

	Total	130106.0	46.139824	51.908603	0.0	0.0	50.00	71.1	1179.00
	Taxes	130106.0	9.189267	9.005918	0.0	0.0	9.83	12.6	196.51

df4["Lineitem name"].unique()

```
array(['Never Grow Up Compression Leggings - XS/8 / Purple',
      'Ironman Compression Leggings - Small / UK10 / Red',
      'Skinny Outdoor Trousers Black - 28W / UK10 / Black', ...,
      'MAX Stretch Skinny Outdoor Trousers Bottle Green - 24L / US8 / Bottle Green',
      'Mid-Weight MAX Stretch Skinny Outdoor Trousers Raspberry - 32P / UK14 / Raspberry',
      'Cargo MAX Stretch Outdoor Trousers Sand - 32L / UK14 / Sand'],
      dtype=object)
```

df4["Payment Method"].unique()





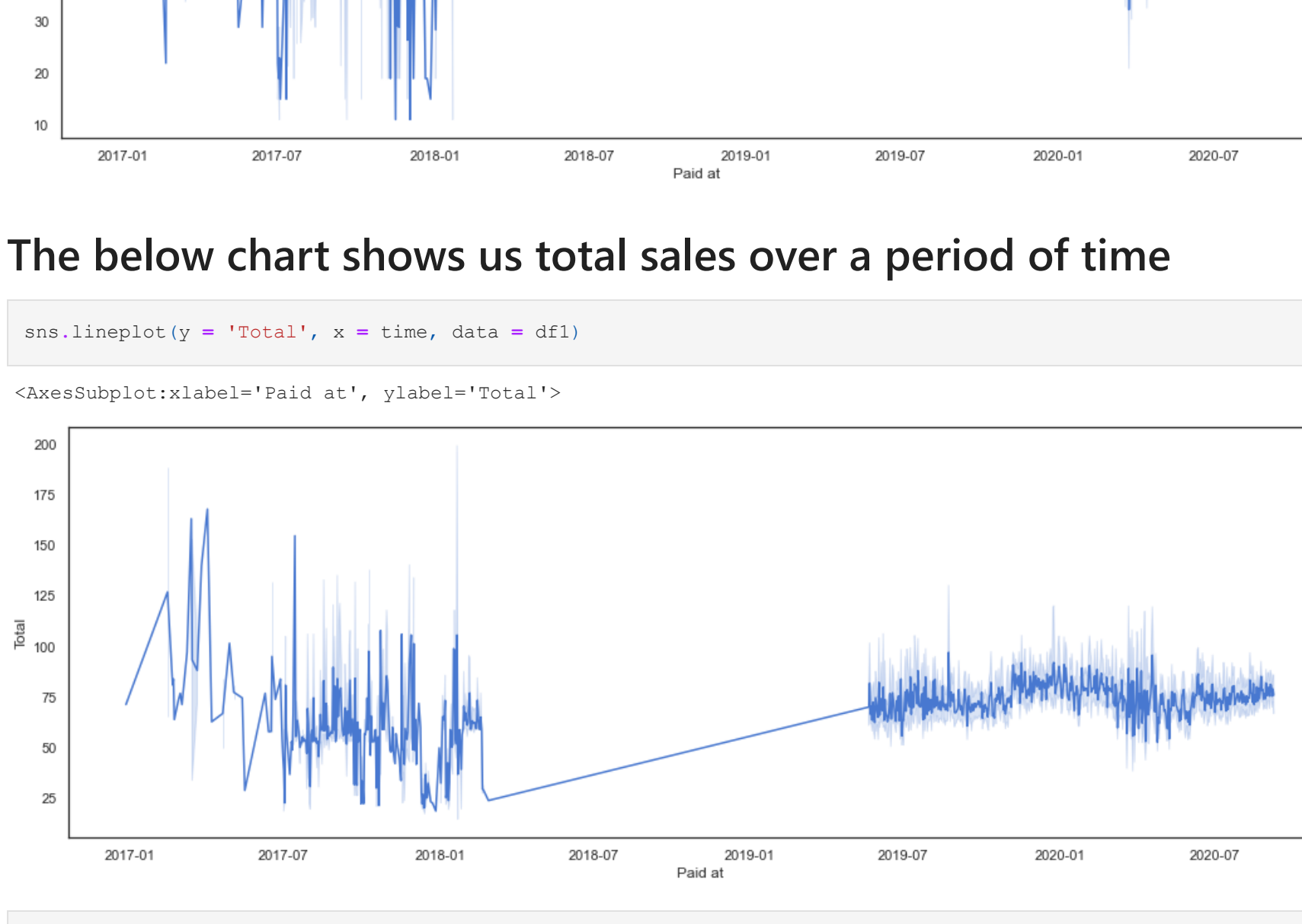


```
457 0 2018-02-21 00:00:00+00:00
458 1 2018-02-21 00:00:00+00:00
459 2 2018-02-21 00:00:00+00:00
460 3 2018-02-21 00:00:00+00:00
461 4 2018-02-21 00:00:00+00:00
462 ...
463 2019-05-21 00:00:00+00:00
464 88449 2019-05-21 00:00:00+00:00
465 88451 2019-05-21 00:00:00+00:00
466 88452 2019-05-21 00:00:00+00:00
467 88453 2019-05-21 00:00:00+00:00
468 88454 2019-05-21 00:00:00+00:00
469 88455 2019-05-21 00:00:00+00:00
470 Name: Paid at, Length: 44247, dtype: datetime64[ns, UTC]
```

## The belowchart tells about the fluctuation in line item price over a period of time

```
In [505]: sns.lineplot(y = 'Lineitem price', x = time, data = df1)

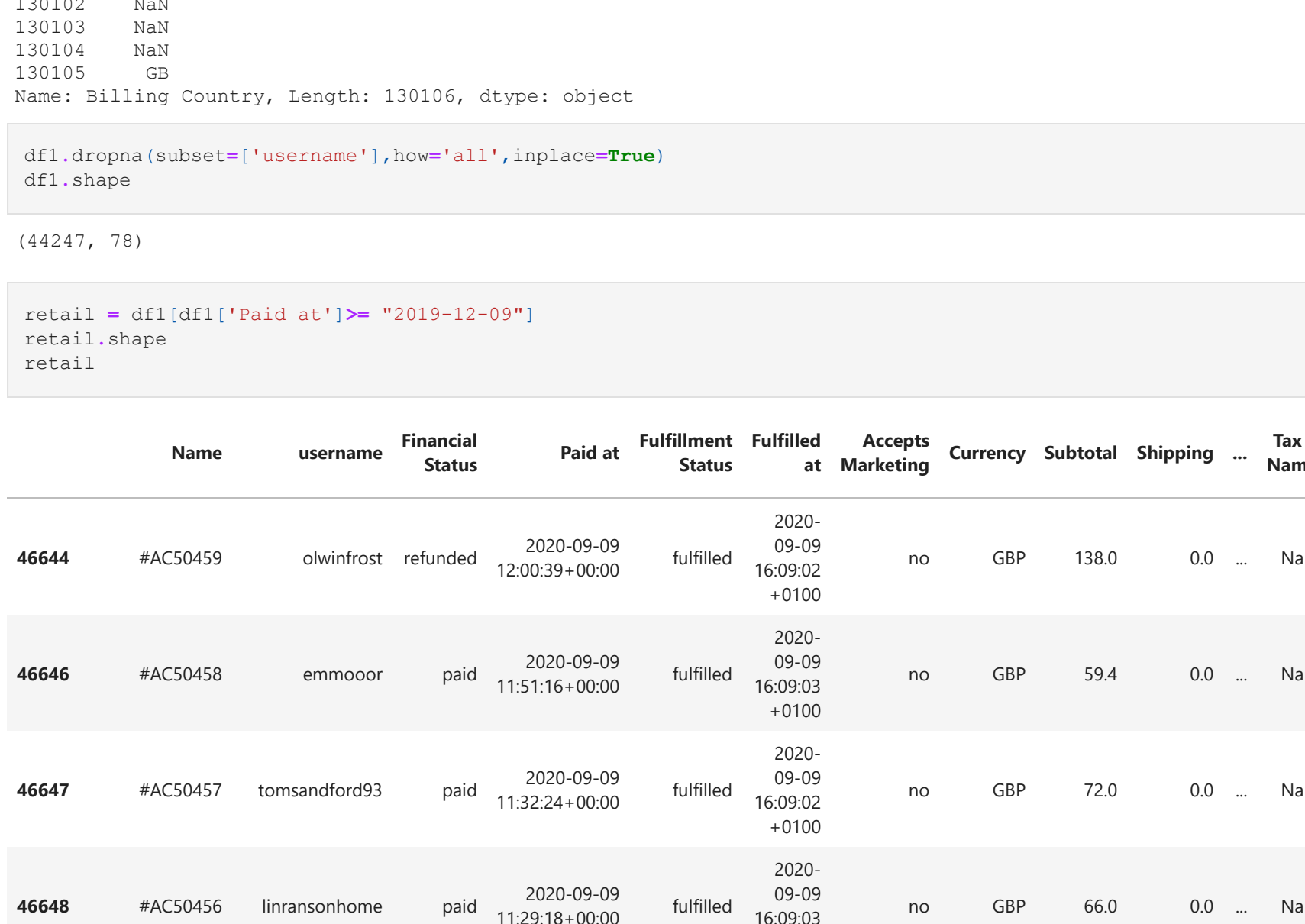
Out[505]: <AxesSubplot:label='Paid at', ylabel='Lineitem price'>
```



## The below chart shows us total sales over a period of time

```
In [506]: sns.lineplot(y = 'Total', x = time, data = df1)

Out[506]: <AxesSubplot:label='Paid at', ylabel='Total'>
```



```
In [376]: retail = df1[['Billing Country']]
          retail.shape
          retail
```

```
Out[376]: 0      US
          1      US
          2      GB
          3      US
          4      US
          ...
          130101 NaN
          130102 NaN
          130103 NaN
          130104 NaN
          130105 GB
          Name: Billing Country, Length: 130106, dtype: object
```

```
In [379]: df1.dropna(subset=['username'],how='all',inplace=True)
          df1.shape
```

```
Out[379]: (44247, 78)
```

```
In [383]: retail = df1[df1['Paid at']>= "2019-12-09"]
          retail.shape
          retail
```

```
Out[383]:
```

	Name	username	Financial Status	Paid at	Fulfillment Status	Fulfilled at	Accepts Marketing	Currency	Subtotal	Shipping	...	Tax 5 Name	Tax 5 Value
46644	#AC50459	olwinfro	refunded	2020-09-09 12:00:39+00:00	fulfilled	2020-09-09 16:09:02+01:00	no	GBP	138.0	0.0	...	NaN	NaN
46646	#AC50458	emmoor	paid	2020-09-09 11:51:16+00:00	fulfilled	2020-09-09 16:09:03+01:00	no	GBP	59.4	0.0	...	NaN	NaN
46647	#AC50457	tomsandfor93	paid	2020-09-09 11:32:24+00:00	fulfilled	2020-09-09 16:09:02+01:00	no	GBP	72.0	0.0	...	NaN	NaN
46648	#AC50456	linransonhome	paid	2020-09-09 11:29:18+00:00	fulfilled	2020-09-09 16:09:03+01:00	no	GBP	66.0	0.0	...	NaN	NaN
46649	#AC50455	tamsincrowson88	paid	2020-09-09 11:21:59+00:00	fulfilled	2020-09-09 16:09:04+01:00	no	GBP	66.0	0.0	...	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...
73964	#AC30406	lmarsden9	paid	2019-12-09 08:47:55+00:00	fulfilled	2019-12-09 15:34:57+00:00	no	GBP	62.1	0.0	...	NaN	NaN
73966	#AC30404	rkimbell3	paid	2019-12-09 08:18:28+00:00	fulfilled	2019-12-09 15:34:59+00:00	no	GBP	72.0	0.0	...	NaN	NaN
73967	#AC30403	gloaming	paid	2019-12-09 08:09:05+00:00	fulfilled	2019-12-09 15:35:00+00:00	no	GBP	26.1	0.0	...	NaN	NaN
73968	#AC30402	dave_leather	paid	2019-12-09 07:56:48+00:00	fulfilled	2019-12-09 15:34:59+00:00	yes	GBP	68.0	0.0	...	NaN	NaN
73970	G6477990878GB	crawford01	paid	2019-12-09 04:04:13+00:00	fulfilled	2019-12-09 10:54:36+00:00	no	GBP	93.1	0.0	...	NaN	NaN

16063 rows x 78 columns

```
In [386]: print("Summary..")
          #exploring the unique values of each attribute
          print("Number of transactions: ", retail['Name'].nunique())
          print("Number of products bought: ", retail['Lineitem name'].nunique())
          print("Number of customers: ", retail['username'].nunique())
          print("Percentage of customers NaN: ", round(retail['username'].isnull().sum() * 100 / len(df1),2),"%")

          Summary..
          Number of transactions: 16063
          Number of products bought: 757
          Number of customers: 12868
          Percentage of customers NaN:  0.9 %
```

```
In [392]: retail['Paid at'].max()

Out[392]: Timestamp('2020-09-09 12:00:39+0000', tz='UTC')
```

```
In [398]: now = dt.date(2020,12,9)
          print(now)
```

```
Out[398]: 2020-12-09
```

```
In [436]: #create a new column called date which contains the date of invoice only
          retail['Paid at'] = pd.DatetimeIndex(retail['Paid at']).date
```

```
In [435]: retail.head()
```

	Name	username	Financial Status	Paid at	Fulfillment Status	Fulfilled at	Accepts Marketing	Currency	Subtotal	Shipping	...	Tax 5 Name	Tax 5 Value
46644	#AC50459	olwinfro	refunded	2020-09-09	fulfilled	2020-09-09 16:09:02+01:00	no	GBP	138.0	0.0	...	NaN	NaN
46646	#AC50458	emmoor	paid	2020-09-09	fulfilled	2020-09-09 16:09:03+01:00	no	GBP	59.4	0.0	...	NaN	NaN
46647	#AC50457	tomsandford93	paid	2020-09-09	fulfilled	2020-09-09 16:09:02+01:00	no	GBP	72.0	0.0	...	NaN	NaN
46648	#AC50456	linransonhome	paid	2020-09-09	fulfilled	2020-09-09 16:09:03+01:00	no	GBP	66.0	0.0	...	NaN	NaN
46649	#AC50455	tamsincrowson88	paid	2020-09-09	fulfilled	2020-09-09 16:09:04+01:00	no	GBP	66.0	0.0	...	NaN	NaN

5 rows x 78 columns

```
In [472]: recency = retail.groupby(by = ['username','lineitem name'], as_index=False)['Paid at'].max()
          recency.columns = ['username','Item name','Fulfilled at']
          recency.reset_index()
```

	username	Item name	Fulfilled at
0	018503	Thermal Skinny Outdoor Trousers Black - 26L / ...	2020-04-07
1	0902179	MAX Stretch Skinny Outdoor Trousers Navy - 28R...	2020-08-25
2	09616007keg	MAX Stretch Skinny Outdoor Trousers Black - 28L...	2020-04-27
3	09sharonlee	GRANITE Mid-weight Outdoor Trousers Grey - 32L...	2020-01-26
4	0megannmitche0	MAX Stretch Skinny Outdoor Trousers Black - 28L...	2020-04-24

## calculate recency at item level

```
In [483]: recency['Recency'] = recency['Fulfilled at'].apply(lambda x: (now - x).days)
          recency.reset_index()
```

	index	username	Item name	Fulfilled at	Recency
0	0	018503	Thermal Skinny Outdoor Trousers Black - 26L / ...	2020-04-07	246
1	1	0902179	MAX Stretch Skinny Outdoor Trousers Navy - 28R...	2020-08-25	106
2	2	09616007keg	MAX Stretch Skinny Outdoor Trousers Black - 28L...	2020-04-27	226
3	3	09sharonlee	GRANITE Mid-weight Outdoor Trousers Grey - 32L...	2020-01-26	318
4	4	0megannmitche0	MAX Stretch Skinny Outdoor Trousers Black - 28L...	2020-04-24	229
...	...	...	...	...	...
15969	15969	ztbraham	Chino Slim Fit Roll Up Trousers Khaki - 28 / U...	2020-04-23	230
15970	15970	zulu1520	Thermal Skinny Outdoor Trousers Black - 28P / ...	2020-03-02	282
15971	15971	zuldawn388	Thermal Skinny Outdoor Trousers Black - 28L / ...	2020-02-01	312
15972	15972	zus_	Casual Stroll Pants Black - 30R / UK12 / Black	2020-08-24	107
15973	15973	zuzana.schunova	Thermal Outdoor Leggings - XSmall / UK8 / Black	2020-01-03	341

15974 rows x 5 columns

## Frequency

Frequency helps us to know how many times a customer purchased from us. To do that we need to check how many invoices are registered by the same customer

```
In [484]: retail_copy = retail
          retail_copy.drop_duplicates(subset=['username','lineitem quantity', 'Name'], keep="first", inplace=True)
          #calculate frequency of purchases
          frequency = retail_copy.groupby(by=['username','lineitem name'], as_index=False)['lineitem quantity'].count()
          frequency.columns = ['username','Item name','Frequency']
          frequency.reset_index()
```

	index	username	Item name	Frequency
0	0	018503	Thermal Skinny Outdoor Trousers Black - 26L / ...	1
1	1	0902179	MAX Stretch Skinny Outdoor Trousers Navy - 28R...	1
2	2	09616007keg	MAX Stretch Skinny Outdoor Trousers Black - 28L...	1
3	3	09sharonlee	GRANITE Mid-weight Outdoor Trousers Grey - 32L...	1
4	4	0megannmitche0	MAX Stretch Skinny Outdoor Trousers Black - 28L...	1
...	...	...	...	...
15969	15969	ztbraham	Chino Slim Fit Roll Up Trousers Khaki - 28 / U...	1
15970	15970	zulu1520	Thermal Skinny Outdoor Trousers Black - 28P / ...	1
15971	15971	zuldawn388	Thermal Skinny Outdoor Trousers Black - 28L / ...	1
15972	15972	zus_	Casual Stroll Pants Black - 30R / UK12 / Black	1
15973	15973	zuzana.schunova	Thermal Outdoor Leggings - XSmall / UK8 / Black	1

15974 rows x 4 columns

```
In [485]: #value count for frequency
          frequency['Frequency'].value_counts()
```

```
Out[485]: 1    15888
          2         83
          3          3
          Name: Frequency, dtype: int64
```

## Monetary

Monetary attribute answers the question: How much money did the customer spent over time?

```
In [491]: retail['Total'] = retail['lineitem quantity'] * retail['lineitem price']
```

```
In [488]: monetary_df = retail.groupby(by=['username','lineitem name'],as_index=False).agg({'Total': 'sum'})
          monetary_df.columns = ['username','Item name','Monetary']
          monetary_df.reset_index()
```

	index	username	Item name	Monetary
0	0	018503	Thermal Skinny Outdoor Trousers Black - 26L / ...	79.0
1	1	0902179	MAX Stretch Skinny Outdoor Trousers Navy - 28R...	66.0
2	2	09616007keg	MAX Stretch Skinny Outdoor Trousers Black - 28L...	66.0
3	3	09sharonlee	GRANITE Mid-weight Outdoor Trousers Grey - 32L...	66.0
4	4	0megannmitche0	MAX Stretch Skinny Outdoor Trousers Black - 28L...	66.0
...	...	...	...	...
15969	15969	ztbraham	Chino Slim Fit Roll Up Trousers Khaki - 28 / U...	40.0
15970	15970	zulu1520	Thermal Skinny Outdoor Trousers Black - 28P / ...	79.0
15971	15971	zuldawn388	Thermal Skinny Outdoor Trousers Black - 28L / ...	79.0
15972	15972	zus_	Casual Stroll Pants Black - 30R / UK12 / Black	72.0
15973	15973	zuzana.schunova	Thermal Outdoor Leggings - XSmall / UK8 / Black	45.0

15974 rows x 4 columns

```
In [489]: #value count for frequency
          frequency['Frequency'].value_counts()
```

```
Out[489]: 1    15888
          2         83
          3          3
          Name: Frequency, dtype: int64
```

## concating all the Recency, Frequency and Monetary Columns

```
In [491]: #rfm_df = recency,frequency,monetary_df
          #rfm_df

          #rf_1 = recency.join(frequency , on = ['username','Item name'], how = "outer")
          #rf_1.head()

          rf_1 = pd.merge(recency,frequency,on = ['username','Item name'])
          rfm_df = pd.merge(rf_1,monetary_df,on = ['username','Item name'])
          rfm_df.head()
```

	username	Item name	Fulfilled at	Recency	Frequency	Monetary
0	018503	Thermal Skinny Outdoor Trousers Black - 26L / ...	2020-04-07	246	1	79.0
1	0902179	MAX Stretch Skinny Outdoor Trousers Navy - 28R...	2020-08-25	106	1	66.0
2	09616007keg	MAX Stretch Skinny Outdoor Trousers Black - 28L...	2020-04-27	226	1	66.0
3	09sharonlee	GRANITE Mid-weight Outdoor Trousers Grey - 32L...	2020-01-26	318	1	70.0
4	0megannmitche0	MAX Stretch Skinny Outdoor Trousers Black - 28L...	2020-04-24	229	1	66.0

```
In [492]: df1[df1['username']=="0902179"]

Out[492]:
```

	Name	username	Financial Status	Paid at	Fulfillment Status	Fulfilled at	Accepts Marketing	Currency	Subtotal	Shipping	...	Tax 5 Name	Tax 5 Value	Phone	Recei Num
49530	#AC48360	0902179	paid	2020-08-25	fulfilled	2020-08-25 15:12:03+01:00	no	GBP	66.0	3.5	...	NaN	NaN	NaN	NaN

1 rows x 78 columns

```
In [493]: (now - dt.date(2020,9,26)).days == 74

Out[493]: True
```

```
In [494]: quantiles = rfm_df.quantile(q=[0.25,0.5,0.75])
          quantiles
```

	Recency	Frequency	Monetary
0.25	136.25	1.0	50.0
0.50	202.00	1.0	66.0
0.75	300.00	1.0	72.0

```
In [495]: # Arguments (x = value, p = recency, monetary_value, frequency, d = quartiles dict)
          def RFMScore(x,p,d):
              if x <= d[p][0.25]:
                  return 4
              elif x <= d[p][0.50]:
                  return 3
              elif x <= d[p][0.75]:
                  return 2
              else:
                  return 1

          # Arguments (x = value, p = recency, monetary_value, frequency, k = quartiles dict)
          def FMScore(x,p,d):
              if x <= d[p][0.25]:
                  return 4
              elif x <= d[p][0.50]:
                  return 3
              elif x <= d[p][0.75]:
                  return 2
              else:
                  return 1
```

```
In [496]: rfm_segmentation = rfm_df
          rfm_segmentation['R_Quartile'] = rfm_segmentation['Recency'].apply(RFScore, args=('Recency',quantiles,))
          rfm_segmentation['F_Quartile'] = rfm_segmentation['Frequency'].apply(FMScore, args=('Frequency',quantiles,))
          rfm_segmentation['M_Quartile'] = rfm_segmentation['Monetary'].apply(FMScore, args=('Monetary',quantiles,))

          rfm_segmentation.head()
```

	username	Item name	Fulfilled at	Recency	Frequency	Monetary	R_Quartile	F_Quartile	M_Quartile
0	018503	Thermal Skinny Outdoor Trousers Black - 26L / ...	2020-04-07	246	1	79.0	2	1	4
1	0902179	MAX Stretch Skinny Outdoor Trousers Navy - 28R...	2020-08-25	106	1	66.0	4	1	2
2	09616007keg	MAX Stretch Skinny Outdoor Trousers Black - 28L...	2020-04-27	226	1	66.0	2	1	2
3	09sharonlee	GRANITE Mid-weight Outdoor Trousers Grey - 32L...	2020-01-26	318	1	70.0	1	1	3
4	0megannmitche0	MAX Stretch Skinny Outdoor Trousers Black - 28L...	2020-04-24	229	1	66.0	2	1	2

## for all R,F,M scores 1 is highest/best score and 4 is the lowest

```
In [498]: rfm_segmentation['RFMScore'] = rfm_segmentation.R_Quartile.map(etr) \
          + rfm_segmentation.F_Quartile.map(etr) \
          + rfm_segmentation.M_Quartile.map(etr)
```

	username	Item name	Fulfilled at	Recency	Frequency	Monetary	R_Quartile	F_Quartile	M_Quartile	RFMScore
0	018503	Thermal Skinny Outdoor Trousers Black - 26L / ...	2020-04-07	246	1	79.0	2	1	4	214
1	0902179	MAX Stretch Skinny Outdoor Trousers Navy - 28R...	2020-08-25	106	1	66.0	4	1	2	412
2	09616007keg	MAX Stretch Skinny Outdoor Trousers Black - 28L...	2020-04-27	226	1	66.0	2	1	2	212
3	09sharonlee	GRANITE Mid-weight Outdoor Trousers Grey - 32L...	2020-01-26	318	1	70.0	1	1	3	113
4	0megannmitche0	MAX Stretch Skinny Outdoor Trousers Black - 28L...	2020-04-24	229	1	66.0	2	1	2	212

```
In [499]: print("Best Customers: ",len(rfm_segmentation[rfm_segmentation['RFMScore']==4]))
          print("Loyal Customers: ",len(rfm_segmentation[rfm_segmentation['F_Quartile']==4]))
          print("Big Spenders: ",len(rfm_segmentation[rfm_segmentation['M_Quartile']==4]))
          print("Almost Lost: ",len(rfm_segmentation[rfm_segmentation['RFMScore']==244]))
          print("Lost Customers: ",len(rfm_segmentation[rfm_segmentation['RFMScore']==144]))
          print("Lost Cheap Customers: ",len(rfm_segmentation[rfm_segmentation['RFMScore']==111]))
```

```
Best Customers: 24
Loyal Customers: 86
Big Spenders: 3077
Almost Lost: 15
Lost Customers: 14
Lost Cheap Customers: 960
```

## Exporting the segmentation files as csv

```
In [508]: rfm_segmentation.to_csv('segmentation.csv')
```

```
In [ ]:
```