In [ ]:
```
%matplotlib inline
import pandas as pd
import matplotlib.pyplot as plt

from datetime import datetime
data1 = pd.read_csv('https://storage.googleapis.com/dqlab-dataset/10%25_original_randomstate%3D42/retail_data_from_1_until_3_reduce.csv')
```

In [ ]:
```
data1.isna().sum()
```

Out[ ]:
```
order_id       0
order_date     0
customer_id    0
city           0
province       0
product_id     0
brand          0
quantity       0
item_price     0
total_price    0
dtype: int64
```

In [ ]:
```
data1.head(100)
```

Out[ ]:

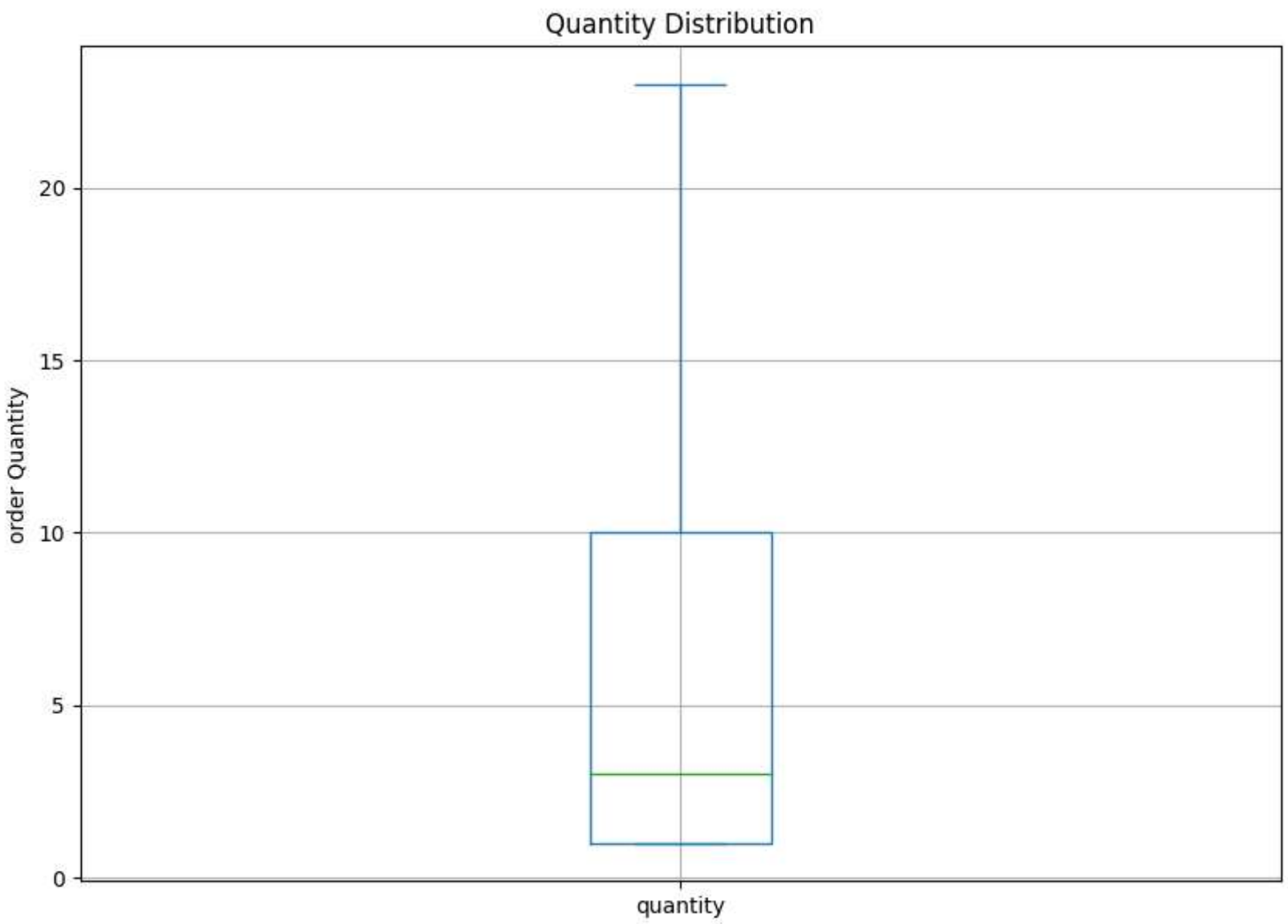
	order_id	order_date	customer_id	city	province	product_id	brand	quantity	item_price	total_price
0	1612885	01-01-19	16293	Malang	Jawa Timur	P1301	BRAND_F	6	747000	4482000
1	1612387	01-01-19	17228	Bogor	Jawa Barat	P2086	BRAND_L	4	590000	2360000
2	1612903	01-01-19	16775	Surakarta	Jawa Tengah	P1656	BRAND_G	3	1325000	3975000
3	1612963	01-01-19	0	unknown	unknown	P3127	BRAND_S	1	1045000	1045000
4	1612915	01-01-19	0	unknown	unknown	P1230	BRAND_E	1	-891000	891000
...	...	...	...	...	...	...	...	...	...	...
95	1612999	01-01-19	0	unknown	unknown	P4086	BRAND_W	3	593000	1779000
96	1612927	01-01-19	0	unknown	unknown	P2736	BRAND_P	1	-891000	-891000
97	1612843	01-01-19	16904	Bandung	Jawa Barat	P1193	BRAND_E	2	2095000	4190000
98	1612915	01-01-19	0	unknown	unknown	P3107	BRAND_S	1	593000	593000
99	1612852	01-01-19	16086	Jakarta Selatan	DKI Jakarta	P1655	BRAND_G	4	1325000	5300000

100 rows × 10 columns

In [ ]:
```
data1['order_date'] = pd.to_datetime(data1['order_date'])
```

In [ ]:
```
ax =data1['quantity'].plot.box(
    showfliers = False,
    grid = True,
    figsize = (10,7)
)
ax.set_ylabel('order Quantity')
ax.set_title('Quantity Distribution')

plt.suptitle("")
plt.show()
```



In [ ]:
```
pd.DataFrame(data1['quantity'].describe())
```

Out[ ]:

	quantity
count	9489.000000
mean	9.933923
std	52.922847
min	1.000000
25%	1.000000
50%	3.000000
75%	10.000000
max	3114.000000

In [ ]:
```
pd.DataFrame(data1['quantity'].describe())
```

Out[ ]:

	quantity
count	9489.000000
mean	9.933923
std	52.922847
min	1.000000
25%	1.000000
50%	3.000000
75%	10.000000
max	3114.000000

In [ ]:
```
data1.loc[data1['quantity'] > 0].shape
```

Out[ ]:
```
(9489, 10)
```

In [ ]:
```
data1.shape
```

Out[ ]:
```
(9489, 10)
```

In [ ]:
```
data1 = data1.loc[data1['quantity'] > 0].shape
```

In [ ]:
```
# mengubah ke bulanan
```

In [ ]:
```
pesanan_bulanan_df = data1.set_index('order_date')['order_id'].resample('M').nunique()
pesanan_bulanan_df
```

Out[ ]:
```
order_date
2019-01-31    337
2019-02-28    548
2019-03-31    686
2019-04-30    105
2019-05-31     71
2019-06-30     90
2019-07-31    105
2019-08-31    117
2019-09-30     48
2019-10-31    105
2019-11-30    103
2019-12-31     57
Freq: M, Name: order_id, dtype: int64
```

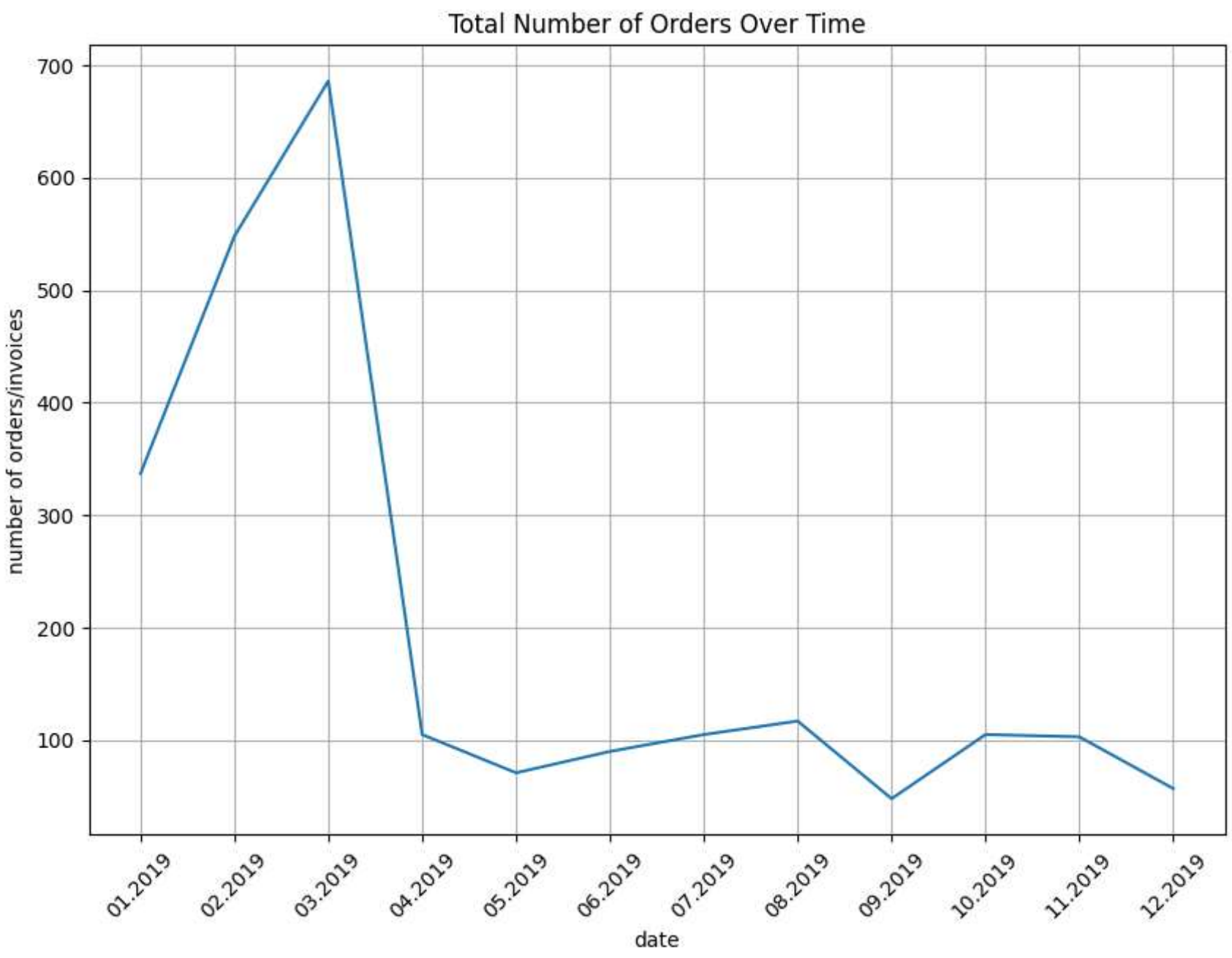
In [ ]:
```
ax = pd.DataFrame(pesanan_bulanan_df.values).plot(
    grid=True,
    figsize=(10,7),
    legend=False
)

ax.set_xlabel('date')
ax.set_ylabel('number of orders/invoices')
ax.set_title('Total Number of Orders Over Time')

plt.xticks(
    range(len(pesanan_bulanan_df.index)),
    [x.strftime('%m.%Y') for x in pesanan_bulanan_df.index],
    rotation = 45

)

plt.show()
```



TOTAL PENDAPATAN PERBULAN

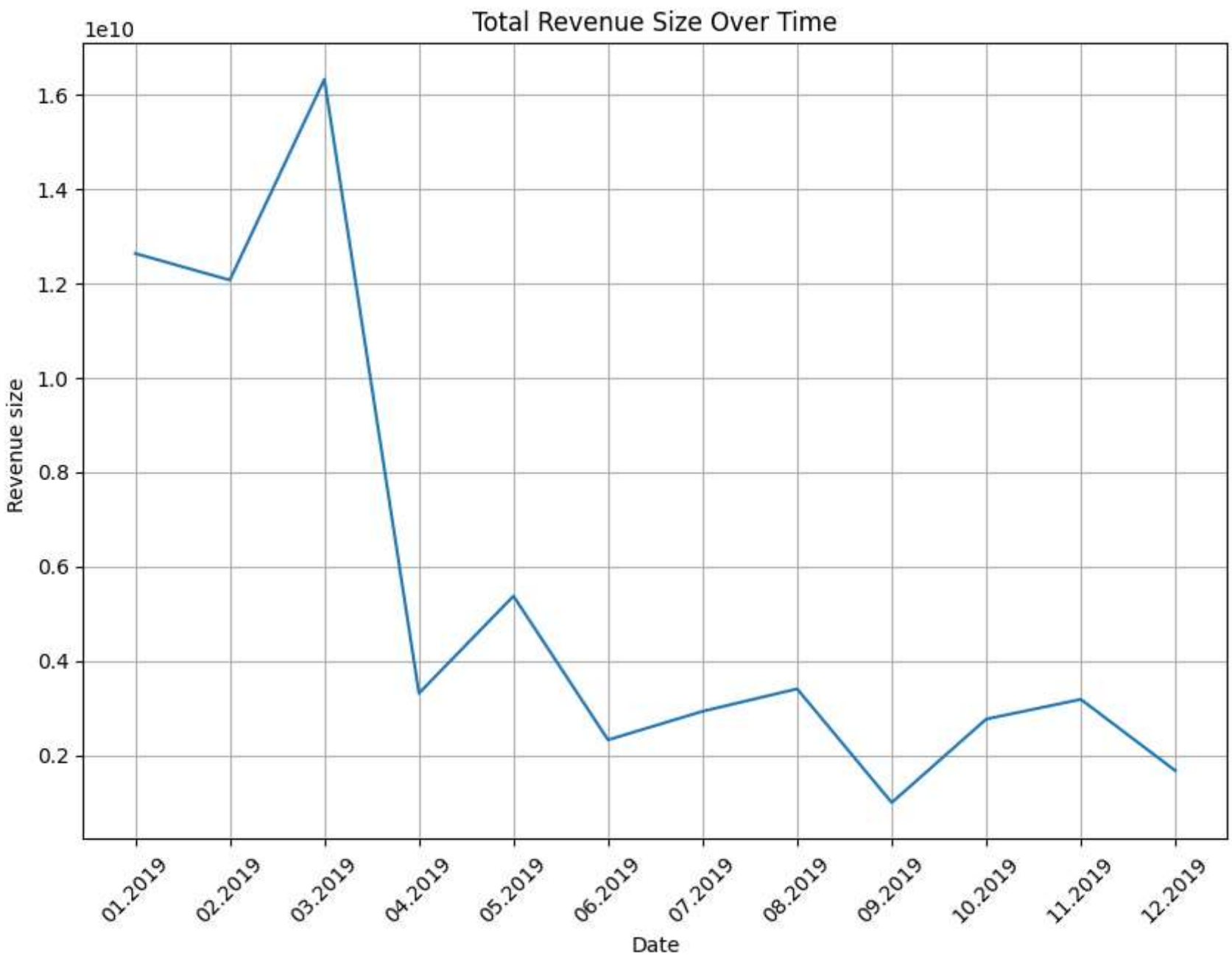
In [ ]:
```
pendapatan_perbulan = data1.set_index('order_date')['total_price'].resample('M').sum()
pendapatan_perbulan
```

Out[ ]:
```
order_date
2019-01-31    12635950000
2019-02-28    12072226000
2019-03-31    16326780000
2019-04-30     3313666000
2019-05-31     5374300000
2019-06-30     2325978000
2019-07-31     2931884000
2019-08-31     3407713000
2019-09-30      996156000
2019-10-31     2767135000
2019-11-30     3184579000
2019-12-31     1672642000
Freq: M, Name: total_price, dtype: int64
```

In [ ]:
```
ax = pd.DataFrame(pendapatan_perbulan.values).plot(
    grid=True,
    figsize= (10,7),
    legend = False
)

ax.set_ylabel('Revenue size ')
ax.set_title('Total Revenue Size Over Time')
ax.set_xlabel('Date')

plt.xticks(
    range(len(pendapatan_perbulan.index)),
    [x.strftime('%m.%Y') for x in pendapatan_perbulan.index],
    rotation = 45
)
plt.show()
```



Repear Order

In [ ]:
```
data1.head()
```

Out[ ]:

	order_id	order_date	customer_id	city	province	product_id	brand	quantity	item_price	total_price
0	1612885	2019-01-01	16293	Malang	Jawa Timur	P1301	BRAND_F	6	747000	4482000
1	1612387	2019-01-01	17228	Bogor	Jawa Barat	P2086	BRAND_L	4	590000	2360000
2	1612903	2019-01-01	16775	Surakarta	Jawa Tengah	P1656	BRAND_G	3	1325000	3975000
3	1612963	2019-01-01	0	unknown	unknown	P3127	BRAND_S	1	1045000	1045000
4	1612915	2019-01-01	0	unknown	unknown	P1230	BRAND_E	1	-891000	891000

In [ ]:
```
customer_repeat_data1 = data1.groupby(
    by =['order_date','order_id']
```

```
    ).agg({
      'total_price' : sum,
      'customer_id' : max,
      'province' : max
    }).reset_index()
```

In [ ]:
```
customer_repeat_data1
```

Out[ ]:

	order_date	order_id	total_price	customer_id	province
0	2019-01-01	1612372	23460000	17511	Banten
1	2019-01-01	1612378	3720000	17470	DKI Jakarta
2	2019-01-01	1612387	5900000	17228	Jawa Barat
3	2019-01-01	1612390	67129000	12681	Sulawesi Selatan
4	2019-01-01	1612393	40168000	14907	DKI Jakarta
...	...	...	...	...	...
2367	2019-12-03	1632223	45609000	14156	Kalimantan Tengah
2368	2019-12-03	1632226	21696000	14156	Kalimantan Tengah
2369	2019-12-03	1632232	15045000	14825	DKI Jakarta
2370	2019-12-03	1632262	12567000	17530	DKI Jakarta
2371	2019-12-03	1632268	51425000	0	unknown

2372 rows × 5 columns

In [ ]:
```
monthly_repeat_order_df = customer_repeat_data1.set_index('order_date').groupby([pd.Grouper(freq='M'), 'customer_id']).filter(lambda x: len
```

In [ ]:
```
monthly_repeat_order_df
```

Out[ ]:
```
order_date
2019-01-31    29
2019-02-28    42
2019-03-31    77
2019-04-30     6
2019-05-31     5
2019-06-30     3
2019-07-31     6
2019-08-31     6
2019-09-30     0
2019-10-31     6
2019-11-30     5
2019-12-31     3
Freq: M, Name: customer_id, dtype: int64
```

In [ ]:
```
monthly_unique_order = data1.set_index('order_date')['customer_id'].resample('M').nunique()
monthly_unique_order
```

Out[ ]:
```
order_date
2019-01-31    239
2019-02-28    443
2019-03-31    524
2019-04-30     86
2019-05-31     59
2019-06-30     80
2019-07-31     95
2019-08-31     91
2019-09-30     48
2019-10-31     97
2019-11-30     87
2019-12-31     45
Freq: M, Name: customer_id, dtype: int64
```

In [ ]:
```
presentase_repeat_order = monthly_repeat_order_df/monthly_unique_order*100.0
presentase_repeat_order
```

Out[ ]:
```
order_date
2019-01-31    12.133891
2019-02-28     9.480813
2019-03-31    14.694656
2019-04-30     6.976744
2019-05-31     8.474576
2019-06-30     3.750000
2019-07-31     6.315789
2019-08-31     6.593407
2019-09-30     0.000000
2019-10-31     6.185567
2019-11-30     5.747126
2019-12-31     6.666667
Freq: M, Name: customer_id, dtype: float64
```

In [ ]:
```
ax = pd.DataFrame(monthly_repeat_order_df.values).plot(
    figsize = (25,10)
```

```
)
pd.DataFrame(monthly_unique_order.values).plot(
    ax = ax,
    grid=True
)

ax2 = pd.DataFrame(presentase_repeat_order.values).plot.bar(
    ax =ax,
    grid = True,
    secondary_y = True,
    color = 'green',
    alpha = 0.2
)


ax.set_xlabel('date')
ax.set_ylabel('number of customers')
ax.set_title('Number of All vs. Repeat Customers Over Time')

ax2.set_ylabel('percentage (%)')

ax.legend(['Repeat Customers', 'All Customers'])
ax2.legend(['Percentage of Repeat'], loc='upper right')

ax.set_ylim([0, monthly_unique_order.values.max()+100])
ax2.set_ylim([0, 100])

plt.xticks(
    range(len(monthly_repeat_order_df.index)),
    [x.strftime('%m.%Y') for x in monthly_repeat_order_df.index],
    rotation=45
)

plt.show()
```