In [2]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
import datetime as datetime
import dateutil.parser
```

In [4]:
```
#importing the dataset
df = pd.read_csv('sales.csv')
df
```

Out[4]:

	date	warehouse	client_type	product_line	quantity	unit_price	total	payment
0	2021-06-01	Central	Retail	Miscellaneous	8	16.85	134.83	Credit card
1	2021-06-01	North	Retail	Breaking system	9	19.29	173.61	Cash
2	2021-06-01	North	Retail	Suspension & traction	8	32.93	263.45	Credit card
3	2021-06-01	North	Wholesale	Frame & body	16	37.84	605.44	Transfer
4	2021-06-01	Central	Retail	Engine	2	60.48	120.96	Credit card
...	...	...	...	...	...	...	...	...
995	2021-08-28	Central	Retail	Electrical system	9	32.87	295.83	Credit card
996	2021-08-	West	Wholesale	Breaking system	32	10.03	320.96	Transfer

In [5]:
```
#checking the first few rows
df.head()
```

Out[5]:

	date	warehouse	client_type	product_line	quantity	unit_price	total	payment
0	2021-06-01	Central	Retail	Miscellaneous	8	16.85	134.83	Credit card
1	2021-06-01	North	Retail	Breaking system	9	19.29	173.61	Cash
2	2021-06-01	North	Retail	Suspension & traction	8	32.93	263.45	Credit card
3	2021-06-01	North	Wholesale	Frame & body	16	37.84	605.44	Transfer
4	2021-06-01	Central	Retail	Engine	2	60.48	120.96	Credit card

In [6]:
```
#checking variable type
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 8 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   date          1000 non-null   object 
 1   warehouse     1000 non-null   object 
 2   client_type   1000 non-null   object 
 3   product_line  1000 non-null   object 
 4   quantity      1000 non-null   int64  
 5   unit_price    1000 non-null   float64
 6   total         1000 non-null   float64
 7   payment       1000 non-null   object 
dtypes: float64(2), int64(1), object(5)
memory usage: 62.6+ KB
```

In [7]:
```
#convert the date column to a datetime object
time_col='date'
df[time_col] = pd.to_datetime(df[time_col])
```

In [8]:
```
df.date.dtype
```

Out[8]:
```
dtype('<M8[ns]')
```

In [9]:
```
#checking for missing value
df.isnull().sum()
```

Out[9]:
```
date            0
warehouse       0
client_type     0
product_line    0
quantity        0
unit_price      0
total           0
payment         0
dtype: int64
```

1. What are the total sales for each payment method?

In [10]:
```
total_sales = df.groupby('payment', as_index=False).sum('total')
total_sales
```

Out[10]:

	payment	quantity	unit_price	total
0	Cash	627	3479.98	19199.10
1	Credit card	3588	19992.33	110271.57
2	Transfer	5180	6849.73	159642.33

3. What is the average unit price for each product line?

In [11]:
```
avg_unit_price = df.groupby('product_line', as_index= False).mean('unit_price')
```

In [12]:
```
avg_unit_price
```

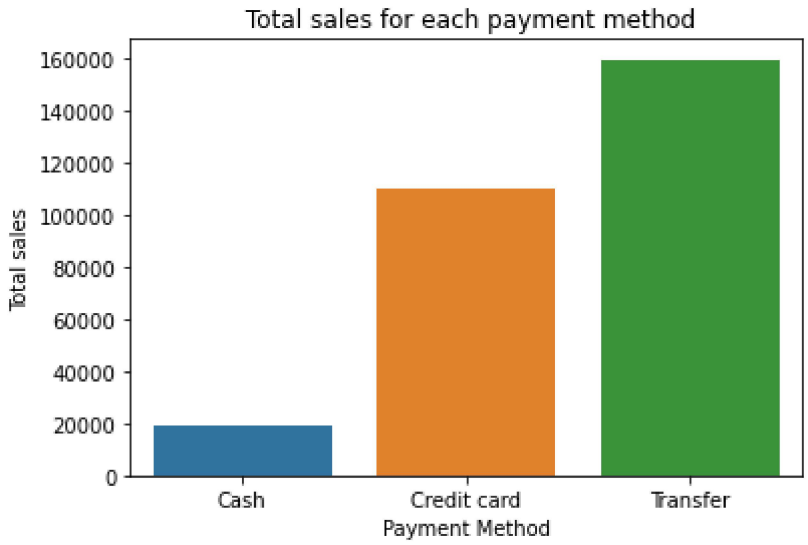
Out[12]:

	product_line	quantity	unit_price	total
0	Breaking system	9.260870	17.740522	166.739783
1	Electrical system	8.797927	25.585130	225.972591
2	Engine	10.278689	60.091803	622.055410
3	Frame & body	9.753012	42.832229	415.811627
4	Miscellaneous	9.639344	22.810738	222.670656
5	Suspension & traction	9.407895	33.969868	320.237763

4. Create plots to visualize findings for questions 1 and 2.

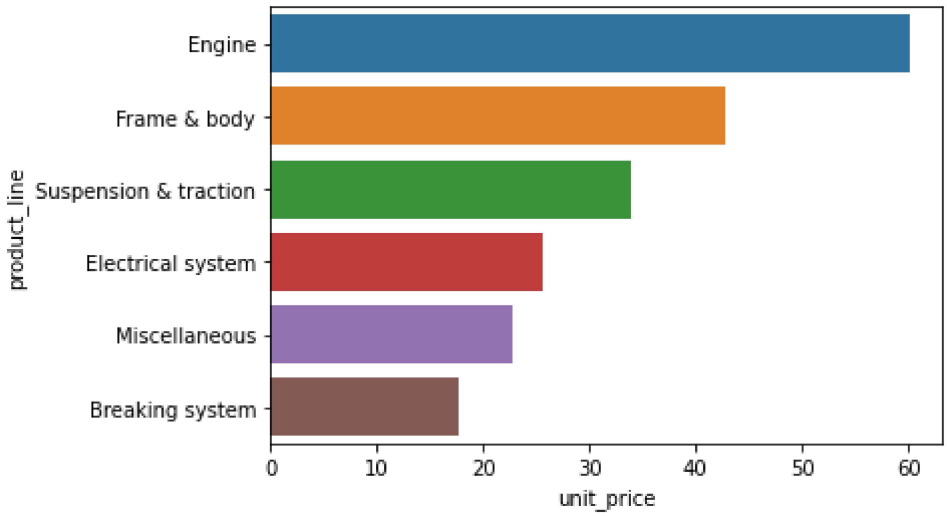
In [13]:
```
#creating barchart to visualize total sales for each product method
sns.barplot(x='payment', y='total', data=total_sales )
plt.title(' Total sales for each payment method')
plt.xlabel('Payment Method')
plt.ylabel('Total sales')
```

Out[13]:
```
Text(0, 0.5, 'Total sales')
```



In [14]:
```
#creating barchat to visualize the average unit price for each product line
sns.barplot(data =avg_unit_price.sort_values('unit_price', ascending=False),
            x='unit_price', y='product_line', )
```

Out[14]: <AxesSubplot:xlabel='unit_price', ylabel='product_line'>



5. Investigate further (e.g., average purchase value by client type, total purchase value by product line, etc.)

In [23]:
```
#average purchase value
avg_purchase =df.groupby('client_type', as_index=False).mean('total')
avg_purchase
```

Out[23]:

	client_type	quantity	unit_price	total
0	Retail	5.438710	30.286852	167.058929
1	Wholesale	23.022222	30.443244	709.521467

In [26]:
```
#total purchase value by product line
total_purchase_client = df.groupby('product_line').sum('total')
total_purchase_client
```

Out[26]:

product_line	quantity	unit_price	total
Breaking system	2130	4080.32	38350.15
Electrical system	1698	4937.93	43612.71
Engine	627	3665.60	37945.38
Frame & body	1619	7110.15	69024.73
Miscellaneous	1176	2782.91	27165.82
Suspension & traction	2145	7745.13	73014.21

In [24]:
```
#most used warehouse
df.groupby('payment').count()
```

Out[24]:

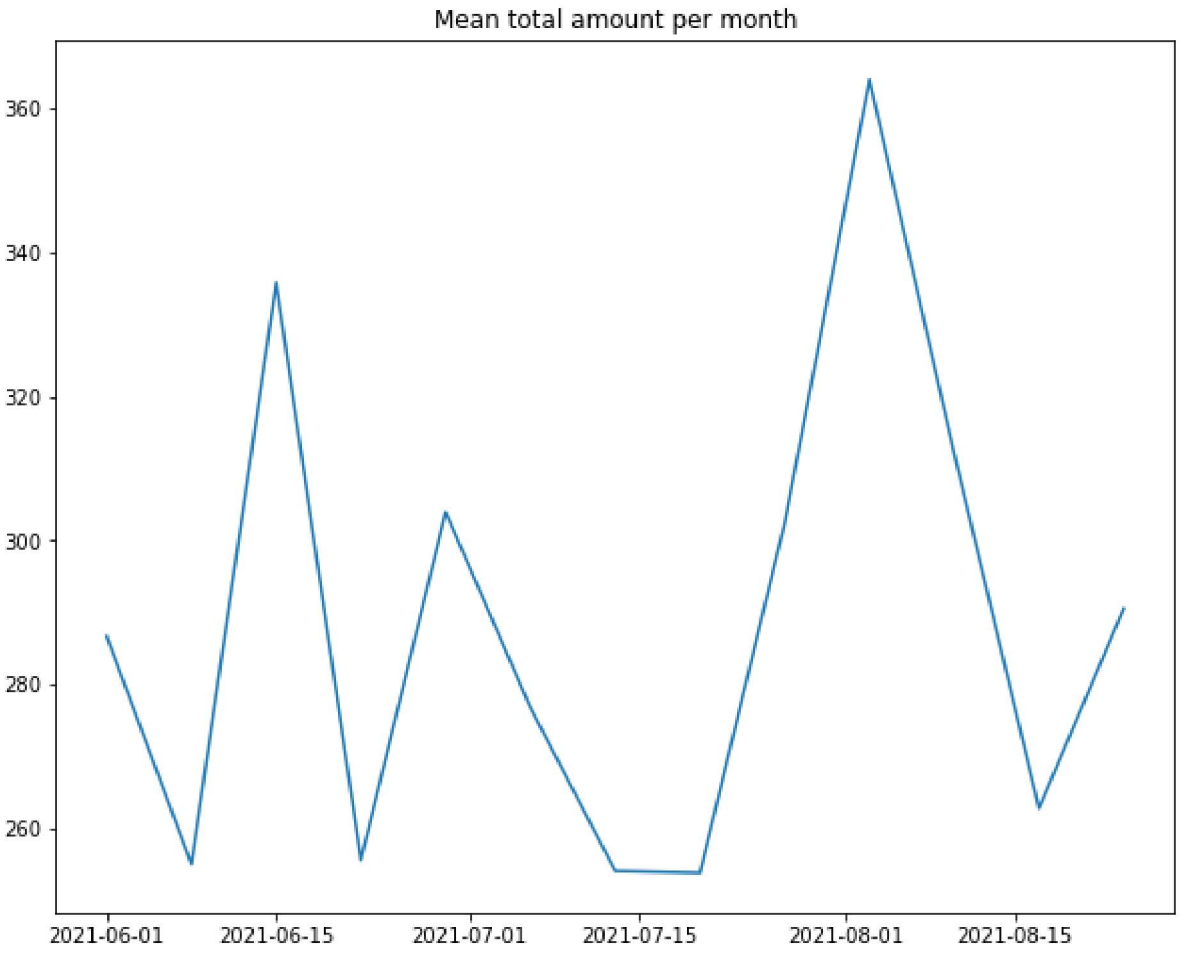
payment	date	warehouse	client_type	product_line	quantity	unit_price	total
Cash	116	116	116	116	116	116	116
Credit card	659	659	659	659	659	659	659
Transfer	225	225	225	225	225	225	225

In [17]:
```
#resampling the date to monthly intervals, taking the mean of the total amount fo

time_date = df.resample('M', on=time_col)['total'].mean().fillna(0)

time_date
```

Out[17]:
```
date
2021-06-30    282.011923
2021-07-31    271.153362
2021-08-31    316.230473
Freq: M, Name: total, dtype: float64
```

In [21]:
```
fig, ax=plt.subplots(figsize=(10,8))
ax.plot(time_date)
plt.title('Mean total amount per month')
```

Out[21]: Text(0.5, 1.0, 'Mean total amount per month')



In [ ]:

In [ ]: