UK Online Store Retail Transactions

Dataset Variable Information:

1. 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.
2. StockCode: Product (item) code. Nominal, a 5-digit integral number uniquely assigned to each distinct product.
3. Description: Product (item) name. Nominal.
4. Quantity: The quantities of each product (item) per transaction. Numeric.
5. InvoiceDate: Invoice Date and time. Numeric, the day and time when each transaction was generated.
6. UnitPrice: Unit price. Numeric, Product price per unit in sterling.
7. CustomerID: Customer number. Nominal, a 5-digit integral number uniquely assigned to each customer.
8. Country: Country name. Nominal, the name of the country where each customer resides.

Establishing Python Library Packages

Show code

Dataset Overview

Show code

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850.0	United Kingdom
1	536365	71053	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850.0	United Kingdom

Dataset Summary Overview

Show code

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 541909 entries, 0 to 541908
Data columns (total 8 columns):
 #   Column       Non-Null Count   Dtype
---  ------       --------------   -----
 0   InvoiceNo    541909 non-null  object
 1   StockCode    541909 non-null  object
 2   Description  540455 non-null  object
 3   Quantity     541909 non-null  int64
 4   InvoiceDate  541909 non-null  datetime64[ns]
 5   UnitPrice    541909 non-null  float64
 6   CustomerID   406829 non-null  float64
 7   Country      541909 non-null  object
dtypes: datetime64[ns](1), float64(2), int64(1), object(4)
memory usage: 33.1+ MB
```

// Observations

Show code

Observation

- Dataset has 8 columns

- Max Row numbers: 541,909
- "Description" and "CustomerID" have lesser row count; possibly null values
- Description = 540,455 total rows
- CustomerID = 406,829 total rows
- "CustomerID" datatype is float64; convert into str object

.

Show code

CLEANING | Null Values

Count nulls

Show code

```
InvoiceNo           0
StockCode           0
Description      1454
Quantity            0
InvoiceDate         0
UnitPrice           0
CustomerID     135080
Country             0
dtype: int64
```

.describe(): 'Description' overview

Show code

	InvoiceNo	StockCode	Description	Country
count	541909	541909	540455	541909
unique	25900	4070	4223	38
top	573585	85123A	WHITE HANGING HEART T-LIGHT HOLDER	United Kingdom
freq	1114	2313	2369	495478

// Observations

Show code

Observation

where using 'StockCode' as identifier:

- 4070 unique rows on 'StockCode'
- 1454 null values on 'StockCode'
- 'StockCode' = 541,909 total row
- 'Description' = 540,455 total rows
- // most likely, 1,454 'StockCode' rows have no corresponding 'Description', (541,909 - 540,455)

.

Show code

[Description] Nulls

Show code

~~ Investigate: 'Description' Null Values

Show code

// Objective: generate a dataframe with 3 columns:

1. 'StockCode' = lists out unique rows
2. 'Count' = shows the number of occurrences of each unique 'StockCode'
3. 'Description' = provides the corresponding description for each 'StockCode'

// Method: create specific dataframes then concatenate ON unique 'StockCode'

.

Create Dataframe: unique 'StockCode' & corresponding counts

Show code

Create Dataframe: excluding 'Description' nulls on 'raw'

Show code

```
<class 'pandas.core.frame.DataFrame'>
Index: 540455 entries, 0 to 541908
Data columns (total 8 columns):
 #   Column       Non-Null Count   Dtype
---  ------       --------------   -----
 0   InvoiceNo    540455 non-null  object
 1   StockCode    540455 non-null  object
 2   Description  540455 non-null  object
 3   Quantity     540455 non-null  int64
 4   InvoiceDate  540455 non-null  datetime64[ns]
 5   UnitPrice    540455 non-null  float64
 6   CustomerID   406829 non-null  float64
 7   Country      540455 non-null  object
dtypes: datetime64[ns](1), float64(2), int64(1), object(4)
memory usage: 37.1+ MB
```

// Observations

Show code

Observation

- 540,455 max rows as per excluding NAs (original 541,909 max rows)
- 'StockCode' = 540,455 total rows (previously 541,909)
- 1,454 rows are 'Description' nulls as per calculation and section: counting nulls

Create Dataframe: unique 'StockCode' & corresponding 'Description'

Show code

```
StockCode
10002     INFLATABLE POLITICAL GLOBE
10080        GROOVY CACTUS INFLATABLE
10120                     DOGGY RUBBER
10123C          HEARTS WRAPPING TAPE
10124A    SPOTS ON RED BOOKCOVER TAPE
Name: Description, dtype: object
```

Concatenate Dataframe: unique StockCode + Counts + 'Description'

Show code

	StockCode	Count	Description
3536	85123A	2313	WHITE HANGING HEART T-LIGHT HOLDER
1348	22423	2203	REGENCY CAKESTAND 3 TIER
3515	85099B	2159	JUMBO BAG RED RETROSPOT
2733	47566	1727	PARTY BUNTING
180	20725	1639	LUNCH BAG RED RETROSPOT
...	...	...	...
885	21854	0	NaN
886	21858	0	NaN
2786	62095B	0	NaN
937	21923	0	NaN
2593	35951	0	NaN

4070 rows × 3 columns

Next steps: Generate code with `unique_stocks` | View recommended plots | New interactive sheet

// Observations

Show code

Observation

- 4070 unique 'StockCode' values (consistent with section:.describe(): 'Description' overview)
- highest count at 2,313 = 'StockCode' 85123A, WHITE HANGING HEART T-LIGHT HOLDER

Merge DataFrames: 'unique_stocks' and 'raw'

Show code

```
<class 'pandas.core.frame.DataFrame'>
Index: 541909 entries, 160128 to 40383
Data columns (total 10 columns):
 #   Column         Non-Null Count   Dtype
---  ------         --------------   -----
 0   InvoiceNo      541909 non-null  object
 1   StockCode      541909 non-null  object
 2   Description_x  540455 non-null  object
 3   Quantity       541909 non-null  int64
 4   InvoiceDate    541909 non-null  datetime64[ns]
 5   UnitPrice      541909 non-null  float64
 6   CustomerID     406829 non-null  float64
 7   Country        541909 non-null  object
 8   Count          541909 non-null  int64
 9   Description_y  541797 non-null  object
dtypes: datetime64[ns](1), float64(2), int64(2), object(5)
memory usage: 45.5+ MB
```

// Observations

Show code

Observation

- Description_x = 540,455 total rows (from 'raw)
- Description_y = 541, 797 total rows (from 'unique_stocks')
- CustomerID = 406, 829 total rows

- CustomerID datatype = float64 (must be converted into 'object')
- 541, 909 max total rows

Refine generated DataFrame

Show code

	InvoiceNo	StockCode	Quantity	InvoiceDate	UnitPrice	CustomerID	Country	Description
0	536365	85123A	6	2010-12-01 08:26:00	2.55	17850.0	United Kingdom	WHITE HANGING HEART T-LIGHT HOLDER
1	536365	71053	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom	WHITE METAL LANTERN
2	536365	84406B	8	2010-12-01 08:26:00	2.75	17850.0	United Kingdom	CREAM CUPID HEARTS COAT HANGER

Count remaining nulls

Show code

```
InvoiceNo           0
StockCode           0
Quantity            0
InvoiceDate         0
UnitPrice           0
CustomerID     135080
Country             0
Description       112
dtype: int64
```

// Observations

Show code

Observation

- There are still 112 nulls on 'Description'
- 135,080 nulls on 'CustomerID'

~~Investigate: 'Description' Remaining Null Values

Show code

// Objective: examine nature of nulls on [Description]; specifically, those that could pose as irrelevant rows for the sales transaction analysis

.

// Method: identify nature of 'Description' nulls accounting corresponding values on the following: (1) 'UnitPrice' (2) 'Quantity' (3) 'CustomerID'

Create DataFrame: examine nulls

Show code

```
<class 'pandas.core.frame.DataFrame'>
Index: 0 entries
Data columns (total 5 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   index        0 non-null      int64
 1   UnitPrice    0 non-null      float64
 2   Description  0 non-null      object
 3   Quantity     0 non-null      int64
 4   CustomerID   0 non-null      object
dtypes: float64(1), int64(2), object(2)
memory usage: 0.0+ bytes
```

```
null_zero_unitprice.describe()
```

	index	UnitPrice	Quantity	CustomerID
count	112.000000	112.0	112.000000	0.0
mean	129823.839286	0.0	-8.196429	NaN
std	83493.428445	0.0	16.003288	NaN
min	1970.000000	0.0	-102.000000	NaN
25%	75228.500000	0.0	-11.000000	NaN
50%	143303.500000	0.0	-4.000000	NaN
75%	171576.500000	0.0	-1.000000	NaN
max	497301.000000	0.0	57.000000	NaN

```
raw[raw['Description'].isna()].describe()
```

	Quantity	InvoiceDate	UnitPrice	CustomerID
count	112.000000	112	112.0	0.0
mean	-8.196429	2011-03-19 12:59:55.178571520	0.0	NaN
min	-102.000000	2010-12-01 14:32:00	0.0	NaN
25%	-11.000000	2011-01-28 14:48:15	0.0	NaN
50%	-4.000000	2011-04-01 16:40:30	0.0	NaN
75%	-1.000000	2011-04-28 15:06:15	0.0	NaN
max	57.000000	2011-11-24 10:36:00	0.0	NaN
std	16.003288	NaN	0.0	NaN

// Observations

Show code

Observation

- 112 rows are zero in 'UnitPrice' and null in both 'Description' and 'CustomerID'
- Considering these 112 have insufficient information, they will be considered as irrelevent rows hence be removed

Remove: null rows

Show code

```
Reviewing null count on working dataset
InvoiceNo           0
StockCode           0
Quantity            0
InvoiceDate         0
UnitPrice           0
CustomerID     135080
Country             0
Description       112
dtype: int64
```

Updated DataFrame

Show code

```
Updated dataset: reviewing null count
InvoiceNo           0
StockCode           0
Quantity            0
InvoiceDate         0
UnitPrice           0
CustomerID     134968
Country             0
```

```
Description         0
dtype: int64
```

// Observations

Show code

Observations

- 'Description' has now zero nulls
- 'CustomerID' has 134,968 nulls

.

Show code

[CustomerID] Nulls

Show code

// Objective: remaining nulls on CustomerID are consired relevant rows hence be kept

// Method:

1. rename those nulls with 'NA'
2. convert datatype float64 into int64 (to remove decimals), then str 'object'

Replace: null values with 'NA'

Show code

```
InvoiceNo      0
StockCode      0
Quantity       0
InvoiceDate    0
UnitPrice      0
CustomerID     0
Country        0
Description    0
dtype: int64
```

```
raw.describe(include='object')
```

	InvoiceNo	StockCode	CustomerID	Country	Description
count	541797	541797	541797	541797	541797
unique	25788	3958	4373	38	3823
top	573585	85123A	NA	United Kingdom	WHITE HANGING HEART T-LIGHT HOLDER
freq	1114	2313	134968	495366	2380

NULL-CLEAN Working Dataset

Show code

```
<class 'pandas.core.frame.DataFrame'>
Index: 541797 entries, 0 to 541908
Data columns (total 8 columns):
 #   Column       Non-Null Count   Dtype
---  ------       --------------   -----
 0   InvoiceNo    541797 non-null  object
 1   StockCode    541797 non-null  object
 2   Quantity     541797 non-null  int64
```

```
 3   InvoiceDate  541797 non-null  datetime64[ns]
 4   UnitPrice    541797 non-null  float64
 5   CustomerID   541797 non-null  object
 6   Country      541797 non-null  object
 7   Description  541797 non-null  object
dtypes: datetime64[ns](1), float64(1), int64(1), object(5)
memory usage: 37.2+ MB
```

.

Show code

CLEANING | Duplicate Rows

Count Duplicate Rows

Show code

```
5270
```

Remove: duplicate rows

Show code

.shape: updated dataset

Show code

```
(536527, 8)
```

// Observations

Show code

Observation

- Updated dataset = 536,527 max total rows (previously 541,797)
- Removed 5,270 duplicate rows

.

Show code

COLUMNS | Examine Nature of numeric values

.describe() numberic values

Show code

	Quantity	InvoiceDate	UnitPrice
count	536527.000000	536527	536527.000000
mean	9.623219	2011-07-04 09:28:59.156911360	4.633627
min	-80995.000000	2010-12-01 08:26:00	-11062.060000
25%	1.000000	2011-03-28 11:34:00	1.250000
50%	3.000000	2011-07-19 14:29:00	2.080000
75%	10.000000	2011-10-18 17:05:00	4.130000
max	80995.000000	2011-12-09 12:50:00	38970.000000
std	219.152804	NaN	97.243424

// Observation

Show code

Observation

- 'Quantity' = -80,995.00 extreme min value
- 'UnitPrice' = -11062.06 extreme min value
- 'InvoiceDate' = December 2010 to 2011 transaction range of dataset

.

Show code

[UnitPrice] Extreme Values

Show code

~~Investigate: 'UnitPrice' extreme values

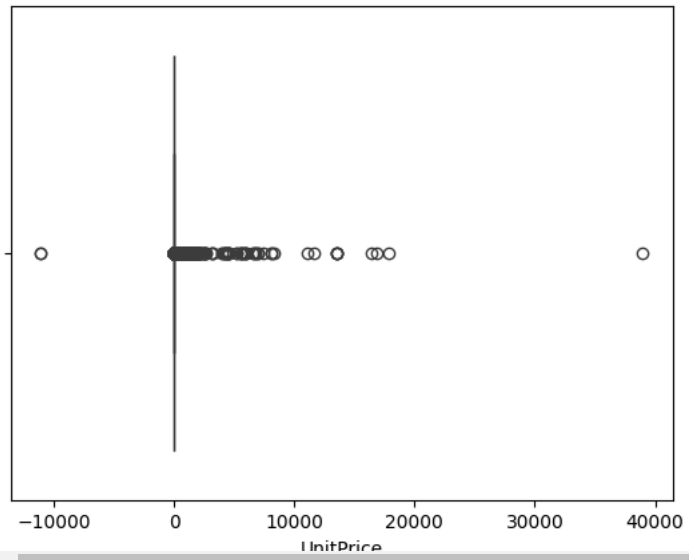
Show code

```
<class 'pandas.core.frame.DataFrame'>
Index: 536527 entries, 0 to 541908
Data columns (total 8 columns):
 #   Column       Non-Null Count   Dtype
---  ------       --------------   -----
 0   InvoiceNo    536527 non-null  object
 1   StockCode    536527 non-null  object
 2   Quantity     536527 non-null  int64
 3   InvoiceDate  536527 non-null  datetime64[ns]
 4   UnitPrice    536527 non-null  float64
 5   CustomerID   536527 non-null  object
 6   Country      536527 non-null  object
 7   Description  536527 non-null  object
dtypes: datetime64[ns](1), float64(1), int64(1), object(5)
memory usage: 36.8+ MB
```

Check Outlier: boxplot 'UnitPrice'

Show code

```
<Axes: xlabel='UnitPrice'>
```



Check Outlier: isolate values

Show code

	InvoiceNo	StockCode	Quantity	InvoiceDate	UnitPrice	CustomerID	Country	Description
15016	C537630	AMAZONFEE	-1	2010-12-07 15:04:00	13541.33	NA	United Kingdom	AMAZON FEE
15017	537632	AMAZONFEE	1	2010-12-07 15:08:00	13541.33	NA	United Kingdom	AMAZON FEE
16232	C537644	AMAZONFEE	-1	2010-12-07 15:34:00	13474.79	NA	United Kingdom	AMAZON FEE
16356	C537651	AMAZONFEE	-1	2010-12-07 15:49:00	13541.33	NA	United Kingdom	AMAZON FEE
43702	C540117	AMAZONFEE	-1	2011-01-05 09:55:00	16888.02	NA	United Kingdom	AMAZON FEE
43703	C540118	AMAZONFEE	-1	2011-01-05 09:57:00	16453.71	NA	United Kingdom	AMAZON FEE
222681	C556445	M	-1	2011-06-10 15:31:00	38970.00	15098	United Kingdom	Manual
299982	A563185	B	1	2011-08-12 14:50:00	11062.06	NA	United Kingdom	Adjust bad debt
524601	C580604	AMAZONFEE	-1	2011-12-05 11:35:00	11586.50	NA	United Kingdom	AMAZON FEE
524602	C580605	AMAZONFEE	1	2011-12-05 11:36:00	17836.46	NA	United Kingdom	AMAZON FEE

// Observation

Show code

Observation

- 'UnitPrice' values > 10000 have alphameric 'StockCode' values {instead of alphanumeric}
- Hence, investigate nature of extreme values accounting columns (1) UnitPrice, (2) StockCode, (3) Quantity, (4) Description

---

Examine 'StockCode' Alphamerics

Show code

// Objective: find patterns on 'StockCode' related to the extreme values found on 'UnitPrice'

// Method:

1. create a dataframe isolating only alphameric values on 'StockCode'
2. create a dataframe of unique alphameric 'StockCode'& corresponding counts

3. create dataframe printing out the following:

- (1) unique alphameric 'StockCode'
- (2) each corresponding 'Description'
- (3) each corresponding count of 'StockCode' occurences
- (4) each corresponding most reoccurring value on 'Quantity' and 'UnitPrice'

Create Dataframe: .info() alphameric 'StockCode'

Show code

```
<class 'pandas.core.frame.DataFrame'>
Index: 2790 entries, 45 to 541768
Data columns (total 4 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   StockCode    2790 non-null   object
 1   Description  2790 non-null   object
 2   Quantity     2790 non-null   int64
 3   UnitPrice    2790 non-null   float64
dtypes: float64(1), int64(1), object(2)
memory usage: 109.0+ KB
```

Create Dataframe: unique alphameric 'StockCode' & corresponding counts

Show code

Create Dataframe: unique alphameric 'StockCode' + Description + Count + Quantity + UnitPrice

Show code

```
Total Count Rows containing Alphameric StockCode Values = 2790
```

	StockCode	Description	Count	Max_Quantity	Max_UnitPrice
0	AMAZONFEE	AMAZON FEE	34	-1	13541.330
1	B	Adjust bad debt	3	1	-11062.060
2	BANK CHARGES	Bank Charges	37	-1	15.000
3	CRUK	CRUK Commission	16	-1	1.600
4	D	Discount	77	-1	11.840
5	DCGSSBOY	BOYS PARTY BAG	11	1	3.290
6	DCGSSGIRL	GIRLS PARTY BAG	13	2	3.290
7	DOT	DOTCOM POSTAGE	710	1	3.290
8	M	Manual	566	-1	1.250
9	PADS	PADS TO MATCH ALL CUSHIONS	4	1	0.001
10	POST	POSTAGE	1256	1	18.000
11	S	SAMPLES	62	-1	33.050
12	m	Manual	1	1	2.550

Next steps: Generate code with combined_df | View recommended plots | New interactive sheet

// Observations

Show code

Observation:

- 2,790 alphameric 'StockCode' rows
- 13 unique alphameric 'StockCode' values
- Most identified alphameric 'StockCode' are not relevant to the sales transaction analysis; all shall be removed except:

1. DCGSSBOY
2. DCGSSGIRL

Remove: certain alphameric values on 'StockCode'

Show code

Remove: alphameric StockCode on Working Dataframe

Show code

```
<class 'pandas.core.frame.DataFrame'>
Index: 533761 entries, 0 to 541908
Data columns (total 8 columns):
 #   Column       Non-Null Count   Dtype
---  ------       --------------   -----
 0   InvoiceNo    533761 non-null  object
 1   StockCode    533761 non-null  object
 2   Quantity     533761 non-null  int64
 3   InvoiceDate  533761 non-null  datetime64[ns]
 4   UnitPrice    533761 non-null  float64
 5   CustomerID   533761 non-null  object
 6   Country      533761 non-null  object
 7   Description  533761 non-null  object
dtypes: datetime64[ns](1), float64(1), int64(1), object(5)
memory usage: 36.7+ MB
```

Check Further 'StockCode' Alphanumeric

Show code

```
StockCode
2    433968
8     62423
4     11368
1      7574
7      7142
3      5691
9      4633
5       633
C       144
6       112
D        39
g        34
Name: count, dtype: int64
```

```
raw[raw['StockCode'].str[0] == 'C']
```

	InvoiceNo	StockCode	Quantity	InvoiceDate	UnitPrice	CustomerID	Country	Description
1423	536540	C2	1	2010-12-01 14:05:00	50.0	14911	EIRE	CARRIAGE
12119	537368	C2	1	2010-12-06 12:40:00	50.0	14911	EIRE	CARRIAGE
12452	537378	C2	1	2010-12-06 13:06:00	50.0	14911	EIRE	CARRIAGE
19975	537963	C2	1	2010-12-09 11:30:00	50.0	13369	United Kingdom	CARRIAGE
20016	538002	C2	1	2010-12-09 11:48:00	50.0	14932	Channel Islands	CARRIAGE
...	...	...	...	...	...	...	...	...
515000	579768	C2	1	2011-11-30 15:08:00	50.0	14911	EIRE	CARRIAGE
516484	579910	C2	1	2011-12-01 08:52:00	50.0	14911	EIRE	CARRIAGE
518854	580124	C2	1	2011-12-01 17:12:00	50.0	NA	EIRE	CARRIAGE
518905	580127	C2	1	2011-12-01 17:51:00	50.0	14911	EIRE	CARRIAGE
524450	580555	C2	1	2011-12-05 10:18:00	50.0	14911	EIRE	CARRIAGE

144 rows × 8 columns

```
raw[raw['StockCode'].str[0] == 'C'].describe(include = 'object')
```

	InvoiceNo	StockCode	CustomerID	Country	Description
count	144	144	144	144	144
unique	144	1	30	4	1
top	536540	C2	14911	EIRE	CARRIAGE
freq	1	144	85	108	144

```
raw[raw['StockCode'].str[0] == 'D']['Description']
```

```
21326     SUNJAR LED NIGHT NIGHT LIGHT
24906               BOXED GLASS ASHTRAY
36460               BOXED GLASS ASHTRAY
39313     SUNJAR LED NIGHT NIGHT LIGHT
40052            CAMOUFLAGE DOG COLLAR
75053            OOH LA LA DOGS COLLAR
76251               BOXED GLASS ASHTRAY
84016                    BOYS PARTY BAG
84017                   GIRLS PARTY BAG
97246                    BOYS PARTY BAG
112723                   BOYS PARTY BAG
112724                  GIRLS PARTY BAG
116891                   BOYS PARTY BAG
116892                  GIRLS PARTY BAG
128107                   BOYS PARTY BAG
128108                  GIRLS PARTY BAG
128269                  GIRLS PARTY BAG
150864                  GIRLS PARTY BAG
160487                   BOYS PARTY BAG
170783      HAYNES CAMPER SHOULDER BAG
176006              BOXED GLASS ASHTRAY
176169                  GIRLS PARTY BAG
178669                   BOYS PARTY BAG
178670                  GIRLS PARTY BAG
262771                   BOYS PARTY BAG
278378                   BOYS PARTY BAG
278379                  GIRLS PARTY BAG
279251                             ebay
279253           CAMOUFLAGE DOG COLLAR
279254           OOH LA LA DOGS COLLAR
279255                             ebay
279256                             ebay
279258              BOXED GLASS ASHTRAY
297098                  GIRLS PARTY BAG
318312                  GIRLS PARTY BAG
365966                   BOYS PARTY BAG
408203                  GIRLS PARTY BAG
474602                  GIRLS PARTY BAG
518711                   BOYS PARTY BAG
Name: Description, dtype: object
```

```
raw[raw['StockCode'].str[0] == 'g']
```

	InvoiceNo	StockCode	Quantity	InvoiceDate	UnitPrice	CustomerID	Country	Description
38248	539492	gift_0001_40	1	2010-12-20 10:14:00	34.04	NA	United Kingdom	Dotcomgiftshop Gift Voucher £40.00
42057	539958	gift_0001_50	1	2010-12-23 13:26:00	42.55	NA	United Kingdom	Dotcomgiftshop Gift Voucher £50.00
44725	540238	gift_0001_30	1	2011-01-05 14:44:00	25.53	NA	United Kingdom	Dotcomgiftshop Gift Voucher £30.00
44794	540238	gift_0001_20	1	2011-01-05 14:44:00	17.02	NA	United Kingdom	Dotcomgiftshop Gift Voucher £20.00
55589	540995	gift_0001_20	1	2011-01-13 09:30:00	16.67	NA	United Kingdom	Dotcomgiftshop Gift Voucher £20.00
91278	544089	gift_0001_20	1	2011-02-15 17:51:00	16.67	NA	United Kingdom	Dotcomgiftshop Gift Voucher £20.00
94054	544323	gift_0001_30	1	2011-02-17 15:51:00	25.00	NA	United Kingdom	Dotcomgiftshop Gift Voucher £30.00
95034	544434	gift_0001_30	1	2011-02-18 16:12:00	25.00	NA	United Kingdom	Dotcomgiftshop Gift Voucher £30.00
112442	545895	gift_0001_10	1	2011-03-07 17:14:00	8.33	NA	United Kingdom	Dotcomgiftshop Gift Voucher £10.00
145463	548893	gift_0001_40	1	2011-04-04 15:54:00	33.33	NA	United Kingdom	Dotcomgiftshop Gift Voucher £40.00
161388	550474	gift_0001_20	2	2011-04-18 13:58:00	16.67	NA	United Kingdom	Dotcomgiftshop Gift Voucher £20.00
163439	550542	gift_0001_10	1	2011-04-19 11:37:00	8.33	NA	United Kingdom	Dotcomgiftshop Gift Voucher £10.00
163440	550542	gift_0001_20	1	2011-04-19 11:37:00	16.67	NA	United Kingdom	Dotcomgiftshop Gift Voucher £20.00
178556	552232	gift_0001_50	1	2011-05-06 15:54:00	41.67	NA	United Kingdom	Dotcomgiftshop Gift Voucher £50.00
191910	553387	gift_0001_10	1	2011-05-16 16:33:00	8.33	NA	United Kingdom	Dotcomgiftshop Gift Voucher £10.00
192071	553389	gift_0001_10	1	2011-05-16 16:37:00	8.33	NA	United Kingdom	Dotcomgiftshop Gift Voucher £10.00
208808	555149	gift_0001_30	1	2011-05-31 15:49:00	25.00	NA	United Kingdom	Dotcomgiftshop Gift Voucher £30.00
228807	556955	gift_0001_20	10	2011-06-16 09:04:00	0.00	NA	United Kingdom	Dotcomgiftshop Gift Voucher £20.00
234137	557500	gift_0001_20	1	2011-06-20 15:27:00	16.67	NA	United Kingdom	Dotcomgiftshop Gift Voucher £20.00
239744	558066	gift_0001_50	1	2011-06-24 15:45:00	41.67	NA	United Kingdom	Dotcomgiftshop Gift Voucher £50.00
239899	558068	gift_0001_20	1	2011-06-24 15:51:00	16.67	NA	United Kingdom	Dotcomgiftshop Gift Voucher £20.00
245515	558614	gift_0001_10	1	2011-06-30	8.33	NA	United	Dotcomgiftshop Gift Voucher

// Observations

Show code

Observations

- 144 rows starting with 'C' = CARRIAGE; remove since these are not sales transactions
- 39 rows starting with 'D' = has several descriptions; but remove 'ebay' records since these are not sales transactions
- 34 rows starting with 'g' = gift vouchers; since no furthere details were found, these will be assumed as purchased vouchers since the values on 'Quantity' are non-negaitives

Remove: alphanumeric 'StockCode' on Working DataFrame

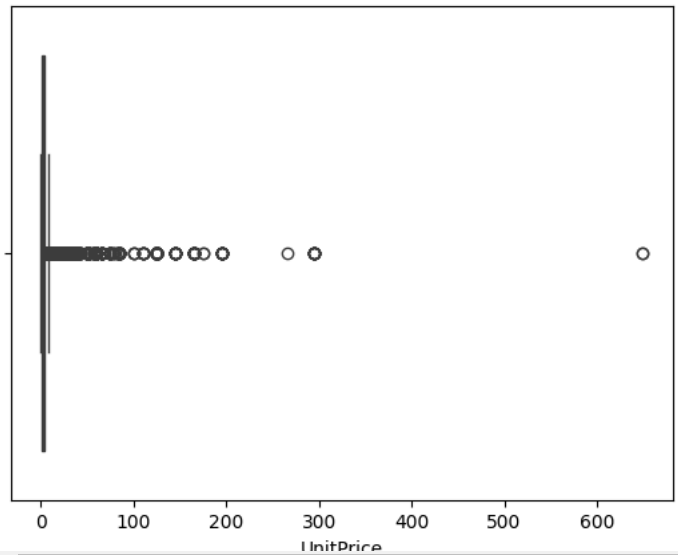
Show code

```
<class 'pandas.core.frame.DataFrame'>
Index: 533614 entries, 0 to 541908
Data columns (total 8 columns):
 #   Column       Non-Null Count   Dtype
---  ------       --------------   -----
 0   InvoiceNo    533614 non-null  object
 1   StockCode    533614 non-null  object
 2   Quantity     533614 non-null  int64
 3   InvoiceDate  533614 non-null  datetime64[ns]
 4   UnitPrice    533614 non-null  float64
 5   CustomerID   533614 non-null  object
 6   Country      533614 non-null  object
 7   Description  533614 non-null  object
dtypes: datetime64[ns](1), float64(1), int64(1), object(5)
memory usage: 36.6+ MB
```

Check Outlier: boxplot 'Unitprice'

Show code

```
<Axes: xlabel='UnitPrice'>
```



// Update: Working DataFrame

Show code

Observation

- removal of particular rows improved the distribution on 'UnitPrice'
- 2, 913 rows were removed
- 533, 614 rows on updated working dataset

[Quantity] Extreme Values

Show code

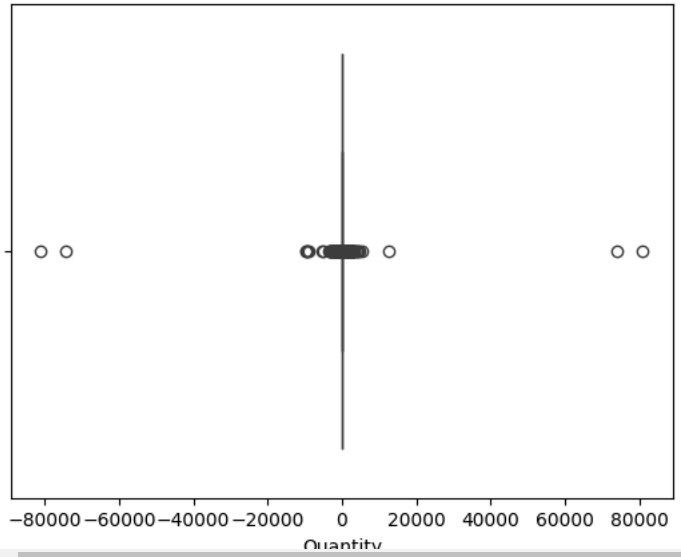
~~Investigate: 'Quantity' extreme values

Show code

Check Outlier: boxplot 'Quantity'

Show code

```
<Axes: xlabel='Quantity'>
```



Check Outlier: isolate values

Show code

```
Rows having extreme values (>= 15000 and <= -15000) on 'Quantity'
```

	InvoiceNo	StockCode	Quantity	InvoiceDate	UnitPrice	CustomerID	Country	Description
61619	541431	23166	74215	2011-01-18 10:01:00	1.04	12346	United Kingdom	MEDIUM CERAMIC TOP STORAGE JAR
61624	C541433	23166	-74215	2011-01-18 10:17:00	1.04	12346	United Kingdom	MEDIUM CERAMIC TOP STORAGE JAR

Next steps: Generate code with `exclude_quantity` | View recommended plots | New interactive sheet

// Observations

Show code

Observations:

- 'Quantity' Negative values could possibly have a corresponding transaction having a (+) value on 'Quantity'

Remove: extreme values

Show code

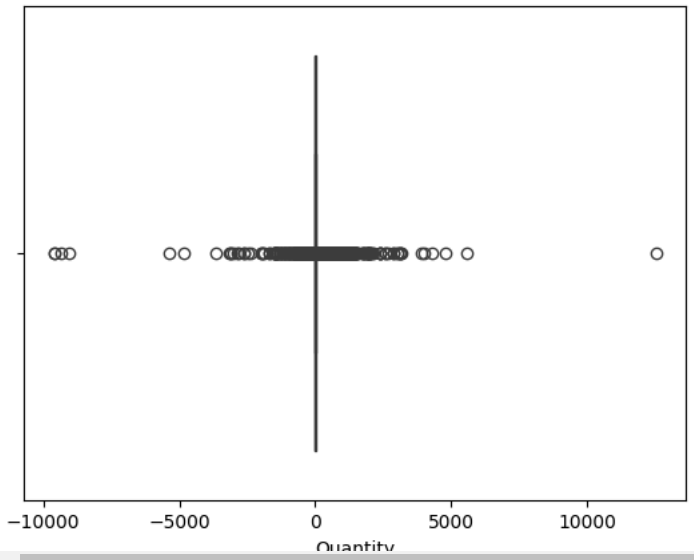
```
<class 'pandas.core.frame.DataFrame'>
Index: 533610 entries, 0 to 541908
Data columns (total 8 columns):
 #   Column       Non-Null Count   Dtype
---  ------       --------------   -----
 0   InvoiceNo    533610 non-null  object
 1   StockCode    533610 non-null  object
 2   Quantity     533610 non-null  int64
 3   InvoiceDate  533610 non-null  datetime64[ns]
```

```
 4   UnitPrice    533610 non-null  float64
 5   CustomerID   533610 non-null  object
 6   Country      533610 non-null  object
 7   Description  533610 non-null  object
dtypes: datetime64[ns](1), float64(1), int64(1), object(5)
memory usage: 36.6+ MB
```

Check Outlier: boxplot 'Quantity'

Show code

```
<Axes: xlabel='Quantity'>
```



// Observations

Show code

Observations:

- 4 rows were considered outliers hence removed
- 533, 610 rows on updated working dataset
- **NOTE:** 'Quantity' Negative values could have a corresponding transaction having a (+) value on 'Quantity; **examine** later in the data mining process
- Several values on 'Quantity' are negative: **examine**

Check negatives on 'Quantity'

Show code

	InvoiceNo	StockCode	Quantity	InvoiceDate	UnitPrice	CustomerID	Country	Description
2406	536589	21777	-10	2010-12-01 16:50:00	0.00	NA	United Kingdom	RECIPE BOX WITH METAL HEART
4347	536764	84952C	-38	2010-12-02 14:42:00	0.00	NA	United Kingdom	MIRROR LOVE BIRD T-LIGHT HOLDER
7188	536996	22712	-20	2010-12-03 15:30:00	0.00	NA	United Kingdom	CARD DOLLY GIRL
7189	536997	22028	-20	2010-12-03 15:30:00	0.00	NA	United Kingdom	PENNY FARTHING BIRTHDAY CARD
7190	536998	85067	-6	2010-12-03 15:30:00	0.00	NA	United Kingdom	CREAM SWEETHEART WALL CABINET
...	...	...	...	...	...	...	...	...
540448	C581490	22178	-12	2011-12-09 09:57:00	1.95	14397	United Kingdom	VICTORIAN GLASS HANGING T-LIGHT
540449	C581490	23144	-11	2011-12-09 09:57:00	0.83	14397	United Kingdom	ZINC T-LIGHT HOLDER STARS SMALL

Next steps: Generate code with `cancelled_transaction_rows` View recommended plots New interactive sheet

// Observations

Show code

Observation

- 9,907 rows have negative (-) values on 'Quantity'
- Some rows begin with 'C' on 'InvoiceNo'; some with '5'

~~Investigate: 'Quantity' negative values

Show code

// Objective: examine nature of negative (-) valued 'Quantity' rows; specifically:

1. those that could have a corresponding transaction having a (+) value on 'Quantity'
2. those that could pose as irrelevant rows for the sales transaction analysis

// Method:

- identify order transactions of those cancelled transactions (having matching details on particular columns while positive (+) on 'Quantity' values) prior the cancellation; accounting the following columns:

-> Exact Values on (1) StockCode (2) Quantity [absolute value] (3) CustomerID (4) Description (5) Country (6) UnitPrice

-> Variation of values on (7) InvoiceNo (8) InvoiceDate

- remove those identified rows that are considered irrelevant cancelled transactions; to then further identify other factors outside cancelled transactions
- identify remaining (-) 'Quantity' transactions; accounting corresponding values on the following: (1) 'InvoiceNo' (2) 'Description' (3) 'Quantity', (4) 'UnitPrice'

Identify Matching Completely Cancelled Transactions

Show code

```
''' where number of exact details placed as orders are equal to the number of exact details of cancelled orders '''
```

	InvoiceNo	StockCode	Quantity	InvoiceDate	UnitPrice	CustomerID	Country	Description
84148	543370	22839	2	2011-02-07 14:51:00	14.95	12359	Cyprus	3 TIER CAKE TIN GREEN AND CREAM
154936	C549955	22839	-2	2011-04-13 13:38:00	14.95	12359	Cyprus	3 TIER CAKE TIN GREEN AND CREAM
423970	573173	22941	2	2011-10-28 10:10:00	8.50	12362	Belgium	CHRISTMAS LIGHTS 10 REINDEER
507365	C579178	22941	-2	2011-11-28 14:55:00	8.50	12362	Belgium	CHRISTMAS LIGHTS 10 REINDEER
423972	573173	22942	2	2011-10-28 10:10:00	8.50	12362	Belgium	CHRISTMAS LIGHTS 10 SANTAS
...	...	...	...	...	...	...	...	...
40658	539739	85126	2	2010-12-21 15:19:00	13.57	NA	United Kingdom	LARGE ROUND CUTGLASS CANDLESTICK
89322	543899	85169C	12	2011-02-14 12:11:00	1.25	NA	EIRE	EAU DE NIL LOVE BIRD CANDLE

Next steps: Generate code with grouped_df | View recommended plots | New interactive sheet

// Observations

Show code

- 4, 202 rows are matching transactions that were once ordered (+ postive in 'Quantity' value) then eventually cancelled (- negative in 'Quantity' value)
- These have to be removed since they didn't not generate a successful sales transaction

Remove: identified matching cancelled transactions

Show code

```
<class 'pandas.core.frame.DataFrame'>
Index: 529555 entries, 0 to 541908
Data columns (total 8 columns):
 #   Column       Non-Null Count   Dtype
---  ------       --------------   -----
 0   InvoiceNo    529555 non-null  object
 1   StockCode    529555 non-null  object
 2   Quantity     529555 non-null  float64
 3   InvoiceDate  529555 non-null  datetime64[ns]
 4   UnitPrice    529555 non-null  float64
 5   CustomerID   529555 non-null  object
 6   Country      529555 non-null  object
 7   Description  529555 non-null  object
dtypes: datetime64[ns](1), float64(2), object(5)
memory usage: 36.4+ MB
```

// Observations

Show code

Observations

- 4, 202 rows were removed
- 529,555 total rows on updated working dataset (previously 533, 757 rows)

identify remaining (-) 'Quantity' transactions; accounting corresponding values on the following: (1) 'InvoiceNo' (2) 'Description' (3) 'Quantity', (4) 'UnitPrice'

Check remaining negatives on 'Quantity'

Show code

	InvoiceNo	StockCode	Quantity	InvoiceDate	UnitPrice	CustomerID	Country	Description
2406	536589	21777	-10.0	2010-12-01 16:50:00	0.00	NA	United Kingdom	RECIPE BOX WITH METAL HEART
4347	536764	84952C	-38.0	2010-12-02 14:42:00	0.00	NA	United Kingdom	MIRROR LOVE BIRD T-LIGHT HOLDER
7188	536996	22712	-20.0	2010-12-03 15:30:00	0.00	NA	United Kingdom	CARD DOLLY GIRL
7189	536997	22028	-20.0	2010-12-03 15:30:00	0.00	NA	United Kingdom	PENNY FARTHING BIRTHDAY CARD
7190	536998	85067	-6.0	2010-12-03 15:30:00	0.00	NA	United Kingdom	CREAM SWEETHEART WALL CABINET
...	...	...	...	...	...	...	...	...
540448	C581490	22178	-12.0	2011-12-09 09:57:00	1.95	14397	United Kingdom	VICTORIAN GLASS HANGING T-LIGHT
540449	C581490	23144	-11.0	2011-12-09 09:57:00	0.83	14397	United Kingdom	ZINC T-LIGHT HOLDER STARS SMALL

Next steps: Generate code with `df_negative_quantity` | View recommended plots | New interactive sheet

// Observations

Show code

Observation:

- 7,806 rows remaining with negative (-) values on 'Quantity'
- Some rows begin with 'C' on 'InvoiceNo'; some with '5'
- Some rows have exact values on 'InvoiceNo', 'CustomerID', 'Country'; **examine**

Count rows: starting with 'C' and '5' on InvoiceNo

Show code

```
InvoiceNo
C    6604
5    1202
Name: count, dtype: int64
```

Check rows: starting with '5' on InvoiceNo

Show code

```
       InvoiceNo StockCode CustomerID         Country             Description
count       1202      1202       1202            1202                    1202
unique      1202       971          1               1                     959
top       536589     21830         NA  United Kingdom  Unsaleable, destroyed.
freq           1         5       1202            1202                       6

          Quantity                    InvoiceDate  UnitPrice
count  1202.000000                           1202     1202.0
mean   -169.785358  2011-06-20 20:33:25.956738560        0.0
min   -9600.000000            2010-12-01 16:50:00        0.0
25%     -95.000000            2011-04-01 11:48:30        0.0
50%     -34.000000            2011-06-10 10:47:00        0.0
75%     -10.000000            2011-10-02 17:53:30        0.0
max      -1.000000            2011-12-08 15:24:00        0.0
std     617.985540                            NaN        0.0
```

	InvoiceNo	StockCode	Quantity	InvoiceDate	UnitPrice	CustomerID	Country	Description
2406	536589	21777	-10.0	2010-12-01 16:50:00	0.0	NA	United Kingdom	RECIPE BOX WITH METAL HEART
4347	536764	84952C	-38.0	2010-12-02 14:42:00	0.0	NA	United Kingdom	MIRROR LOVE BIRD T-LIGHT HOLDER
7188	536996	22712	-20.0	2010-12-03 15:30:00	0.0	NA	United Kingdom	CARD DOLLY GIRL
7189	536997	22028	-20.0	2010-12-03 15:30:00	0.0	NA	United Kingdom	PENNY FARTHING BIRTHDAY CARD
7190	536998	85067	-6.0	2010-12-03 15:30:00	0.0	NA	United Kingdom	CREAM SWEETHEART WALL CABINET
...	...	...	...	...	...	...	...	...
535333	581210	23395	-26.0	2011-12-07 18:36:00	0.0	NA	United Kingdom	BELLE JARDINIERE CUSHION COVER
535335	581212	22578	-1050.0	2011-12-07 18:38:00	0.0	NA	United Kingdom	WOODEN STAR CHRISTMAS SCANDINAVIAN

// Observations

Show code

Observations:

- All rows starting with '5' and with negative values on 'Quantity' have:

1. 'NA' values on 'CustomerID'
2. zero 0 values on 'UnitPrice'

Check rows: starting with 'C' on InvoiceNo

Show code

	InvoiceNo	StockCode	Quantity	InvoiceDate	UnitPrice	CustomerID	Country	Description
154	C536383	35004C	-1.0	2010-12-01 09:49:00	4.65	15311	United Kingdom	SET OF 3 COLOURED FLYING DUCKS
237	C536391	21983	-24.0	2010-12-01 10:24:00	0.29	17548	United Kingdom	PACK OF 12 BLUE PAISLEY TISSUES
236	C536391	21984	-24.0	2010-12-01 10:24:00	0.29	17548	United Kingdom	PACK OF 12 PINK PAISLEY TISSUES
235	C536391	22556	-12.0	2010-12-01 10:24:00	1.65	17548	United Kingdom	PLASTERS IN TIN CIRCUS PARADE
239	C536391	21484	-12.0	2010-12-01 10:24:00	3.45	17548	United Kingdom	CHICK GREY HOT WATER BOTTLE
...	...	...	...	...	...	...	...	...
540448	C581490	22178	-12.0	2011-12-09 09:57:00	1.95	14397	United Kingdom	VICTORIAN GLASS HANGING T-LIGHT
540449	C581490	23144	-11.0	2011-12-09 09:57:00	0.83	14397	United Kingdom	ZINC T-LIGHT HOLDER STARS SMALL

Check Summary: remaining negatives on 'Quantity'

Show code

```
Total Unique 'CustomerID' = 1311
Total Unique 'InvoiceNo' = 4073

Total Unique 'InvoiceNo' starting with '5' = 1202
       // note: all rows starting with '5' and with negative values on 'Quantity' have:
            (1) 'NA' values on 'CustomerID'
            (2) zero 0 values on 'UnitPrice'

Total Unique 'InvoiceNo' starting with 'C' = 2871
```

~~Investigate: starting with 'C' and negative on 'Quantity'

Show code

Questions:

What is the nature of those cancelled transactions with negative (-) values on Quantity?

```
df_negative_quantity[df_negative_quantity['InvoiceNo'].str.startswith('C')].sort_values(by='CustomerID')
```

	InvoiceNo	StockCode	Quantity	InvoiceDate	UnitPrice	CustomerID	Country	Description
129742	C547388	22413	-6.0	2011-03-22 16:07:00	2.95	12352	Norway	METAL SIGN TAKE IT OR LEAVE IT
129739	C547388	22784	-3.0	2011-03-22 16:07:00	4.95	12352	Norway	LANTERN CREAM GAZEBO
129741	C547388	22645	-12.0	2011-03-22 16:07:00	1.45	12352	Norway	CERAMIC HEART FAIRY CAKE MONEY BANK
129740	C547388	22701	-6.0	2011-03-22 16:07:00	2.95	12352	Norway	PINK DOG BOWL
129743	C547388	21914	-12.0	2011-03-22 16:07:00	1.25	12352	Norway	BLUE HARMONICA IN BOX
...	...	...	...	...	...	...	...	...
41620	C539948	21888	-4.0	2010-12-23 11:48:00	3.75	NA	EIRE	BINGO SET
285960	C561966	22371	-1.0	2011-08-01 13:11:00	4.13	NA	United Kingdom	AIRLINE BAG VINTAGE TOKYO 78