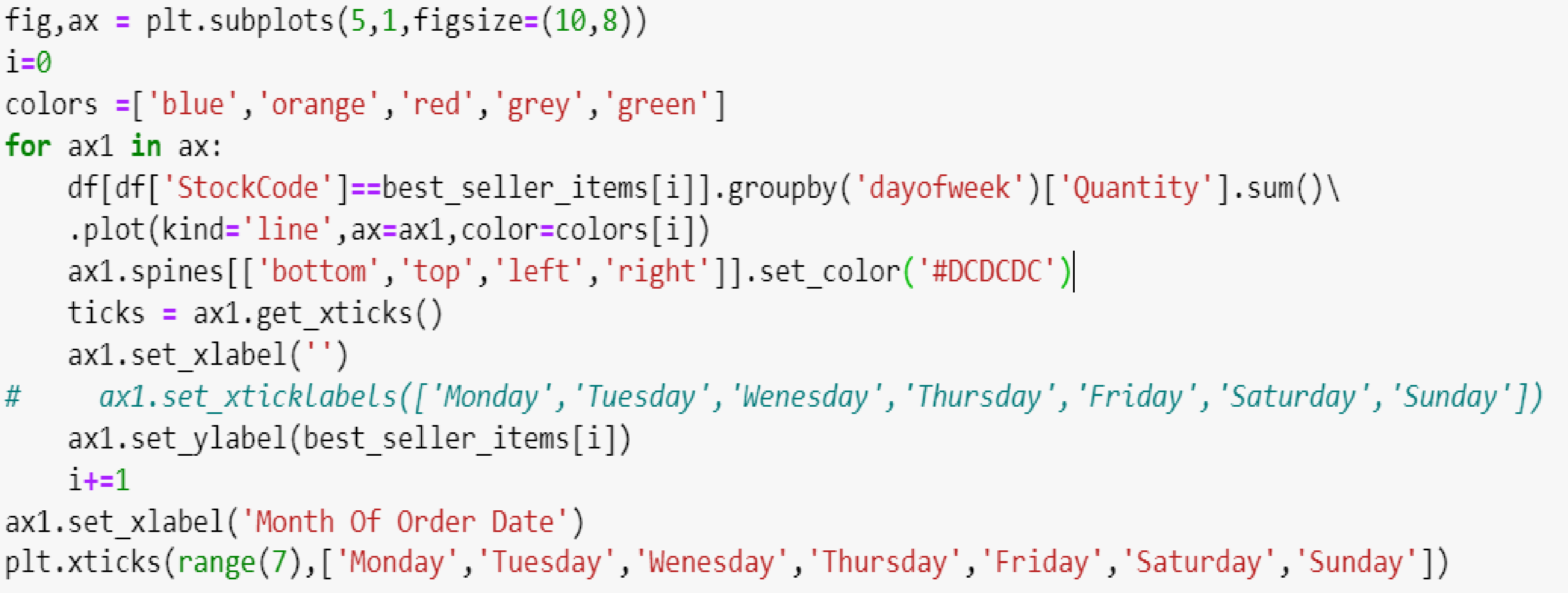


```
df[((df['StockCode']==84826)&(df['order_year_month']=='2011-11'))]
```

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country	CancelOrder	OrderCancel	order_year_month	year_month
452377	575336	84826	ASSTD DESIGN 3D PAPER STICKERS	4	2011-11-09 13:58:00	0.85	14968.0	United Kingdom	NaN	NaN	2011-11	2011-11
452421	575337	84826	ASSTD DESIGN 3D PAPER STICKERS	5	2011-11-09 14:11:00	0.85	17867.0	United Kingdom	NaN	NaN	2011-11	2011-11
452454	575337	84826	ASSTD DESIGN 3D PAPER STICKERS	1	2011-11-09 14:11:00	0.85	17867.0	United Kingdom	NaN	NaN	2011-11	2011-11
458281	575767	84826	ASSTD DESIGN 3D PAPER STICKERS	1	2011-11-11 11:11:00	0.85	17348.0	United Kingdom	NaN	NaN	2011-11	2011-11
502122	578841	84826	ASSTD DESIGN 3D PAPER STICKERS	12540	2011-11-25 15:57:00	0.00	13256.0	United Kingdom	NaN	NaN	2011-11	2011-11

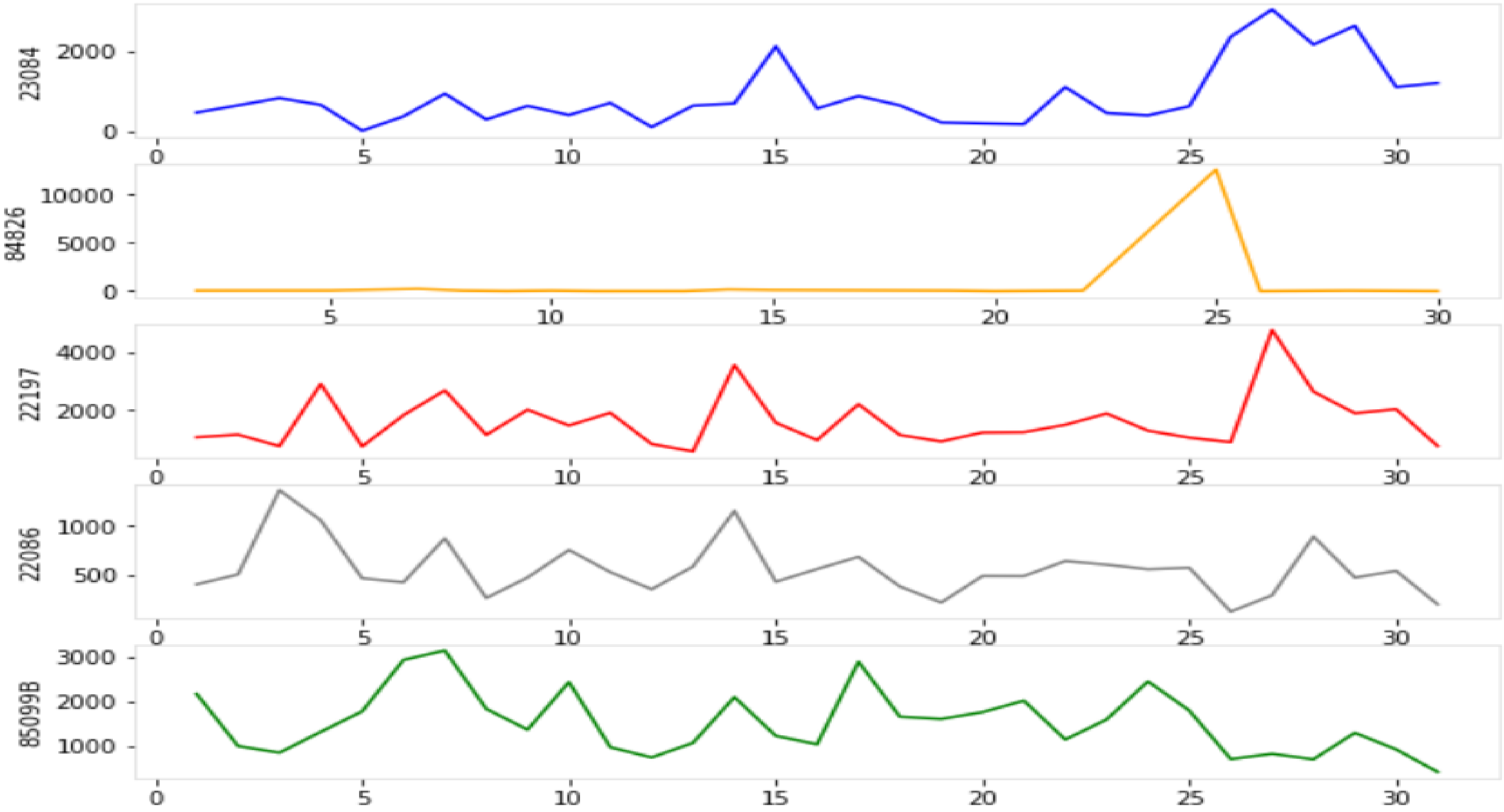
Trending Items Over Time





```
fig,ax = plt.subplots(5,1,figsize=(10,8))
i=0
colors =['blue','orange','red','grey','green']
for ax1 in ax:
    df[df['StockCode']==best_seller_items[i]].groupby('dayofweek')['Quantity'].sum()\
    .plot(kind='line',ax=ax1,color=colors[i])
    ax1.spines[['bottom','top','left','right']].set_color('#DCDCDC')
    ticks = ax1.get_xticks()
    ax1.set_xlabel('')
#     ax1.set_xticklabels(['Monday','Tuesday','Wenesday','Thursday','Friday','Saturday','Sunday'])
    ax1.set_ylabel(best_seller_items[i])
    i+=1
ax1.set_xlabel('Month Of Order Date')
plt.xticks(range(7),['Monday','Tuesday','Wenesday','Thursday','Friday','Saturday','Sunday'])
```

Trending Items Over Time



More Discussion

- Is UnitPrice changing over time?
- Does this change affect Sales and Revenues?
- Do the bestsellers change per country?
- Do the sales increase on weekends and end of each month?

Thank you!