ASSIGNMENT NO: 6 (Group Activity)

By

Prasad Khandake	**PA-12**
Nikhil Mundey	**PA-23**
Masoom Raza	**PA-17**

Problem Statement

Using visualization(s) to tell a story with data

Theory: Here is one way to start.

Step 1. Pick a domain and data set that you are interested in. https://www.kaggle.com/ or any other repository Choose the one of greatest interest to you. If you would like to explore a different data set, you are free to do so but be aware of how much work might be needed to clean the data and get it into a usable format.

Step 2. Explore the data. Find a story. Ask questions. Start by asking questions. For example: Is there a relationship between melting point and atomic number? Are the brightness and color of stars correlated? Are there different patterns of nucleotides in different regions in human DNA?

Step 3. Assess the fitness of the data for answering your question. Inspect the data -- it is invariably helpful to first look at the raw values. Does the data seem appropriate for answering your question? If not, you may need to start the process over. If so, does the data need to be reformatted or cleaned prior to analysis? Perform any steps necessary to get the data into shape prior to visual analysis.

Step 4. Create the visualization(s) that tell a story about the data. You will likely need to create several and see what works best. Keep a record of things you tried and discarded.

Input: Link of dataset which you have used

https://www.kaggle.com/hendraherviawan/customer-purchasing-patterns/data?select=data.csv

Create a Doc/ppt that contains the following information:

1. Describe the data set you chose and why?

E-Commerce Data

Actual transactions from UK retailer

Now a days there are many purchase are happing in retail sector so we though why not to understand customers behaviour towards buying products and their spending pattern.

Analyses for this dataset could include time series, clustering, classification and more.

2. What were the question(s) that you set out to answer?

How many orders made by the customers?

Check TOP 5 most number of orders

How much money spent by the customers?

Check TOP 5 highest money spent

How many orders (per month)?

How many orders (per day)?

How many orders (per hour)?

Discover patterns for Unit Price

Discover patterns for each Country

How many orders for each country?

How much money spent by each country?

3. What visualization tool(s)/programming did you use?

We have used python programming language

4. Describe any transformations of the dataset that you needed to perform to get the data into the format needed by the visualization tool(s).

Data Cleaning

Check missing values for each column

Remove rows with missing values

Remove Quantity with negative values

Add the column - amount_spent

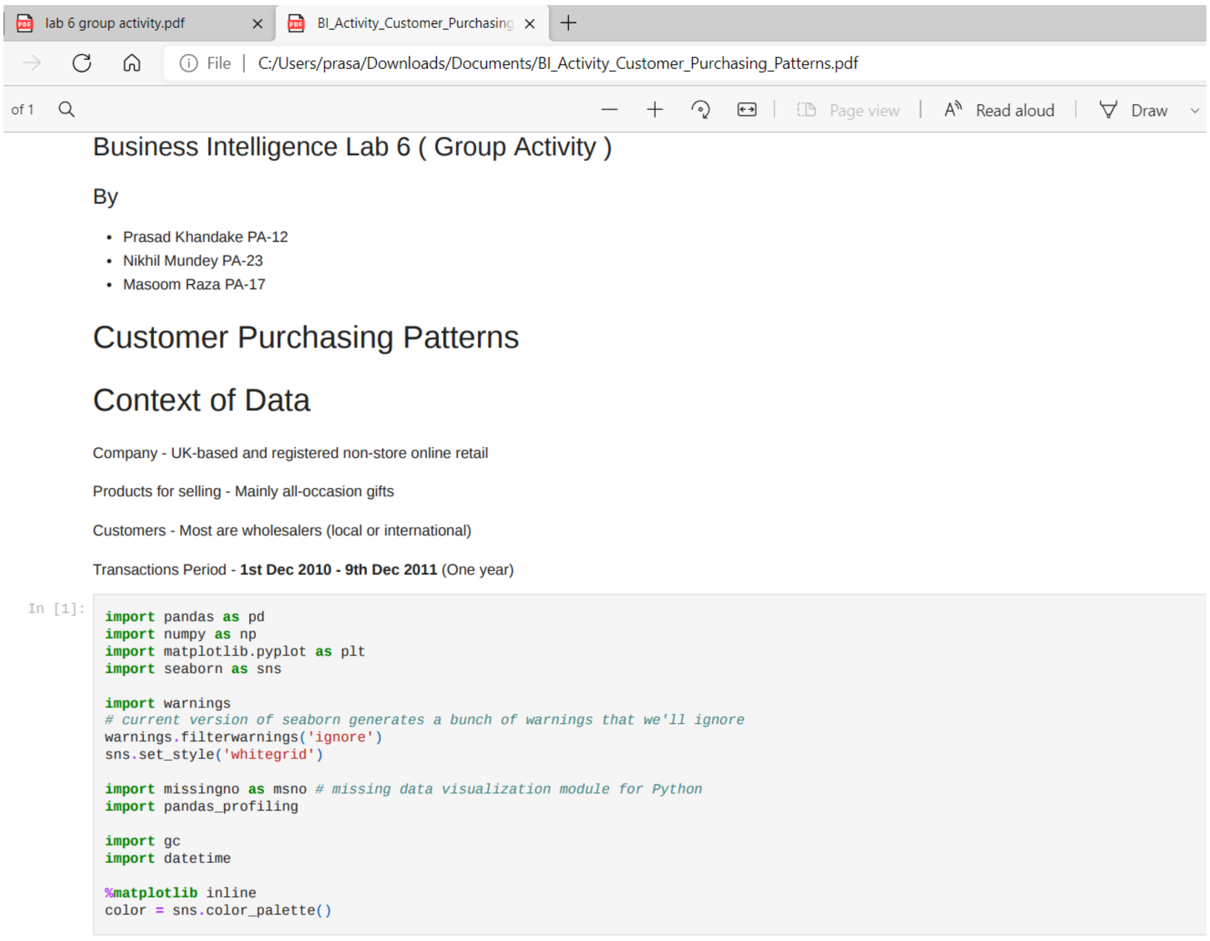
Add the columns - Month, Day and Hour for the invoice

5. Write a couple of paragraphs telling the story, describing the visualization, and saying how it answers the questions you posed.

Added in code and in Result section.

6. Include screenshots and brief explanations of some of your draft work to show your process and how you got to your final visualization.

Output: Link of your a story telling based on data.


lab 6 group activity.pdf
BI_Activity_Customer_Purchasing
File | C:/Users/prasa/Downloads/Documents/BI_Activity_Customer_Purchasing_Patterns.pdf
of 1
Page view
Read aloud
Draw

Business Intelligence Lab 6 (Group Activity)

By

- Prasad Khandake PA-12
- Nikhil Mundey PA-23
- Masoom Raza PA-17

Customer Purchasing Patterns

Context of Data

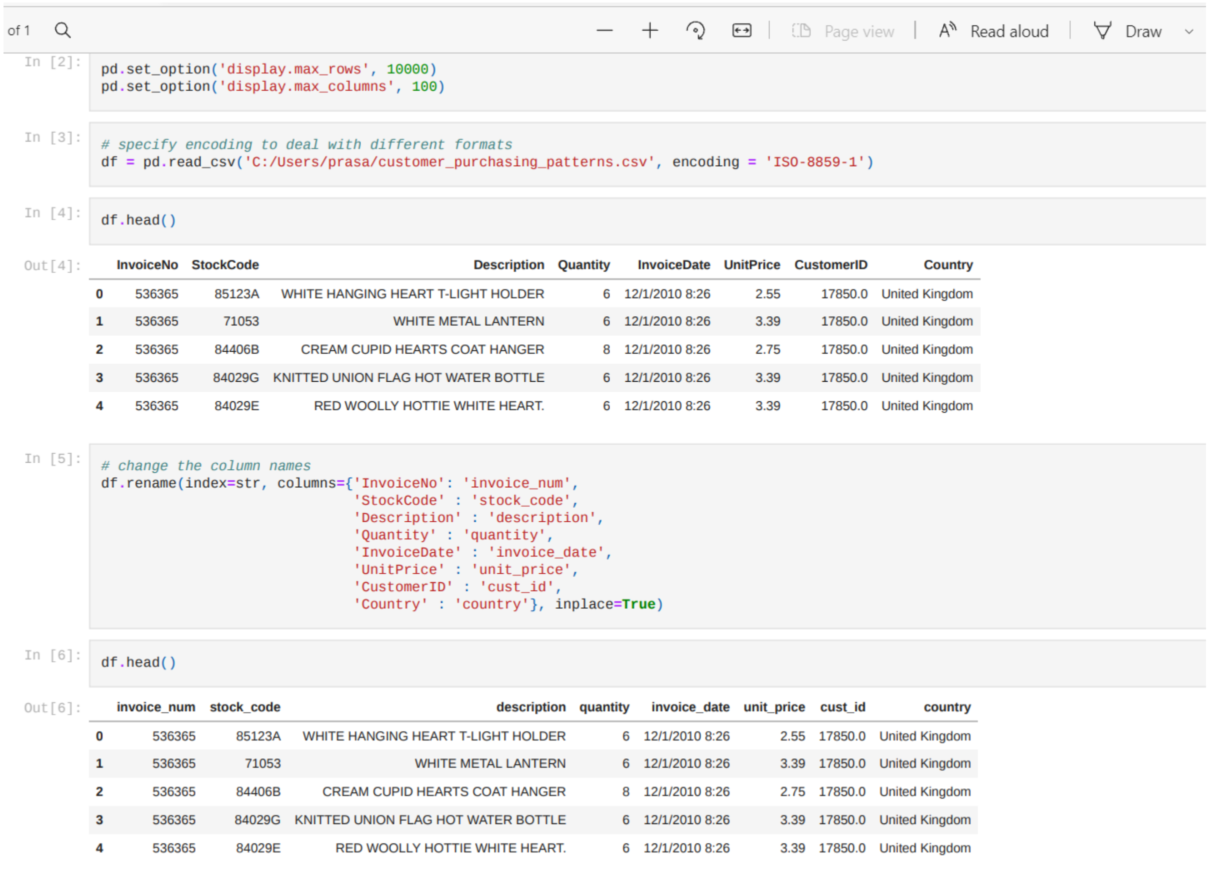
Company - UK-based and registered non-store online retail

Products for selling - Mainly all-occasion gifts

Customers - Most are wholesalers (local or international)

Transactions Period - **1st Dec 2010 - 9th Dec 2011** (One year)

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

import warnings
# current version of seaborn generates a bunch of warnings that we'll ignore
warnings.filterwarnings('ignore')
sns.set_style('whitegrid')

import missingno as msno # missing data visualization module for Python
import pandas_profiling

import gc
import datetime

%matplotlib inline
color = sns.color_palette()
```



In [2]:
```
pd.set_option('display.max_rows', 10000)
pd.set_option('display.max_columns', 100)
```

In [3]:
```
# specify encoding to deal with different formats
df = pd.read_csv('C:/Users/prasa/customer_purchasing_patterns.csv', encoding = 'ISO-8859-1')
```

In [4]:
```
df.head()
```

Out[4]:

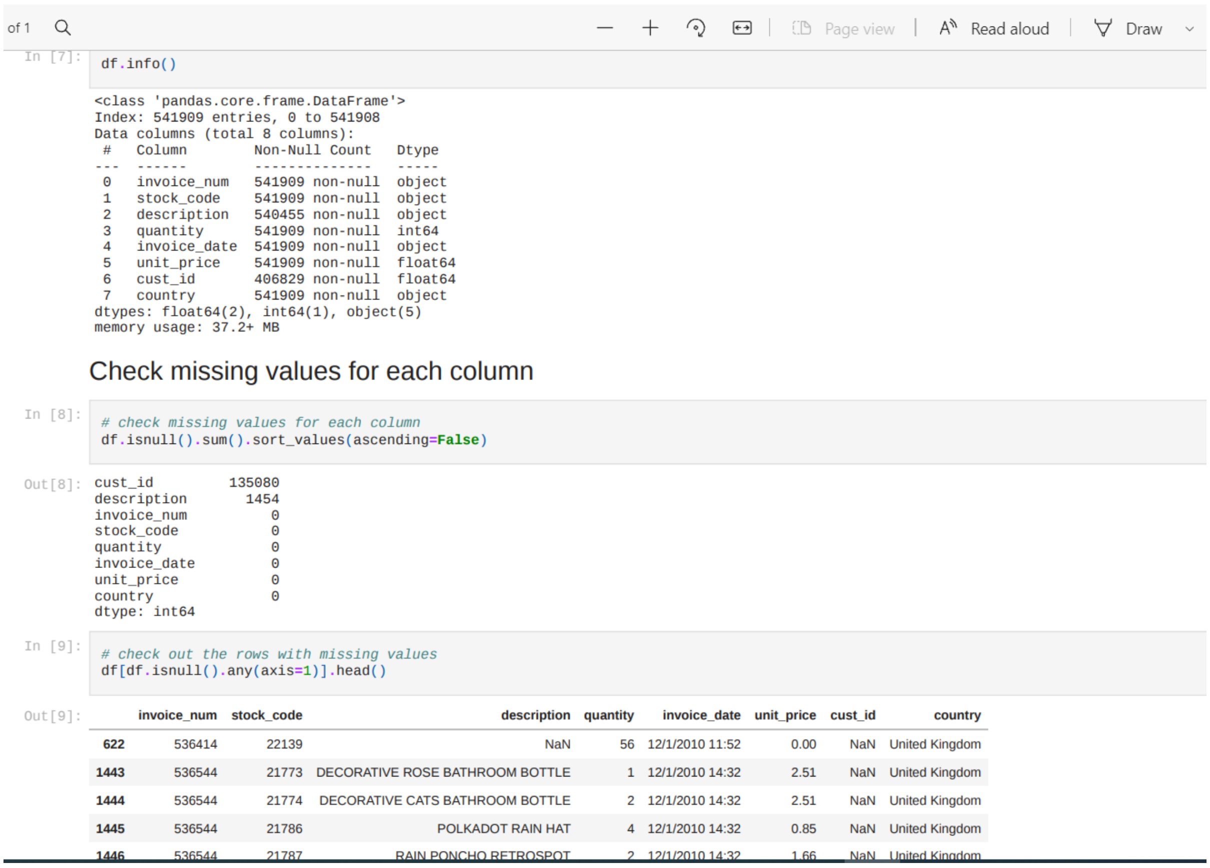
	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	12/1/2010 8:26	2.55	17850.0	United Kingdom
1	536365	71053	WHITE METAL LANTERN	6	12/1/2010 8:26	3.39	17850.0	United Kingdom
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	12/1/2010 8:26	2.75	17850.0	United Kingdom
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	12/1/2010 8:26	3.39	17850.0	United Kingdom
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	12/1/2010 8:26	3.39	17850.0	United Kingdom

In [5]:
```
# change the column names
df.rename(index=str, columns={'InvoiceNo': 'invoice_num',
                              'StockCode' : 'stock_code',
                              'Description' : 'description',
                              'Quantity' : 'quantity',
                              'InvoiceDate' : 'invoice_date',
                              'UnitPrice' : 'unit_price',
                              'CustomerID' : 'cust_id',
                              'Country' : 'country'}, inplace=True)
```

In [6]:
```
df.head()
```

Out[6]:

	invoice_num	stock_code	description	quantity	invoice_date	unit_price	cust_id	country
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	12/1/2010 8:26	2.55	17850.0	United Kingdom
1	536365	71053	WHITE METAL LANTERN	6	12/1/2010 8:26	3.39	17850.0	United Kingdom
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	12/1/2010 8:26	2.75	17850.0	United Kingdom
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	12/1/2010 8:26	3.39	17850.0	United Kingdom
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	12/1/2010 8:26	3.39	17850.0	United Kingdom



In [7]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 541909 entries, 0 to 541908
Data columns (total 8 columns):
 #   Column        Non-Null Count   Dtype  
---  ------        --------------   -----  
 0   invoice_num   541909 non-null  object 
 1   stock_code    541909 non-null  object 
 2   description   540455 non-null  object 
 3   quantity      541909 non-null  int64  
 4   invoice_date  541909 non-null  object 
 5   unit_price    541909 non-null  float64
 6   cust_id       406829 non-null  float64
 7   country       541909 non-null  object 
dtypes: float64(2), int64(1), object(5)
memory usage: 37.2+ MB
```

Check missing values for each column

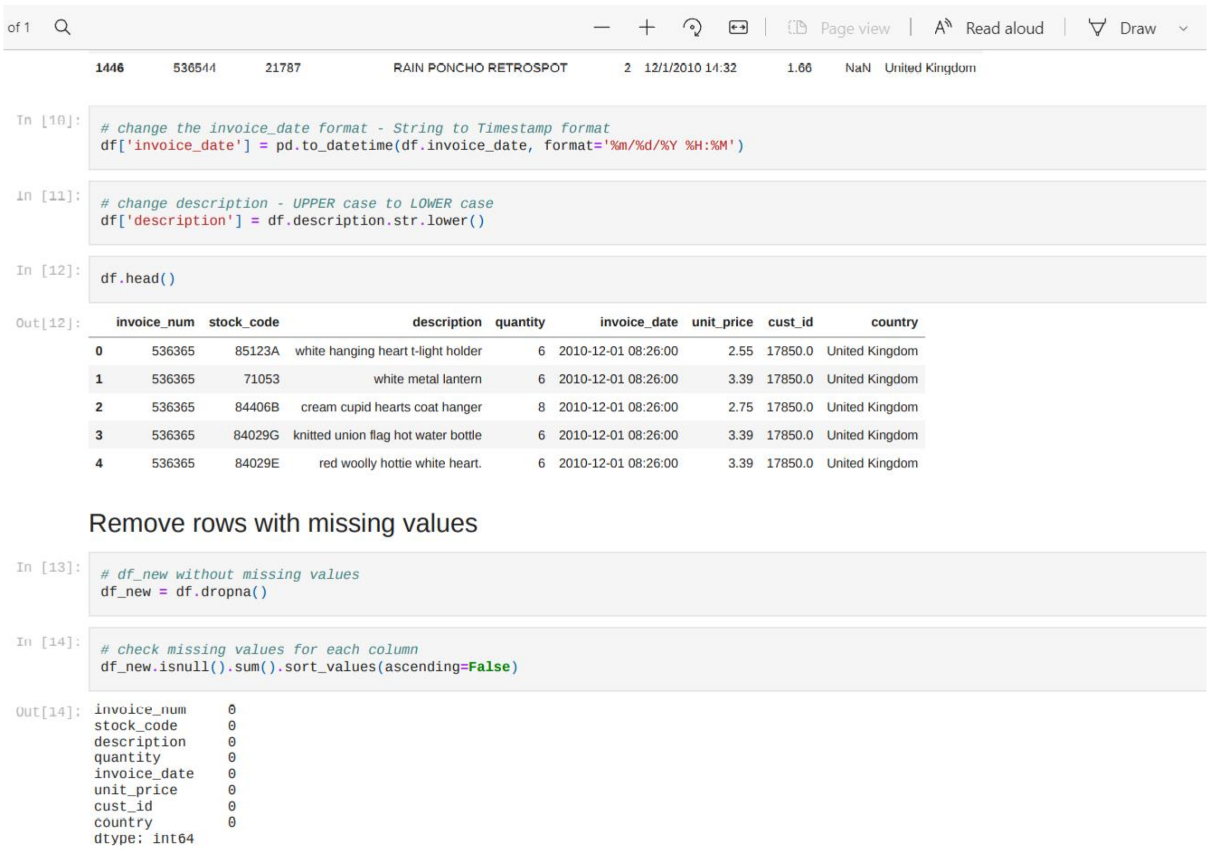
In [8]:
```
# check missing values for each column
df.isnull().sum().sort_values(ascending=False)
```

Out[8]:
```
cust_id         135080
description       1454
invoice_num          0
stock_code           0
quantity             0
invoice_date         0
unit_price           0
country              0
dtype: int64
```

In [9]:
```
# check out the rows with missing values
df[df.isnull().any(axis=1)].head()
```

Out[9]:

	invoice_num	stock_code	description	quantity	invoice_date	unit_price	cust_id	country
622	536414	22139	NaN	56	12/1/2010 11:52	0.00	NaN	United Kingdom
1443	536544	21773	DECORATIVE ROSE BATHROOM BOTTLE	1	12/1/2010 14:32	2.51	NaN	United Kingdom
1444	536544	21774	DECORATIVE CATS BATHROOM BOTTLE	2	12/1/2010 14:32	2.51	NaN	United Kingdom
1445	536544	21786	POLKADOT RAIN HAT	4	12/1/2010 14:32	0.85	NaN	United Kingdom
1446	536544	21787	RAIN PONCHO RETROSPOT	2	12/1/2010 14:32	1.66	NaN	United Kingdom



In [10]:
```
# change the invoice_date format - String to Timestamp format
df['invoice_date'] = pd.to_datetime(df.invoice_date, format='%m/%d/%Y %H:%M')
```

In [11]:
```
# change description - UPPER case to LOWER case
df['description'] = df.description.str.lower()
```

In [12]:
```
df.head()
```

Out[12]:

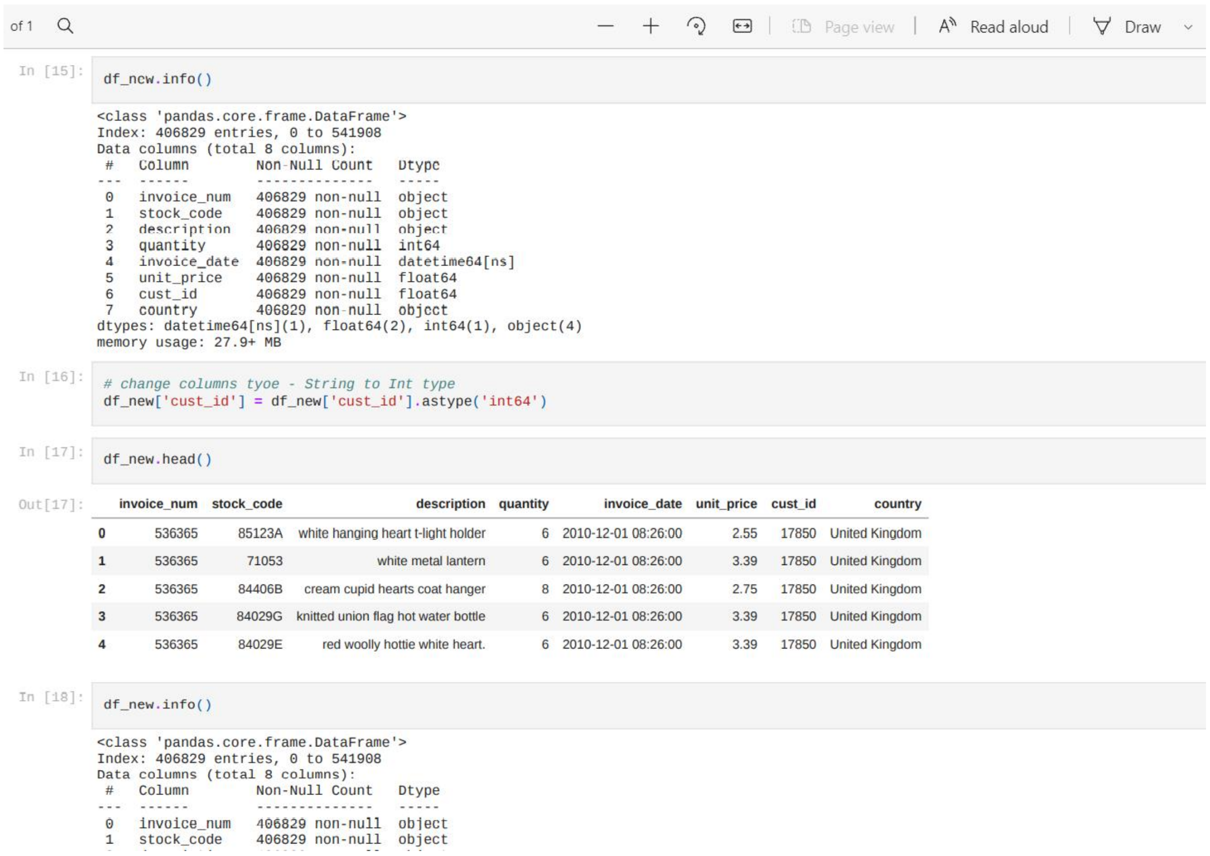
	invoice_num	stock_code	description	quantity	invoice_date	unit_price	cust_id	country
0	536365	85123A	white hanging heart t-light holder	6	2010-12-01 08:26:00	2.55	17850.0	United Kingdom
1	536365	71053	white metal lantern	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
2	536365	84406B	cream cupid hearts coat hanger	8	2010-12-01 08:26:00	2.75	17850.0	United Kingdom
3	536365	84029G	knitted union flag hot water bottle	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
4	536365	84029E	red woolly hottie white heart.	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom

Remove rows with missing values

In [13]:
```
# df_new without missing values
df_new = df.dropna()
```

In [14]:
```
# check missing values for each column
df_new.isnull().sum().sort_values(ascending=False)
```

Out[14]:
```
invoice_num     0
stock_code      0
description     0
quantity        0
invoice_date    0
unit_price      0
cust_id         0
country         0
dtype: int64
```



In [15]:
```
df_new.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 406829 entries, 0 to 541908
Data columns (total 8 columns):
 #   Column        Non-Null Count   Dtype
---  ------        --------------   -----
 0   invoice_num   406829 non-null  object
 1   stock_code    406829 non-null  object
 2   description   406829 non-null  object
 3   quantity      406829 non-null  int64
 4   invoice_date  406829 non-null  datetime64[ns]
 5   unit_price    406829 non-null  float64
 6   cust_id       406829 non-null  float64
 7   country       406829 non-null  object
dtypes: datetime64[ns](1), float64(2), int64(1), object(4)
memory usage: 27.9+ MB
```

In [16]:
```
# change columns tyoe - String to Int type
df_new['cust_id'] = df_new['cust_id'].astype('int64')
```

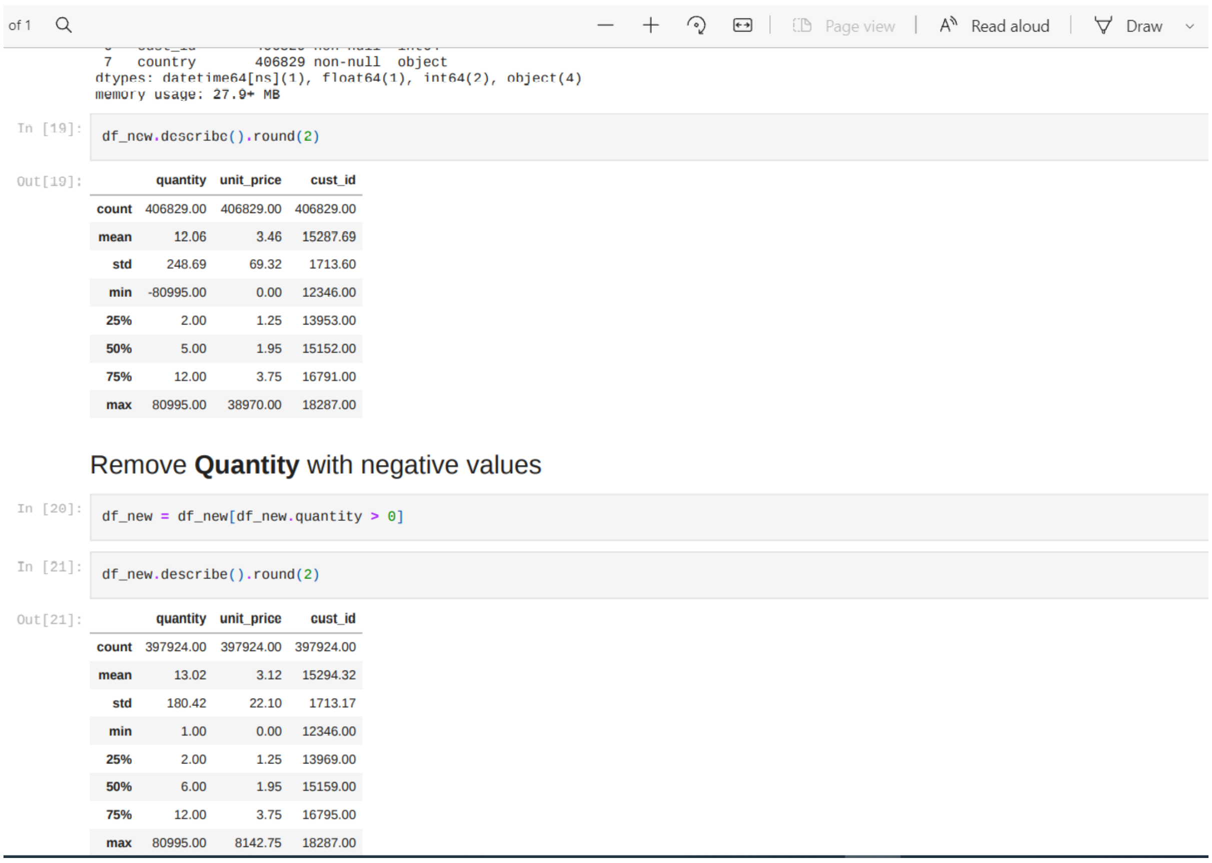
In [17]:
```
df_new.head()
```

Out[17]:

	invoice_num	stock_code	description	quantity	invoice_date	unit_price	cust_id	country
0	536365	85123A	white hanging heart t-light holder	6	2010-12-01 08:26:00	2.55	17850	United Kingdom
1	536365	71053	white metal lantern	6	2010-12-01 08:26:00	3.39	17850	United Kingdom
2	536365	84406B	cream cupid hearts coat hanger	8	2010-12-01 08:26:00	2.75	17850	United Kingdom
3	536365	84029G	knitted union flag hot water bottle	6	2010-12-01 08:26:00	3.39	17850	United Kingdom
4	536365	84029E	red woolly hottie white heart.	6	2010-12-01 08:26:00	3.39	17850	United Kingdom

In [18]:
```
df_new.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 406829 entries, 0 to 541908
Data columns (total 8 columns):
 #   Column        Non-Null Count   Dtype
---  ------        --------------   -----
 0   invoice_num   406829 non-null  object
 1   stock_code    406829 non-null  object
```



```
 7   country       406829 non-null  object
dtypes: datetime64[ns](1), float64(1), int64(2), object(4)
memory usage: 27.9+ MB
```

In [19]:
```
df_new.describe().round(2)
```

Out[19]:

	quantity	unit_price	cust_id
count	406829.00	406829.00	406829.00
mean	12.06	3.46	15287.69
std	248.69	69.32	1713.60
min	-80995.00	0.00	12346.00
25%	2.00	1.25	13953.00
50%	5.00	1.95	15152.00
75%	12.00	3.75	16791.00
max	80995.00	38970.00	18287.00

Remove **Quantity** with negative values

In [20]:
```
df_new = df_new[df_new.quantity > 0]
```

In [21]:
```
df_new.describe().round(2)
```

Out[21]:

	quantity	unit_price	cust_id
count	397924.00	397924.00	397924.00
mean	13.02	3.12	15294.32
std	180.42	22.10	1713.17
min	1.00	0.00	12346.00
25%	2.00	1.25	13969.00
50%	6.00	1.95	15159.00
75%	12.00	3.75	16795.00
max	80995.00	8142.75	18287.00

of 1 Page view Read aloud Draw Highlight

Add the column - amount_spent

In [22]:
```
df_new['amount_spent'] = df_new['quantity'] * df_new['unit_price']
```

In [23]:
```
# rearrange all the columns for easy reference
df_new = df_new[['invoice_num','invoice_date','stock_code','description','quantity','unit_price','amount_spent','cust_id','country']]
```

Add the columns - Month, Day and Hour for the invoice

In [24]:
```
df_new.insert(loc=2, column='year_month', value=df_new['invoice_date'].map(lambda x: 100*x.year + x.month))
df_new.insert(loc=3, column='month', value=df_new.invoice_date.dt.month)
# +1 to make Monday=1.....until Sunday=7
df_new.insert(loc=4, column='day', value=(df_new.invoice_date.dt.dayofweek)+1)
df_new.insert(loc=5, column='hour', value=df_new.invoice_date.dt.hour)
```

In [25]:
```
df_new.head()
```

Out[25]:

	invoice_num	invoice_date	year_month	month	day	hour	stock_code	description	quantity	unit_price	amount_spent	cust_id	country
0	536365	2010-12-01 08:26:00	201012	12	3	8	85123A	white hanging heart t-light holder	6	2.55	15.30	17850	United Kingdom
1	536365	2010-12-01 08:26:00	201012	12	3	8	71053	white metal lantern	6	3.39	20.34	17850	United Kingdom
2	536365	2010-12-01 08:26:00	201012	12	3	8	84406B	cream cupid hearts coat hanger	8	2.75	22.00	17850	United Kingdom
3	536365	2010-12-01 08:26:00	201012	12	3	8	84029G	knitted union flag hot water bottle	6	3.39	20.34	17850	United Kingdom
4	536365	2010-12-01 08:26:00	201012	12	3	8	84029E	red woolly hottie white heart.	6	3.39	20.34	17850	United Kingdom

Exploratory Data Analysis (EDA)

How many orders made by the customers?

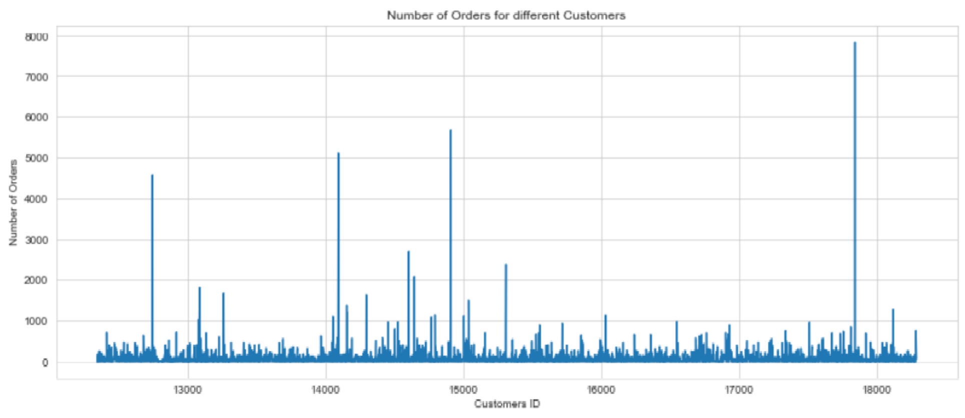
In [26]:
```
df_new.groupby(by=['cust_id','country'], as_index=False)['invoice_num'].count().head()
```

of 1 Page view Read aloud Draw

Out[26]:

	cust_id	country	invoice_num
0	12346	United Kingdom	1
1	12347	Iceland	182
2	12348	Finland	31
3	12349	Italy	73
4	12350	Norway	17

In [27]:
```
orders = df_new.groupby(by=['cust_id','country'], as_index=False)['invoice_num'].count()

plt.subplots(figsize=(15,6))
plt.plot(orders.cust_id, orders.invoice_num)
plt.xlabel('Customers ID')
plt.ylabel('Number of Orders')
plt.title('Number of Orders for different Customers')
plt.show()
```



of 1 Page view Read aloud Draw

Customers ID

Check TOP 5 most number of orders

In [28]:
```
print('The TOP 5 customers with most number of orders...')
orders.sort_values(by='invoice_num', ascending=False).head()
```

```
The TOP 5 customers with most number of orders...
```

Out[28]:

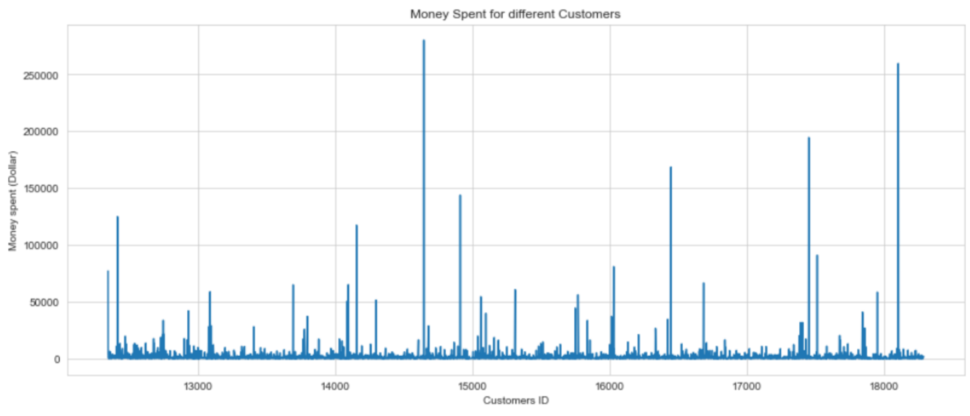
	cust_id	country	invoice_num
4019	17841	United Kingdom	7847
1888	14911	EIRE	5677
1298	14096	United Kingdom	5111
334	12748	United Kingdom	4596
1670	14606	United Kingdom	2700

How much money spent by the customers?

In [29]:
```
money_spent = df_new.groupby(by=['cust_id','country'], as_index=False)['amount_spent'].sum()

plt.subplots(figsize=(15,6))
plt.plot(money_spent.cust_id, money_spent.amount_spent)
plt.xlabel('Customers ID')
plt.ylabel('Money spent (Dollar)')
plt.title('Money Spent for different Customers')
plt.show()
```

of 1 Page view Read aloud Draw

```
plt.ylabel('Money spent (Dollar)')
plt.title('Money Spent for different Customers')
plt.show()
```



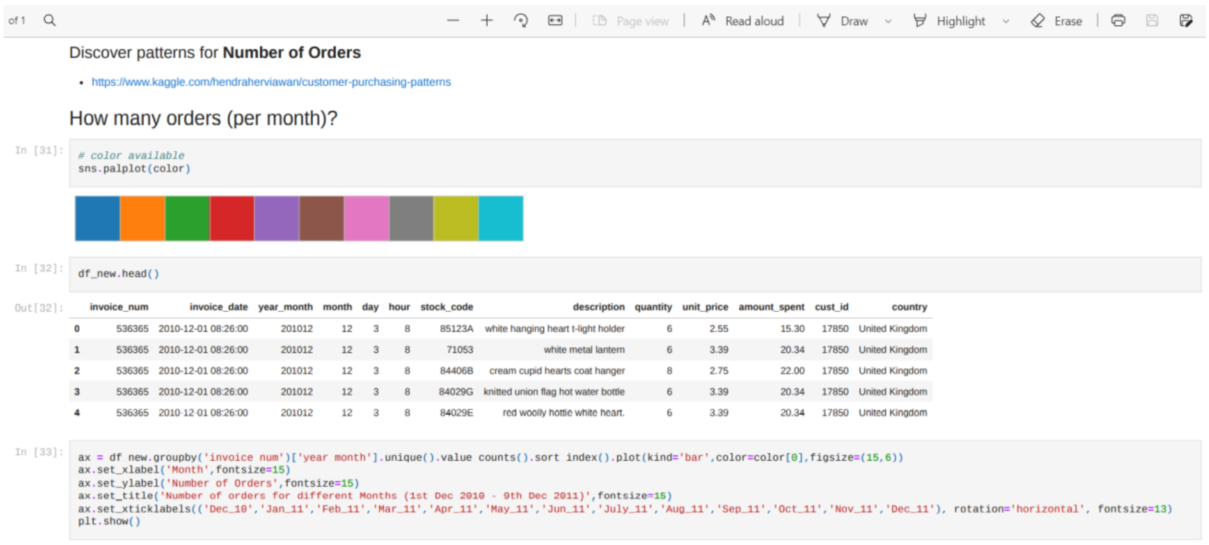
Check TOP 5 highest money spent

In [30]:
```
print('The TOP 5 customers with highest money spent...')
money_spent.sort_values(by='amount_spent', ascending=False).head()
```

```
The TOP 5 customers with highest money spent...
```

Out[30]:

	cust_id	country	amount_spent
1698	14646	Netherlands	280206.02
4210	18102	United Kingdom	259657.30
3737	17450	United Kingdom	194550.79
3017	16446	United Kingdom	168472.50
1888	14911	EIRE	143825.06



Discover patterns for **Number of Orders**

- https://www.kaggle.com/hendraherviawan/customer-purchasing-patterns

How many orders (per month)?

```
In [31]: # color available
         sns.palplot(color)
```

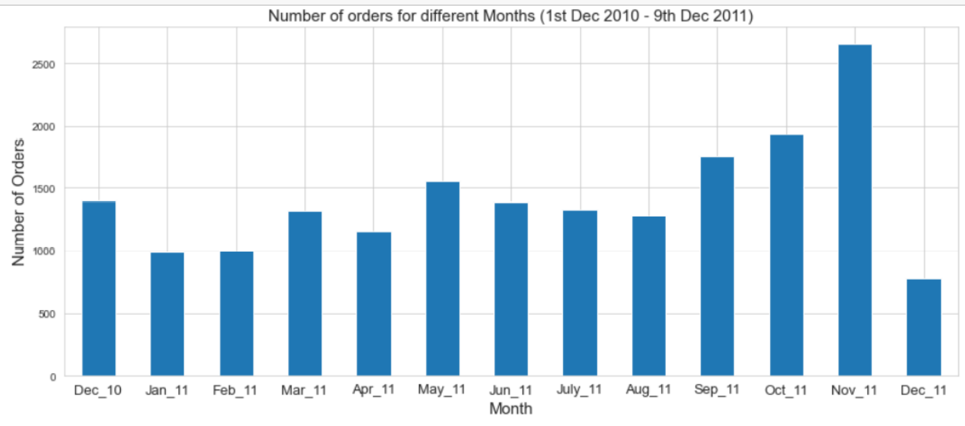
🟦🟧🟩🟥🟪🟫▨▨▨▨

```
In [32]: df_new.head()
```

Out[32]:

	invoice_num	invoice_date	year_month	month	day	hour	stock_code	description	quantity	unit_price	amount_spent	cust_id	country
0	536365	2010-12-01 08:26:00	201012	12	3	8	85123A	white hanging heart t-light holder	6	2.55	15.30	17850	United Kingdom
1	536365	2010-12-01 08:26:00	201012	12	3	8	71053	white metal lantern	6	3.39	20.34	17850	United Kingdom
2	536365	2010-12-01 08:26:00	201012	12	3	8	84406B	cream cupid hearts coat hanger	8	2.75	22.00	17850	United Kingdom
3	536365	2010-12-01 08:26:00	201012	12	3	8	84029G	knitted union flag hot water bottle	6	3.39	20.34	17850	United Kingdom
4	536365	2010-12-01 08:26:00	201012	12	3	8	84029E	red woolly hottie white heart.	6	3.39	20.34	17850	United Kingdom

```
In [33]: ax = df_new.groupby('invoice_num')['year_month'].unique().value_counts().sort_index().plot(kind='bar',color=color[0],figsize=(15,6))
         ax.set_xlabel('Month',fontsize=15)
         ax.set_ylabel('Number of Orders',fontsize=15)
         ax.set_title('Number of orders for different Months (1st Dec 2010 - 9th Dec 2011)',fontsize=15)
         ax.set_xticklabels(('Dec_10','Jan_11','Feb_11','Mar_11','Apr_11','May_11','Jun_11','July_11','Aug_11','Sep_11','Oct_11','Nov_11','Dec_11'), rotation='horizontal', fontsize=13)
         plt.show()
```



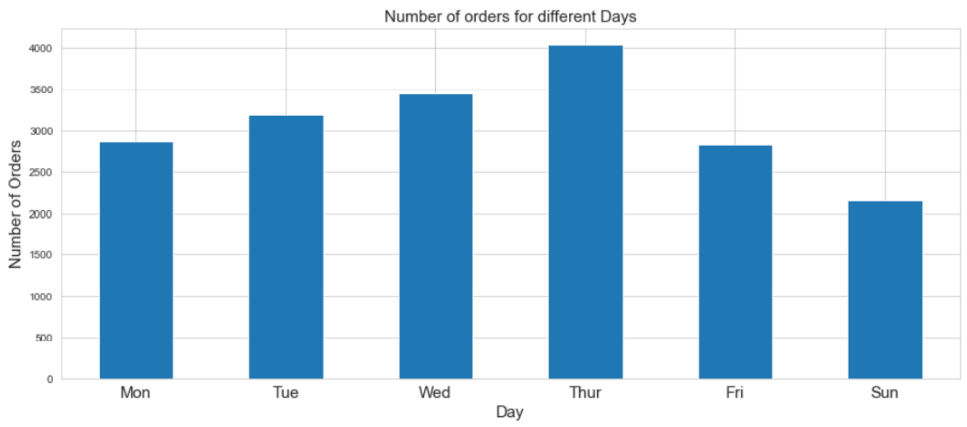
How many orders (per day)?

```
In [52]: df_new.groupby('invoice_num')['day'].unique().value_counts().sort_index()

---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
pandas\_libs\hashtable_class_helper.pxi in pandas._libs.hashtable.PyObjectHashTable.map_locations()

TypeError: unhashable type: 'numpy.ndarray'
Exception ignored in: 'pandas._libs.index.IndexEngine._call_map_locations'
Traceback (most recent call last):
  File "pandas\_libs\hashtable_class_helper.pxi", line 4588, in pandas._libs.hashtable.PyObjectHashTable.map_locations
TypeError: unhashable type: 'numpy.ndarray'

Out[52]: [1]    2863
         [2]    3185
         [3]    3455
         [4]    4033
         [5]    2831
         [7]    2169
         Name: day, dtype: int64
```

of 1 Page view | Read aloud | Draw

```
Name: day, dtype: int64
```

In [35]:
```
ax = df_new.groupby('invoice_num')['day'].unique().value_counts().sort_index().plot(kind='bar',color=color[0],figsize=(15,6))
ax.set_xlabel('Day',fontsize=15)
ax.set_ylabel('Number of Orders',fontsize=15)
ax.set_title('Number of orders for different Days',fontsize=15)
ax.set_xticklabels(('Mon','Tue','Wed','Thur','Fri','Sun'), rotation='horizontal', fontsize=15)
plt.show()
```



of 1 Page view | Read aloud | Draw

How many orders (per hour)?

In [36]:
```
#df_new.groupby('invoice_num')['hour'].value_counts().iloc[:-1].sort_index()
```

In [37]:
```
df_new.groupby('invoice_num')['hour'].agg('count').value_counts().iloc[:-1].sort_index()
# .unique()
```

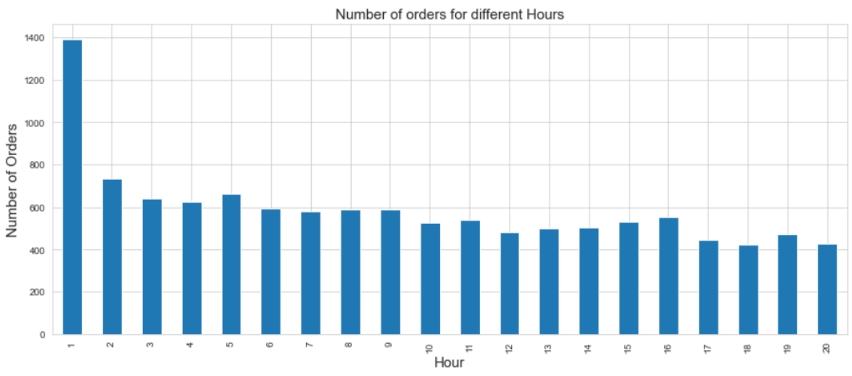
Out[37]:
```
1       1392
2        735
3        638
4        626
5        659
6        595
7        581
8        588
9        589
10       525
11       540
12       480
13       500
14       503
15       531
16       555
17       444
18       423
19       472
20       428
21       393
22       332
23       348
24       312
25       240
26       236
27       227
28       219
29       278
30       209
31       192
32       173
33       160
34       157
35       135
36       116
37       128
38       114
```

of 1 Page view Read aloud Draw Highlight

```
Name: hour, dtype: int64
```

In [39]:

```
ax = df_new.groupby('invoice_num')['hour'].agg('count').value_counts().iloc[:-1].sort_index().head(20).plot(kind='bar',color=color[0],figsize=(15,6))
ax.set_xlabel('Hour',fontsize=15)
ax.set_ylabel('Number of Orders',fontsize=15)
ax.set_title('Number of orders for different Hours',fontsize=15)
#ax.set_xticklabels(range(6,21), rotation='horizontal', fontsize=15)
plt.show()
#
```



of 1 Page view Read aloud Draw

Discover patterns for **Unit Price**

In [40]:

```
df_new.unit_price.describe()
```

Out[40]:

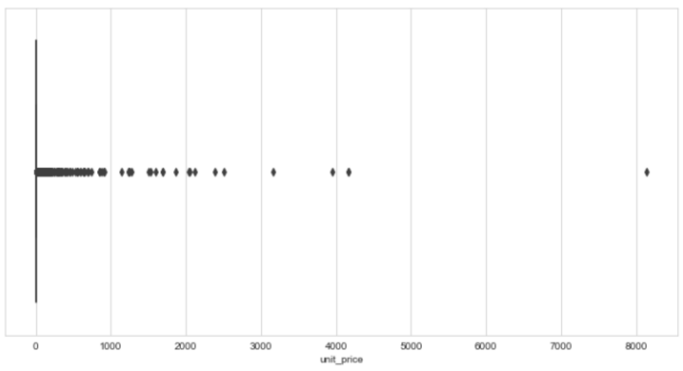
```
count    397924.000000
mean          3.116174
std          22.096788
min           0.000000
25%           1.250000
50%           1.950000
75%           3.750000
max        8142.750000
Name: unit_price, dtype: float64
```

We see that there are unit price = 0 (FREE items)

There are some free items given to customers from time to time.

In [41]:

```
# check the distribution of unit price
plt.subplots(figsize=(12,6))
sns.boxplot(df_new.unit_price)
plt.show()
```



of 1 Page view Read aloud Draw Highlight Erase

In [42]:
```
df_free = df_new[df_new.unit_price == 0]
```

In [43]:
```
df_free.head()
```

Out[43]:

	invoice_num	invoice_date	year_month	month	day	hour	stock_code	description	quantity	unit_price	amount_spent	cust_id	country
9302	537197	2010-12-05 14:02:00	201012	12	7	14	22841	round cake tin vintage green	1	0.0	0.0	12647	Germany
33576	539263	2010-12-16 14:36:00	201012	12	4	14	22580	advent calendar gingham sack	4	0.0	0.0	16560	United Kingdom
40089	539722	2010-12-21 13:45:00	201012	12	2	13	22423	regency cakestand 3 tier	10	0.0	0.0	14911	EIRE
47068	540372	2011-01-06 16:41:00	201101	1	4	16	22090	paper bunting retrospot	24	0.0	0.0	13081	United Kingdom
47070	540372	2011-01-06 16:41:00	201101	1	4	16	22553	plasters in tin skulls	24	0.0	0.0	13081	United Kingdom

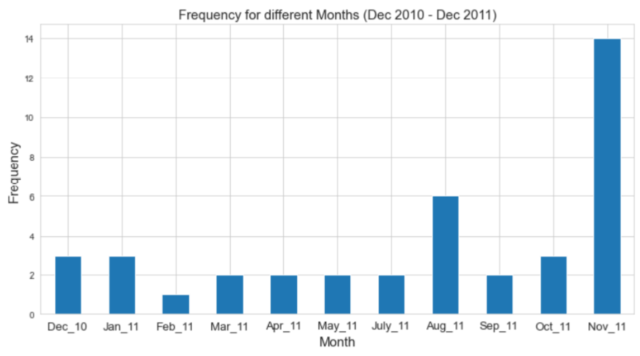
In [44]:
```
df_free.year_month.value_counts().sort_index()
```

Out[44]:
```
201012     3
201101     3
201102     1
201103     2
201104     2
201105     2
201107     2
201108     6
201109     2
201110     3
201111    14
Name: year_month, dtype: int64
```

of 1 Page view Read aloud Draw Highlight Erase

```
Name: year_month, dtype: int64
```

In [45]:
```
ax = df_free.year_month.value_counts().sort_index().plot(kind='bar',figsize=(12,6), color=color[0])
ax.set_xlabel('Month',fontsize=15)
ax.set_ylabel('Frequency',fontsize=15)
ax.set_title('Frequency for different Months (Dec 2010 - Dec 2011)',fontsize=15)
ax.set_xticklabels(('Dec_10','Jan_11','Feb_11','Mar_11','Apr_11','May_11','July_11','Aug_11','Sep_11','Oct_11','Nov_11'), rotation='horizontal', fontsize=13)
plt.show()
```



Not clear why there are FREE items given to certain customers

- On average, the company gave out 2-4 times FREE items to customers each month (Except in June 2011)

Discover patterns for each **Country**

In [46]:
```
df_new.head()
```

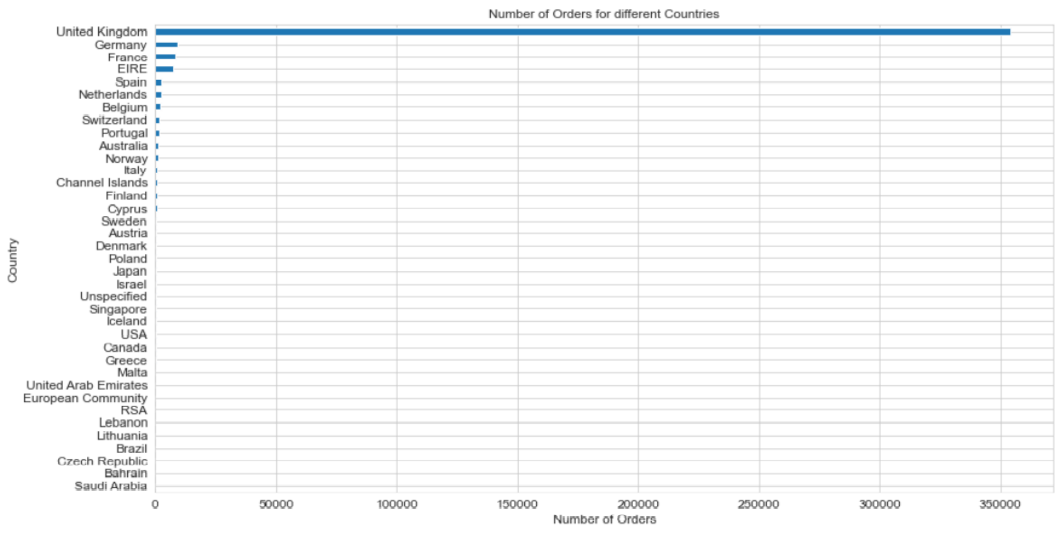
Out[46]:

	invoice_num	invoice_date	year_month	month	day	hour	stock_code	description	quantity	unit_price	amount_spent	cust_id	country
0	536365	2010-12-01 08:26:00	201012	12	3	8	85123A	white hanging heart t-light holder	6	2.55	15.30	17850	United Kingdom
1	536365	2010-12-01 08:26:00	201012	12	3	8	71053	white metal lantern	6	3.39	20.34	17850	United Kingdom
2	536365	2010-12-01 08:26:00	201012	12	3	8	84406B	cream cupid hearts coat hanger	8	2.75	22.00	17850	United Kingdom
3	536365	2010-12-01 08:26:00	201012	12	3	8	84029G	knitted union flag hot water bottle	6	3.39	20.34	17850	United Kingdom
4	536365	2010-12-01 08:26:00	201012	12	3	8	84029E	red woolly hottie white heart.	6	3.39	20.34	17850	United Kingdom

of 1 Page view | Read aloud | Draw

How many orders for each country?

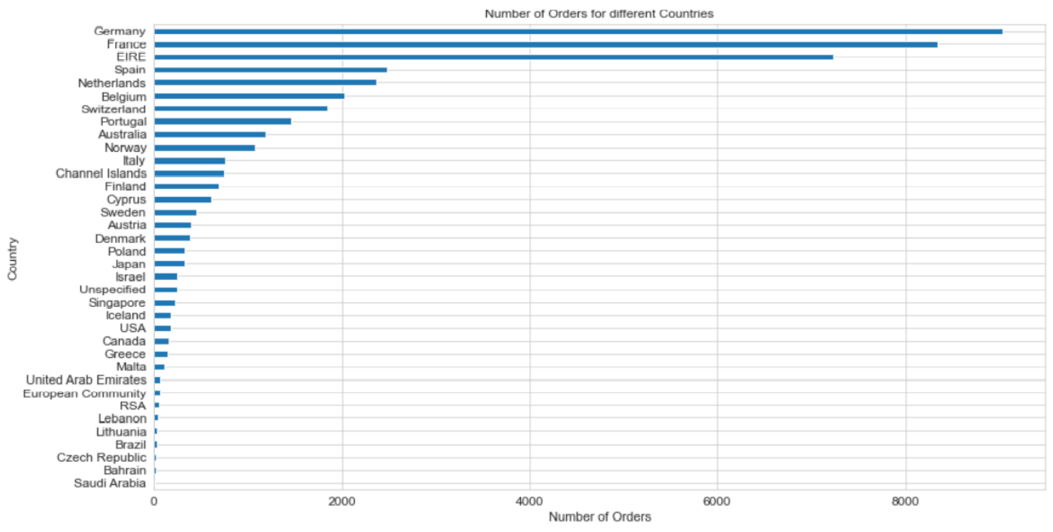
In [47]:
```python
group_country_orders = df_new.groupby('country')['invoice_num'].count().sort_values()
# del group_country_orders['United Kingdom']

# plot number of unique customers in each country (with UK)
plt.subplots(figsize=(15,8))
group_country_orders.plot(kind='barh', fontsize=12, color=color[0])
plt.xlabel('Number of Orders', fontsize=12)
plt.ylabel('Country', fontsize=12)
plt.title('Number of Orders for different Countries', fontsize=12)
plt.show()
```



of 1 Page view | Read aloud | Draw

Number of Orders

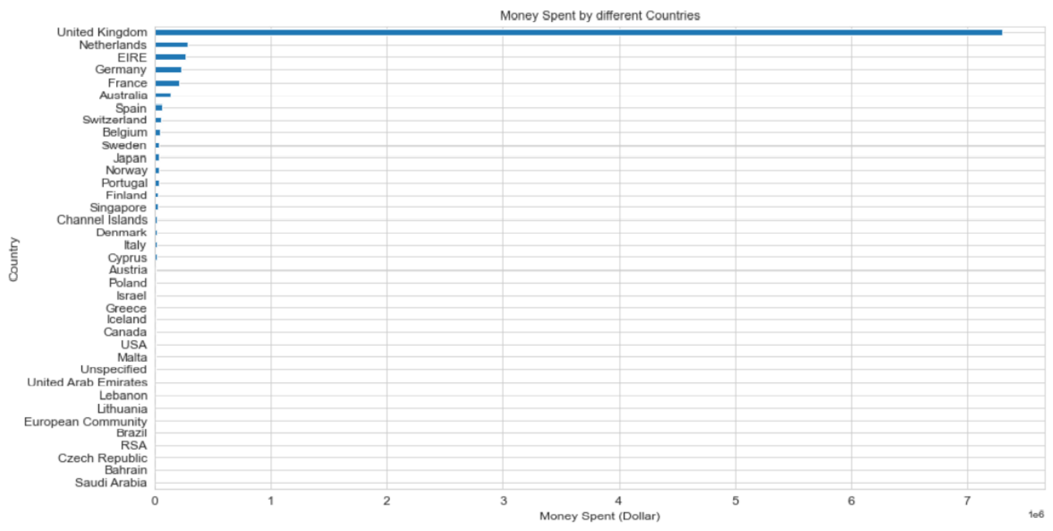
In [48]:
```python
group_country_orders = df_new.groupby('country')['invoice_num'].count().sort_values()
del group_country_orders['United Kingdom']

# plot number of unique customers in each country (without UK)
plt.subplots(figsize=(15,8))
group_country_orders.plot(kind='barh', fontsize=12, color=color[0])
plt.xlabel('Number of Orders', fontsize=12)
plt.ylabel('Country', fontsize=12)
plt.title('Number of Orders for different Countries', fontsize=12)
plt.show()
```



of 1 Page view Read aloud Draw

How much money spent by each country?

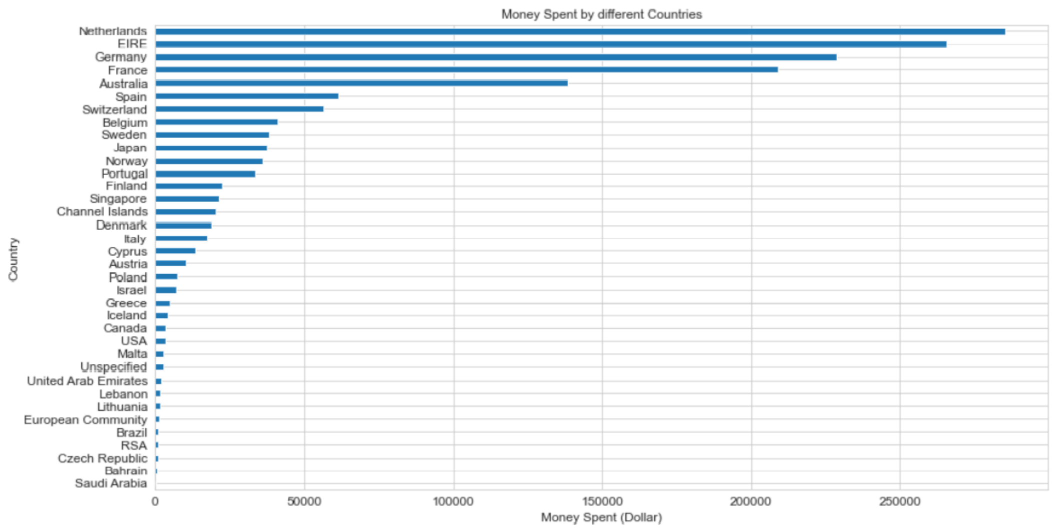
In [49]:
```python
group_country_amount_spent = df_new.groupby('country')['amount_spent'].sum().sort_values()
# del group_country_orders['United Kingdom']

# plot number of unique customers in each country (with UK)
plt.subplots(figsize=(15,8))
group_country_amount_spent.plot(kind='barh', fontsize=12, color=color[0])
plt.xlabel('Money Spent (Dollar)', fontsize=12)
plt.ylabel('Country', fontsize=12)
plt.title('Money Spent by different Countries', fontsize=12)
plt.show()
```

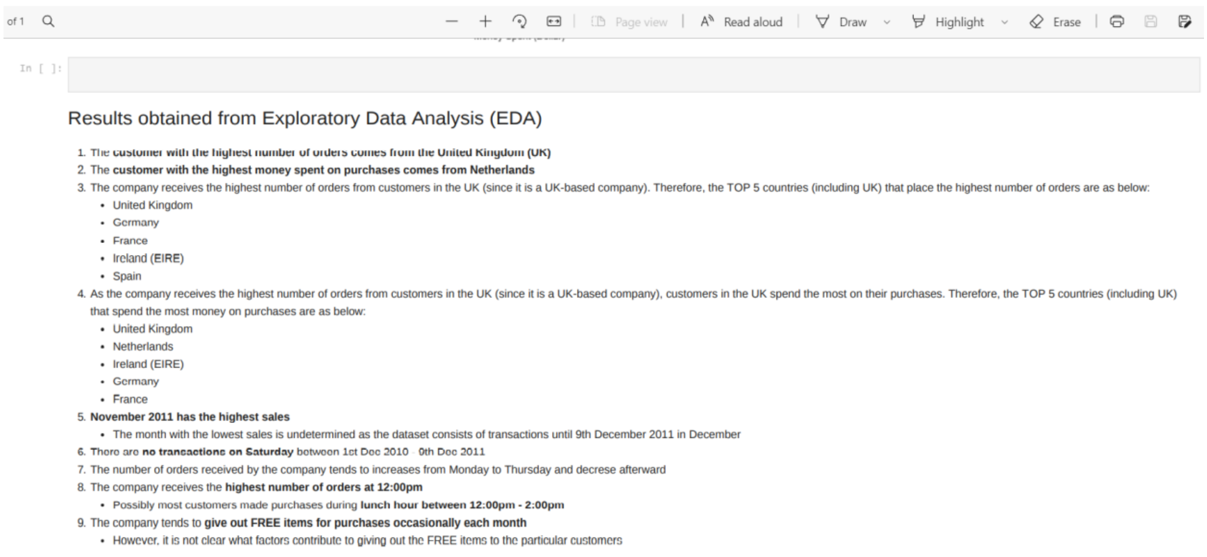


of 1 Page view Read aloud Draw

Money Spent (Dollar) 1e6

In [50]:
```python
group_country_amount_spent = df_new.groupby('country')['amount_spent'].sum().sort_values()
del group_country_amount_spent['United Kingdom']

# plot number of unique customers in each country (without UK)
plt.subplots(figsize=(15,8))
group_country_amount_spent.plot(kind='barh', fontsize=12, color=color[0])
plt.xlabel('Money Spent (Dollar)', fontsize=12)
plt.ylabel('Country', fontsize=12)
plt.title('Money Spent by different Countries', fontsize=12)
plt.show()
```





Results obtained from Exploratory Data Analysis (EDA)

1. The **customer with the highest number of orders comes from the United Kingdom (UK)**
2. The **customer with the highest money spent on purchases comes from Netherlands**
3. The company receives the highest number of orders from customers in the UK (since it is a UK-based company). Therefore, the TOP 5 countries (including UK) that place the highest number of orders are as below:
   - United Kingdom
   - Germany
   - France
   - Ireland (EIRE)
   - Spain
4. As the company receives the highest number of orders from customers in the UK (since it is a UK-based company), customers in the UK spend the most on their purchases. Therefore, the TOP 5 countries (including UK) that spend the most money on purchases are as below:
   - United Kingdom
   - Netherlands
   - Ireland (EIRE)
   - Germany
   - France
5. **November 2011 has the highest sales**
   - The month with the lowest sales is undetermined as the dataset consists of transactions until 9th December 2011 in December
6. There are **no transactions on Saturday** between 1st Dec 2010 - 9th Dec 2011
7. The number of orders received by the company tends to increases from Monday to Thursday and decrese afterward
8. The company receives the **highest number of orders at 12:00pm**
   - Possibly most customers made purchases during **lunch hour between 12:00pm - 2:00pm**
9. The company tends to **give out FREE items for purchases occasionally each month**
   - However, it is not clear what factors contribute to giving out the FREE items to the particular customers

Results obtained from Exploratory Data Analysis (EDA)

1. The **customer with the highest number of orders comes from the United Kingdom (UK)**

2. The **customer with the highest money spent on purchases comes from Netherlands**

3. The company receives the highest number of orders from customers in the UK (since it is a UK-based company). Therefore, the TOP 5 countries (including UK) that place the highest number of orders are as below:

- United Kingdom

    - Germany

    - France

    - Ireland (EIRE)

    - Spain

4. As the company receives the highest number of orders from customers in the UK (since it is a UK-based company), customers in the UK spend the most on their purchases. Therefore, the TOP 5 countries (including UK) that spend the most money on purchases are as below:

- United Kingdom

    - Netherlands

    - Ireland (EIRE)

    - Germany

    - France

5. **November 2011 has the highest sales**

- The month with the lowest sales is undetermined as the dataset consists of transactions until 9th December 2011 in December

6. There are **no transactions on Saturday** between 1st Dec 2010 - 9th Dec 2011

7. The number of orders received by the company tends to increases from Monday to Thursday and decrese afterward

8. The company receives the **highest number of orders at 12:00pm**

- Possibly most customers made purchases during **lunch hour between 12:00pm - 2:00pm**

9. The company tends to **give out FREE items for purchases occasionally each month**

- However, it is not clear what factors contribute to giving out the FREE items to the particular customers.

Conclusion:

Hence, understood better the process of using visualizations to tell a story about a data set.