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

```
Create a report to answer your colleague's questions. Include:

What are the total sales for each payment method?
What is the average unit price for each product line?
Create plots to visualize findings for questions 1 and 2.
[Optional] Investigate further (e.g., average purchase value by client type,
total purchase value by product line, etc.)
Summarize your findings.
```

In [2]:
```
spareparts = pd.read_csv ('sales_data.csv')

spareparts.head()
```

Out[2]:

	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 [3]:
```
spareparts.columns
```

Out[3]:
```
Index(['date', 'warehouse', 'client_type', 'product_line', 'quantity',
       'unit_price', 'total', 'payment'],
      dtype='object')
```

In [5]:
```
spareparts.describe()
```

Out[5]:

	quantity	unit_price	total
count	1000.000000	1000.000000	1000.000000
mean	9.395000	30.322040	289.113000
std	9.659207	12.256488	345.227596
min	1.000000	10.030000	10.350000
25%	4.000000	21.085000	93.687500
50%	6.500000	28.570000	178.360000
75%	10.000000	37.917500	321.690000
max	40.000000	66.620000	2546.330000

In [6]:
```
# question 1. total sales for each payment method
paymentmethod = spareparts.groupby('payment', as_index = False).sum('total')
paymentmethod
# The 'groupby' synthax was used to find the total sales for each payment method.
```

Out[6]:

	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

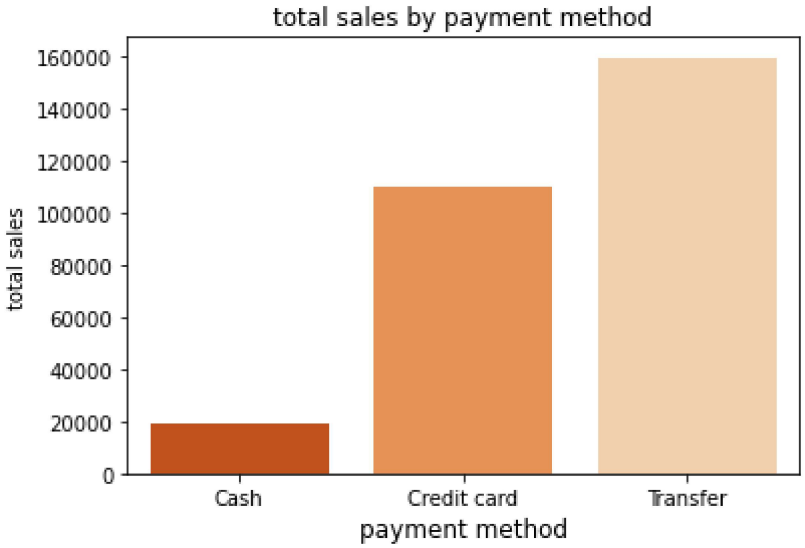
In [7]:
```
#question 2. What is the average unit price for each product line?
avgunitprice = spareparts.groupby('product_line', as_index = False).mean('unit_pr
avgunitprice
```

Out[7]:

	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

In [8]:
```
#3.Create plots to visualize findings for questions 1 and 2.
sns.barplot(x = 'payment', y = 'total',  data=paymentmethod, palette = 'Oranges_r
plt.xlabel('payment method', size = 12)
plt.ylabel('total sales')
plt.title('total sales by payment method')
plt.show
```

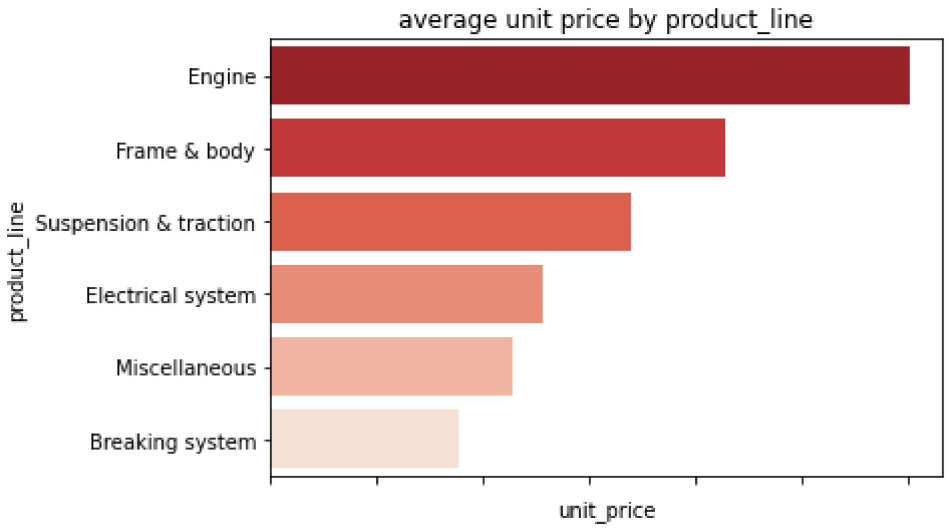
Out[8]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```



In [9]:
```
visual1 = sns.barplot(x = 'unit_price', y = 'product_line',palette = 'Reds_r', da
visual1.set (title = 'average unit price by product_line')
visual1.set_xticklabels(visual1.get_xticklabels())
```

```
C:\Users\USER\AppData\Local\Temp\ipykernel_13264\3785204768.py:3: UserWarning:
FixedFormatter should only be used together with FixedLocator
  visual1.set_xticklabels(visual1.get_xticklabels())
```

Out[9]:
```
[Text(0.0, 0, ''),
 Text(10.0, 0, ''),
 Text(20.0, 0, ''),
 Text(30.0, 0, ''),
 Text(40.0, 0, ''),
 Text(50.0, 0, ''),
 Text(60.0, 0, ''),
 Text(70.0, 0, '')]
```



In [ ]:
```
#[Optional] Investigate further (e.g., average purchase value by client type, tot
#Summarize your findings]
```

In [10]:
```
#total purchase value by product line
purchasevalue = spareparts.groupby('product_line', as_index = False).sum('total')
purchasevalue
```

Out[10]:

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

In [23]:
```
mostsold= spareparts.groupby('product_line', as_index = True).sum('quantity')
mostsold
```

Out[23]:

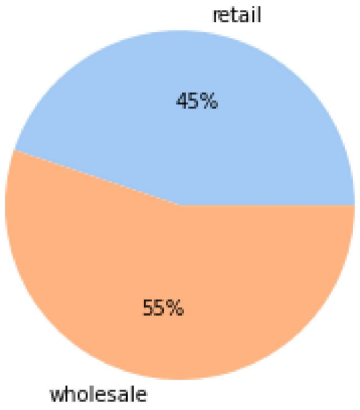
product_line	quantity	unit_price	total
Breaking system	2130	4080.32	38350.15
Electrical system	1698	4937.93	43612.71
Engine	627	3665.60	37945.38
Frame & body	1619	7110.15	69024.73
Miscellaneous	1176	2782.91	27165.82
Suspension & traction	2145	7745.13	73014.21

In [25]:
```
clienttype = spareparts.groupby('client_type', as_index = True).sum('quantity')
clienttype
#this is to find out the client type that made the most purchase
```

Out[25]:

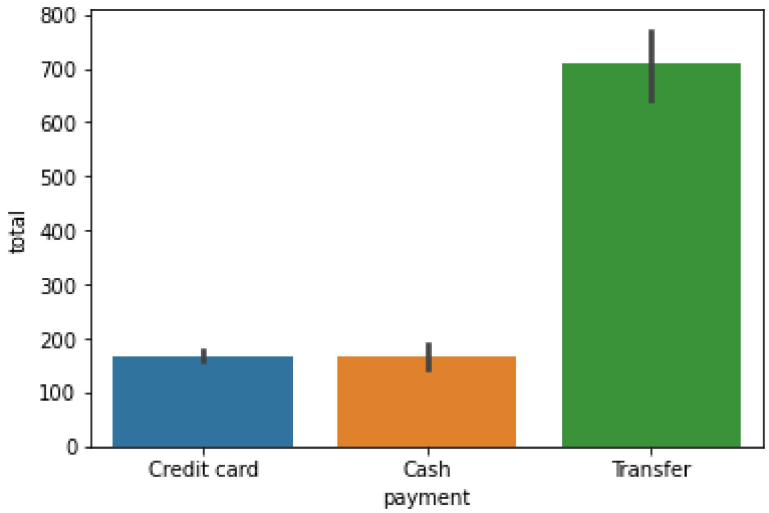
client_type	quantity	unit_price	total
Retail	4215	23472.31	129470.67
Wholesale	5180	6849.73	159642.33

In [35]:
```
data = [4215, 5180]
labels = ['retail', 'wholesale']
colors = sns.color_palette('pastel')[0:5]
plt.pie(data, labels= labels, colors = colors, autopct='%.0f%%')
plt.show()
```



In [44]:
```
#to find out the most used payment method
mostusedpayment = sns.barplot(x = 'payment', y = 'total', data = spareparts)
mostusedpayment
```

Out[44]: <AxesSubplot:xlabel='payment', ylabel='total'>



Summary

- More funds came in through the transfer mode of payment, it shows that tranfer was most used mode of payment.
- Breaking system has the lowest average unit price
- Suspension and taction was the most purchased with quantity of 2145 and total of 73014.21
- Even though Breaking system has a quantity of 2130 its total was 38350.15 while Frame and body had a quantity of 1619 total 0f 69024.73 this shows tha total is independent of quantity rather unit price.
- Wholesalers bouht more quantity than retailers