In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import copy
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")
```

In [2]:

```
df=pd.read_csv("dataset_group.csv")
```

In [3]:

```
df.head()
```

Out[3]:

	Date	Order_id	Product
0	**2018-01-01**	**1**	**yogurt**
1	**2018-01-01**	**1**	**pork**
2	**2018-01-01**	**1**	**sandwich bags**
3	**2018-01-01**	**1**	**lunch meat**
4	**2018-01-01**	**1**	**all- purpose**

In [4]:

```
df.columns
```

Out[4]:

```
Index(['Date', 'Order_id', 'Product'], dtype='object')
```

In [5]:

```
df.shape
```

Out[5]:

```
(20641, 3)
```

In [6]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20641 entries, 0 to 20640
Data columns (total 3 columns):
 #   Column    Non-Null Count  Dtype 
---  ------    --------------  ----- 
 0   Date      20641 non-null  object
 1   Order_id  20641 non-null  int64 
 2   Product   20641 non-null  object
dtypes: int64(1), object(2)
memory usage: 483.9+ KB
```

In [7]:

```
df.describe().T
```

Out[7]:

	count	mean	std	min	25%	50%	75%	max

	count	mean	std	min	25%	50%	75%	max
Order_id	20641.0	575.986289	328.557078	1.0	292.0	581.0	862.0	1139.0

In [8]:

```
df.describe(include='all').T
```

Out[8]:

	count	unique	top	freq	mean	std	min	25%	50%	75%	max
Date	20641	603	2019-02-08	183	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Order_id	20641	NaN	NaN	NaN	575.986	328.557	1	292	581	862	1139
Product	20641	37	poultry	640	NaN	NaN	NaN	NaN	NaN	NaN	NaN

In [9]:

```
df.duplicated().sum()
```

Out[9]:

```
4730
```

In [10]:

```
df.isnull().sum()
```

Out[10]:

```
Date        0
Order_id    0
Product     0
dtype: int64
```

In [11]:

```
cat_df = df.select_dtypes(include=['object']).copy()
```

In [12]:

```
cat_df.head()
```

Out[12]:

	Date	Product
0	2018-01-01	yogurt
1	2018-01-01	pork
2	2018-01-01	sandwich bags
3	2018-01-01	lunch meat
4	2018-01-01	all- purpose

In [13]:

```
cat=[]
num=[]
for i in df.columns:
    if df[i].dtype=="object":
        cat.append(i)
    else:
        num.append(i)
print(cat)
print(num)
```

```
['Date', 'Product']
['Order_id']
```

In [14]:

In [14]:

```
for column in df.columns:
    if df[column].dtype == 'object':
        print(column.upper(),': ',df[column].nunique())
        print(df[column].value_counts().sort_values())
        print('\n')
```

```
DATE :  603
2020-02-26      3
2018-09-24      4
2019-09-05      4
2019-03-11      4
2018-03-18      5
              ...
2018-05-17    123
2018-03-01    127
2018-03-06    134
2019-02-20    146
2019-02-08    183
Name: Date, Length: 603, dtype: int64


PRODUCT :  37
hand soap                       502
sandwich loaves                 523
fruits                          529
pork                            531
sugar                           533
sandwich bags                   536
spaghetti sauce                 536
pasta                           542
laundry detergent               542
tortillas                       543
individual meals                544
yogurt                          545
ketchup                         548
dishwashing liquid/detergent    551
all- purpose                    551
mixes                           554
milk                            555
butter                          555
flour                           555
paper towels                    556
beef                            561
shampoo                         562
coffee/tea                      565
aluminum foil                   566
dinner rolls                    567
toilet paper                    569
eggs                            570
juice                           570
bagels                          573
lunch meat                      573
soap                            574
waffles                         575
cheeses                         578
ice cream                       579
cereals                         591
soda                            597
poultry                         640
Name: Product, dtype: int64
```

In [15]:

```
print(cat_df.isnull().values.sum())
```

```
0
```

In [16]:

```
print(cat_df['Product'].value_counts())
```

```
poultry                          640
soda                             597
cereals                          591
ice cream                        579
cheeses                          578
waffles                          575
soap                             574
lunch meat                       573
bagels                           573
juice                            570
eggs                             570
toilet paper                     569
dinner rolls                     567
aluminum foil                    566
coffee/tea                       565
shampoo                          562
beef                             561
paper towels                     556
flour                            555
milk                             555
butter                           555
mixes                            554
all- purpose                     551
dishwashing liquid/detergent     551
ketchup                          548
yogurt                           545
individual meals                 544
tortillas                        543
laundry detergent                542
pasta                            542
spaghetti sauce                  536
sandwich bags                    536
sugar                            533
pork                             531
fruits                           529
sandwich loaves                  523
hand soap                        502
Name: Product, dtype: int64
```

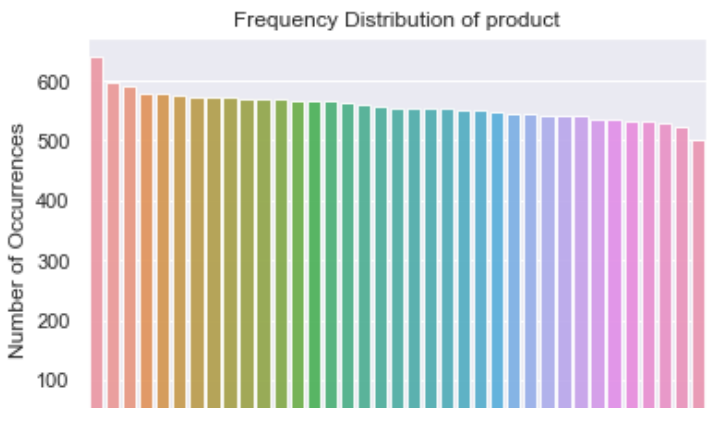
In [17]:

```
print(cat_df['Product'].value_counts().count())
```

```
37
```

In [18]:

```
import seaborn as sns
import matplotlib.pyplot as plt
product_count = cat_df['Product'].value_counts()
sns.set(style="darkgrid")
sns.barplot(product_count.index, product_count.values, alpha=0.9)
plt.title('Frequency Distribution of product')
plt.ylabel('Number of Occurrences', fontsize=12)
plt.xlabel('Product', fontsize=12)
plt.show()
```

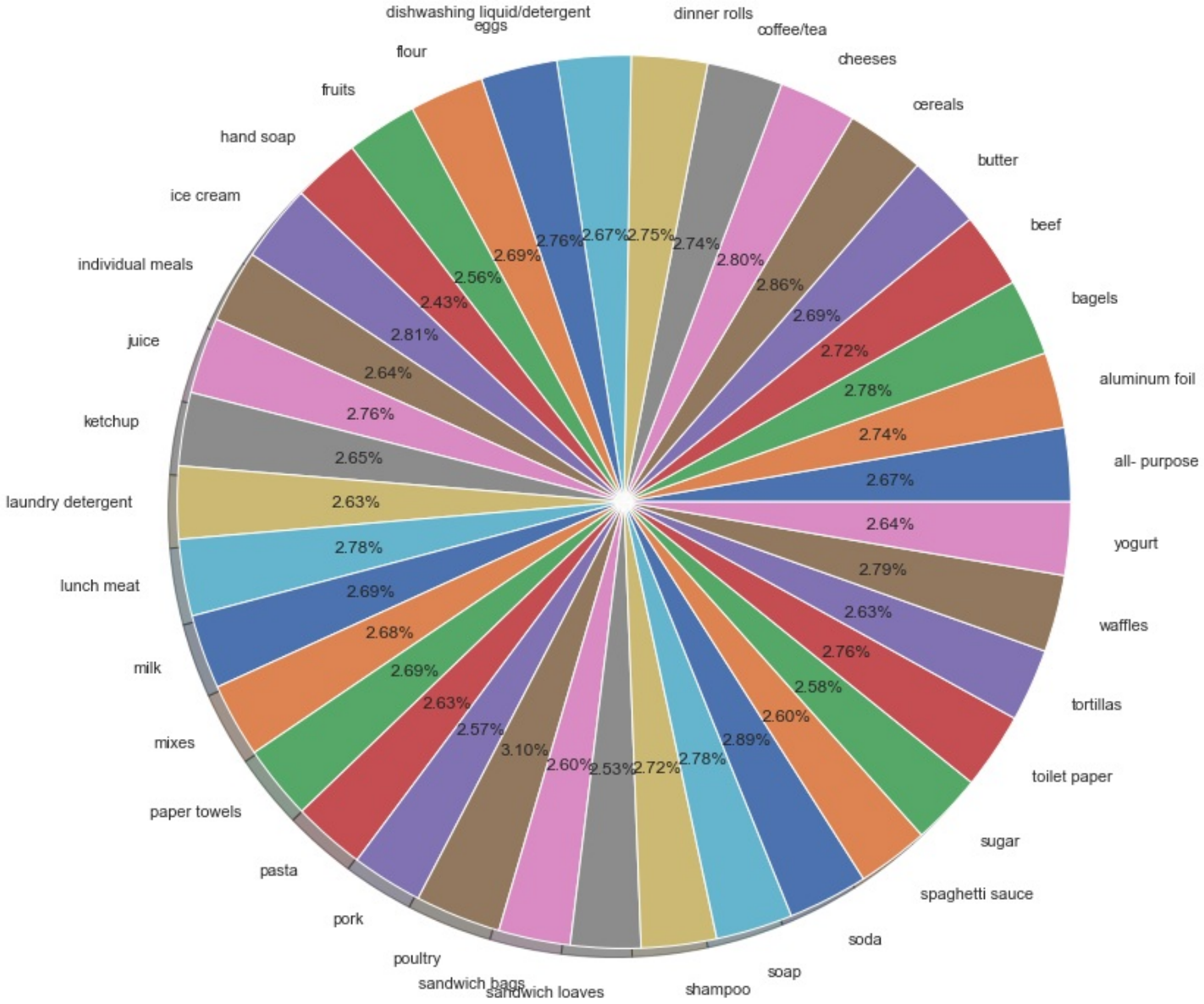


In [19]:

```
labels = cat_df['Product'].astype('category').cat.categories.tolist()
```

In [20]:

```
labels = cat_df['Product'].astype('category').cat.categories.tolist()
counts = cat_df['Product'].value_counts()
sizes = [counts[var_cat] for var_cat in labels]
fig1, ax1 = plt.subplots(figsize=(12,15))
ax1.pie(sizes, labels=labels, autopct='%1.2f%%', shadow=True) #autopct is show the % on plot
ax1.axis('equal')
plt.show()
```

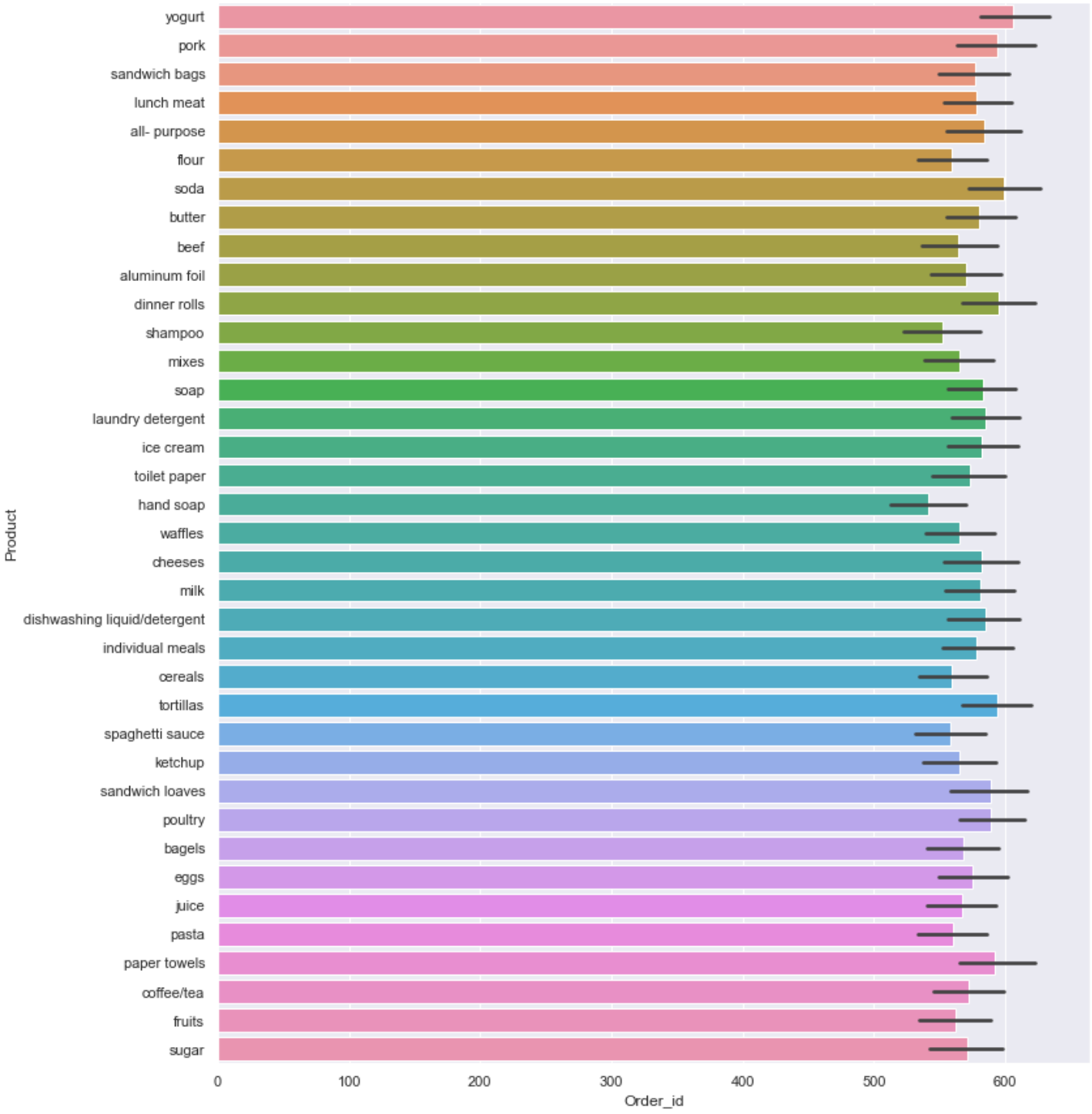


The piechart denotes the percentage of sales of each product over a span of 3 years.

In [ ]:

In [21]:

```
plt.figure(figsize=(12,15))
sns.barplot(x=df['Order_id'], y=df['Product'], orient='h')
plt.show()
```



The barplot denotes the frequency of each product sold in 3 years. The maximum no.of items sold are poultry product and least sold products are hand soaps.

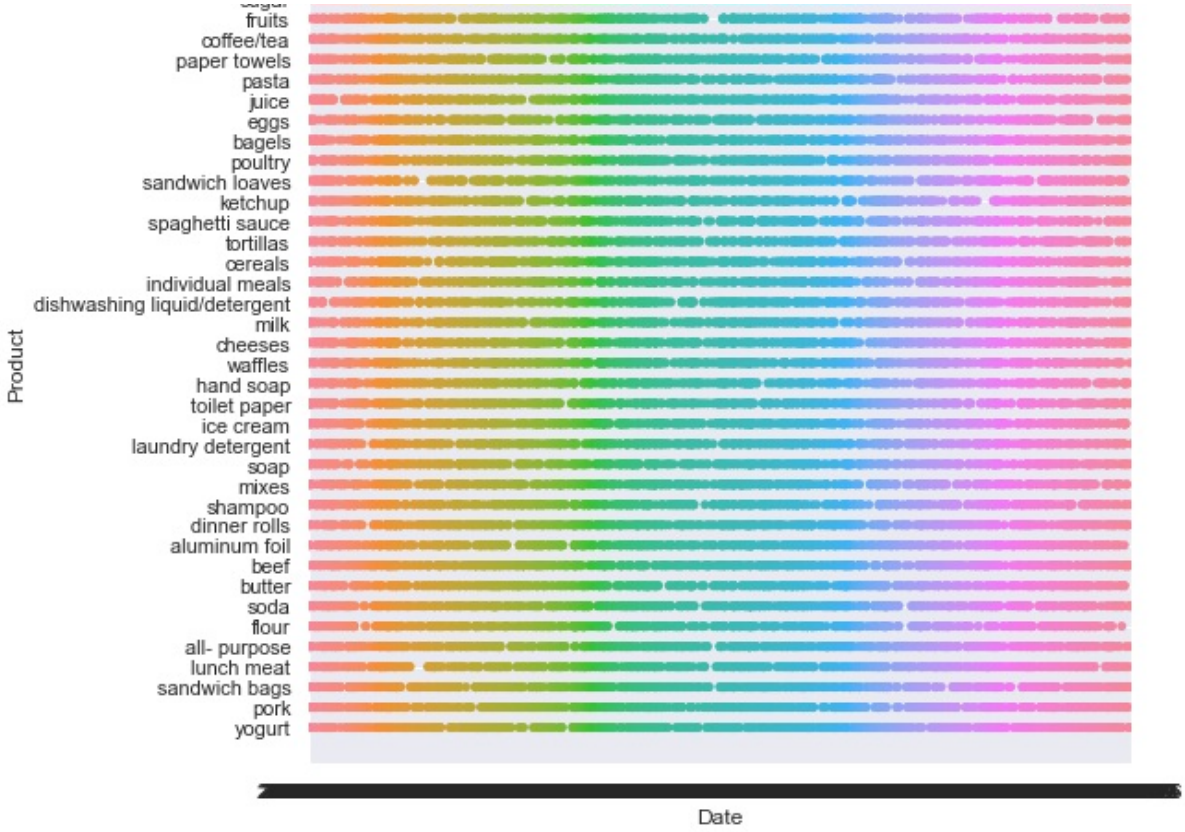
In [ ]:

```

```

In [22]:

```
plt.figure(figsize=(8,8))
sns.stripplot(df["Date"], df['Product'], jitter=True)
plt.show()
```



The above stripplot compares the sales of different items across 3 years.

In [ ]:

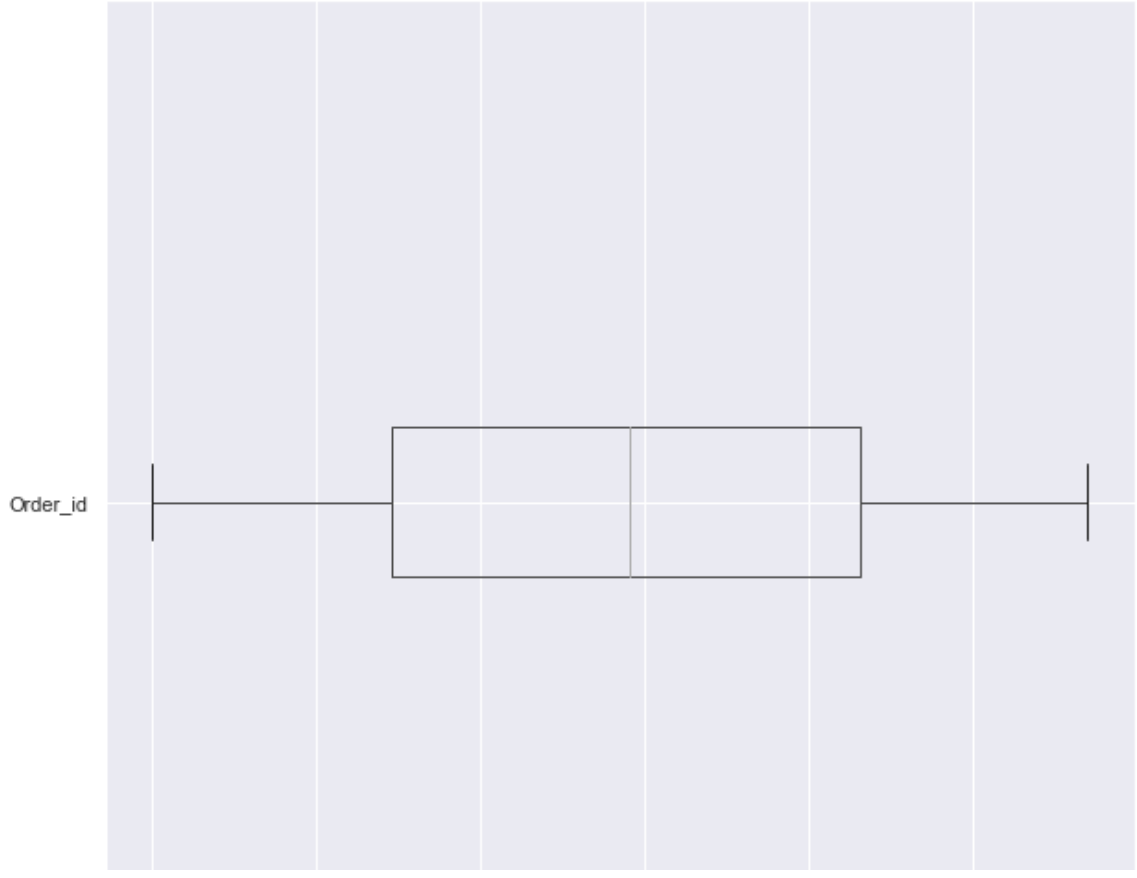
```
```

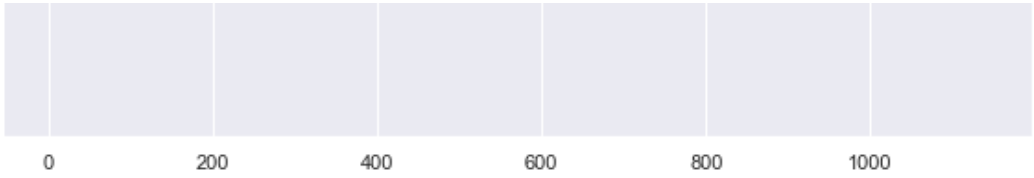
In [23]:

```
plt.figure(figsize=(10,10))
df[num].boxplot(vert=0)
```

Out[23]:

```
<AxesSubplot:>
```





In [24]:

```
dft = pd.read_csv("dataset_group.csv",parse_dates=True,squeeze=True,index_col=0)
```

In [25]:

```
dft.head()
```

Out[25]:

Date	Order_id	Product
2018-01-01	1	yogurt
2018-01-01	1	pork
2018-01-01	1	sandwich bags
2018-01-01	1	lunch meat
2018-01-01	1	all- purpose

In [26]:

```
dft.tail()
```

Out[26]:

Date	Order_id	Product
2020-02-25	1138	soda
2020-02-25	1138	paper towels
2020-02-26	1139	soda
2020-02-26	1139	laundry detergent
2020-02-26	1139	shampoo

In [27]:

```
dft.plot();
plt.grid()
```

Weekly Plot

In [28]:

```
df_daily_sum = dft.resample('D').sum()
df_daily_sum
```

Out[28]:

Date	Order_id
2018-01-01	59
2018-01-02	367
2018-01-03	154
2018-01-04	109
2018-01-05	627
...	...
2020-02-22	40854
2020-02-23	22720
2020-02-24	26151
2020-02-25	21622
2020-02-26	3417

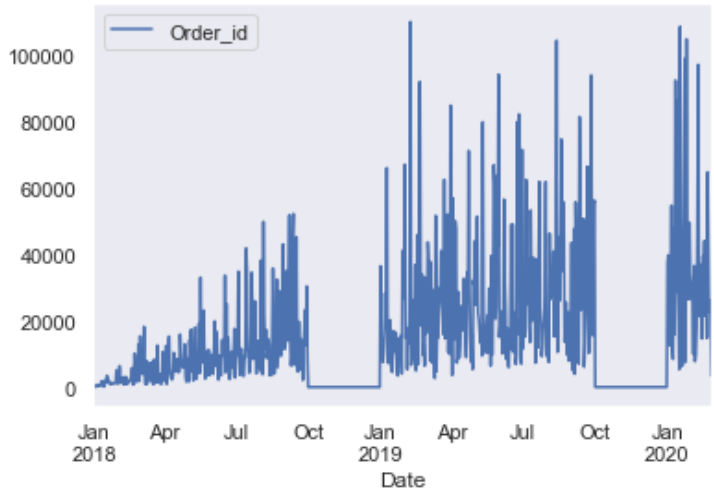
787 rows × 1 columns

The values which the original series cannot provide is taken as 0 by python if we try to resample the data on a daily basis.

In [29]:

```
plt.figure(figsize=(30,20))
df_daily_sum.plot()
plt.grid();
```

```
<Figure size 2160x1440 with 0 Axes>
```



In [30]:

```
#In 2018, there was a rapid decline in sales.
# From 2019 daily sales got increased compared to 2018.
```

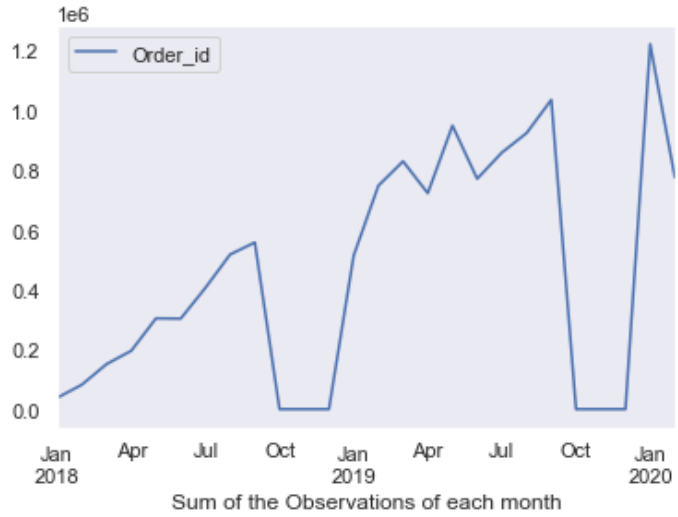
Monthly Plot

fig, ax = plt.subplots(figsize=(22,5)) sns.boxplot(dft.index.month, dft, ax=ax,whis=1.5) plt.grid();

In [31]:

```
df_monthly_sum = dft.resample('M').sum()
df_monthly_sum.head()

df_monthly_sum.plot();
plt.grid()
plt.xlabel('Sum of the Observations of each month');
```



In [32]:

```
# The sales increase till Jul and rapidly decrease by October. The most profitable month is August.
# The least profitable months are October, November and December.
```

In [33]:

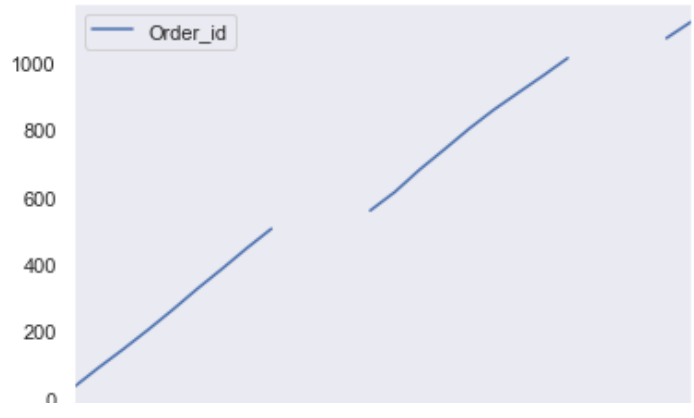
```
df_monthly_mean = dft.resample('M').mean()
df_monthly_mean.head()
```

Out[33]:

	Order_id
Date	
2018-01-31	**32.235897**
2018-02-28	**90.245865**
2018-03-31	**145.322083**
2018-04-30	**202.875260**
2018-05-31	**262.793939**

In [34]:

```
df_monthly_mean.plot();
plt.grid()
plt.xlabel('Mean of the Observations of each month');
```



Mean of the Observations of each month

Quarterly Plot

In [35]:

```
df_quarterly_sum = dft.resample('Q').sum()
df_quarterly_sum.head()
```

Out[35]:

Date	Order_id
2018-03-31	270268
2018-06-30	801291
2018-09-30	1479807
2018-12-31	0
2019-03-31	2088497

In [36]:

```
plt.figure(figsize=(15,10))
df_quarterly_sum.plot();
plt.grid()
```

```
<Figure size 1080x720 with 0 Axes>
```

The sales follow decreasing trend in Q4

Sales increase in Q1 and Q2 but the net effect considering all the quarters is still a loss in sales for the company in Q4

In [ ]:

```

```

In [37]:

```
df_quarterly_mean = dft.resample('Q').mean()
df_quarterly_mean.head()
```

Out[37]:

	Order_id
Date	
2018-03-31	86.791265
2018-06-30	262.977027
2018-09-30	445.993671
2018-12-31	NaN
2019-03-31	622.688432

In [38]:

```
df_quarterly_mean.plot();
plt.grid()
```

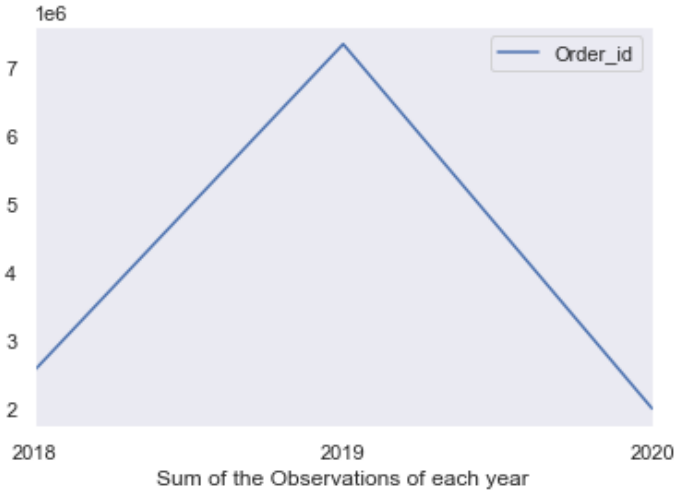
YEARLY PLOT

fig, ax = plt.subplots(figsize=(22,8)) sns.boxplot(dft.index.year, dft, ax=ax,whis=1.5) plt.grid(); plt.xlabel('Years'); plt.ylabel('Yearly Sales Variation');

In [39]:

```
df_yearly_sum = dft.resample('A').sum()
df_yearly_sum.head()

df_yearly_sum.plot();
plt.grid()
plt.xlabel('Sum of the Observations of each year');
```



In [40]:

```
# It is observed that sales are maximum in 2019 and sales drop drastically by 2020.
```

```
#So, market basket analysis of data may provide recommendations for higher profits.
```

In [ ]:

```

```

In [41]:

```
df_yearly_mean = dft.resample('Y').mean()
df_yearly_mean.head()
```

Out[41]:

Date	Order_id
2018-12-31	269.159827
2019-12-31	786.761277
2020-12-31	1090.609076

In [42]:

```
df_yearly_mean.plot();
plt.grid()
plt.xlabel('Mean of the Observations of each year');
```