```
import pandas as pd
import numpy as ny
from google.colab import files
```

```
uploaded = files.upload()
```

```
Choose Files no files selected Upload widget is only available when the cell has been
executed in the current browser session. Please rerun this cell to enable.
Saving amazon-sales.csv to amazon-sales.csv
```

```
df = pd.read_csv('amazon-sales.csv')
```

```
/tmp/ipython-input-4-139105417.py:1: DtypeWarning: Columns (23) have mixed
  df = pd.read_csv('amazon-sales.csv')
```

```
print(df.shape)
```

```
(128975, 24)
```

```
df.head(-10)
```

	index	Order ID	Date	Status	Fulfilment	Sales Channel	ship-service-level	Sty
0	0	405-8078784-5731545	04-30-22	Cancelled	Merchant	Amazon.in	Standard	SET3
1	1	171-9198151-1101146	04-30-22	Shipped - Delivered to Buyer	Merchant	Amazon.in	Standard	JNE37
2	2	404-0687676-7273146	04-30-22	Shipped	Amazon	Amazon.in	Expedited	JNE33
3	3	403-9615377-8133951	04-30-22	Cancelled	Merchant	Amazon.in	Standard	J03
4	4	407-1069790-7240320	04-30-22	Shipped	Amazon	Amazon.in	Expedited	JNE36
...	...	...	...	...	...	...	...	
128960	128960	402-0468123-8401109	05-31-22	Shipped	Amazon	Amazon.in	Expedited	J03
128961	128961	402-0082204-6323568	05-31-22	Cancelled	Amazon	Amazon.in	Expedited	JNE37
128962	128962	408-9803724-6565965	05-31-22	Cancelled	Amazon	Amazon.in	Expedited	MEN50
128963	128963	404-5963451-7335564	05-31-22	Shipped	Amazon	Amazon.in	Expedited	J03
128964	128964	404-2225394-8024308	05-31-22	Shipped	Amazon	Amazon.in	Expedited	J01

128965 rows × 24 columns

```
df = df.drop(columns=['Unnamed: 22'])
df.head(5)
```

	index	Order ID	Date	Status	Fulfilment	Sales Channel	ship-service-level	Style	
0	0	405-8078784-5731545	04-30-22	Cancelled	Merchant	Amazon.in	Standard	SET389	S KI
1	1	171-9198151-1101146	04-30-22	Shipped - Delivered to Buyer	Merchant	Amazon.in	Standard	JNE3781	JN KF
2	2	404-0687676-7273146	04-30-22	Shipped	Amazon	Amazon.in	Expedited	JNE3371	JN
3	3	403-9615377-8133951	04-30-22	Cancelled	Merchant	Amazon.in	Standard	J0341	
4	4	407-1069790-7240320	04-30-22	Shipped	Amazon	Amazon.in	Expedited	JNE3671	JN TU

5 rows × 23 columns

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

```
index                    0
Order ID                 0
Date                     0
Status                   0
Fulfilment               0
Sales Channel            0
ship-service-level       0
Style                    0
SKU                      0
Category                 0
Size                     0
ASIN                     0
Courier Status        6872
Qty                      0
currency              7795
Amount                7795
ship-city               33
ship-state              33
ship-postal-code        33
ship-country            33
promotion-ids        49153
B2B                      0
fulfilled-by         89698
dtype: int64
```

```
df['fulfilled-by'].value_counts()
```

fulfilled-by	count
Easy Ship	39277

dtype: int64

As here all the shipment is fulfilled by "Easy Ship", there is no harm in assuming that the missing shipment duty must have been assigned to the same company. Whether or not the shipment was delivered, it must have been assigned to "Easy Ship". We will fill those missing values with "Easy Ship"

```
df['fulfilled-by'] = df['fulfilled-by'].fillna('Easy Ship')
df['fulfilled-by'].value_counts()
```

	count
fulfilled-by	
Easy Ship	128975

dtype: int64

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

```
index                    0
Order ID                 0
Date                     0
Status                   0
Fulfilment               0
Sales Channel            0
ship-service-level       0
Style                    0
SKU                      0
Category                 0
Size                     0
ASIN                     0
Courier Status        6872
Qty                      0
currency              7795
Amount                7795
ship-city               33
ship-state              33
ship-postal-code        33
ship-country            33
promotion-ids        49153
B2B                      0
fulfilled-by             0
dtype: int64
```

```
alignment = df['currency'].isnull() == df['Amount'].isnull()
all_aligned = alignment.all()
print(all_aligned)
```

```
True
```

This above code ensures that entries found missing in the "currency" and "Amount" columns match exactly. Means the missing values are matched pair wise.

I suspected that because the number of missing values matched exactly.

Let's check if that's the case with the other four columns where missing values match!

```
new_alignment = (df['ship-city'].isnull() == df['ship-state'].isnull()) & \
                 (df['ship-state'].isnull() == df['ship-postal-code'].isnull())
                 (df['ship-postal-code'].isnull() == df['ship-country'].isnull(
new_all_aligned = new_alignment.all()
print(new_all_aligned)
```

```
True
```

They match!

Below we are dropping Courier Status and Promotion Ids columns as they are not important for our analysis in this project

```
df = df.drop(columns=['Courier Status', 'promotion-ids'])
```

```
mask_col2_missing = df['ship-city'].isnull()

mask_col1_missing = df['currency'].isnull()

all_col2_missing_in_col1 = (mask_col1_missing[mask_col2_missing]).all()
print(all_col2_missing_in_col1)
```

```
False
```

Here we got to know that the missing values form the two groups, where the missing values matched in rows, do not match.

We are going to drop all these messing data rows

We are doing this as it's impact on our analysis would be hardly more than 6%

We can work with that buffer.

```
df.dropna(subset=['currency'], inplace=True)
df.dropna(subset=['ship-city'], inplace=True)
print(df.isnull().sum())
```

```
index                 0
Order ID              0
Date                  0
Status                0
Fulfilment            0
Sales Channel         0
ship-service-level    0
Style                 0
SKU                   0
Category              0
Size                  0
ASIN                  0
Qty                   0
currency              0
Amount                0
ship-city             0
ship-state            0
ship-postal-code      0
ship-country          0
B2B                   0
fulfilled-by          0
dtype: int64
```

Hereby, we have successfully cleaned the data from Amazon Sales with a loss of 6% of data only.

This dataframe is now ready to be utilized for building a report.

```
print(df.shape)
```

```
(121149, 21)
```

```
df.to_csv('cleaned_data.csv', index=False)
files.download('cleaned_data.csv')
```