UNIQLO Sales Data Analysis

Background: UNIQLO, Japanese pronunciation: ユニクロ, is the core brand of Japan Fast Retailing Company, established in 1984, an internationally renowned clothing brand. The current chairman and general manager of Uniqlo, Liu Jing Zheng, introduced a hypermarket-style clothing sales method for the first time in Japan. The unique product planning, development and sales system are used to realize the low cost of the store operation, which have led to the big sale of Uniqlo. The meaning of Uniqlo refers to the storage of warehouses that neglect unnecessary decoration. The supermarket-style self-service shopping method provides customers with the cheap and good casual wear at a reasonable and reliable price. “UNIQLO” is Unique Clothing, means that provide customer the "low-cost, quality assurance" business philosophy.
According to the UNIQLO sales data, visualization by using Python and solve the follow questions:
1. How’s the sales status changing with the time?
2. What’ the sales of the different products, and which purchase way does customer prefer?
3.What’s the correlation between the revenue and the cost?

- Data diagnosis and cleaning, especially the null values
- Recognize the fields meaning in the table
- For the first question, plot the bar chats for data related to sales, including revenue, quant, customer number by time (weekdays and weekends)
- For the second question, plot the revenue bar chat by different product categories, and for the most popular purchasing channel, can check the revenue, product quant and customer number from different aspects such as customer’s gender, age and city
- For the third question, margin can be calculated from revenue-unit cost*quant, plot the bar chart for the margin of different products, then plot the heatmap and calculate the correlation factor between the revenue and the cost, plot the scatter plot to show their correlation more visually

Attached the coding screen shot as below:

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

In [2]:
```
U=pd.read_csv('UNIQLO.csv')
```

Question1:what's the correlation between sales and time

In [3]:
```
U.isnull().any()#check if there's null values
```

Out[3]:
```
store_id        False
city            False
channel         False
gender_group    False
age_group       False
wkd_ind         False
product         False
customer        False
revenue         False
order           False
quant           False
unit_cost       False
dtype: bool
```

In [4]:
```
U.describe()#check if got any abnormal data by min/max/mean
```

Out[4]:

	store_id	customer	revenue	order	quant	unit_cost
count	22293.000000	22293.000000	22293.000000	22293.000000	22293.000000	22293.000000
mean	335.391558	1.629480	159.531371	1.651998	1.858072	46.124658
std	230.236167	1.785605	276.254066	1.861480	2.347301	19.124347
min	19.000000	1.000000	-0.660000	1.000000	1.000000	9.000000
25%	142.000000	1.000000	64.000000	1.000000	1.000000	49.000000
50%	315.000000	1.000000	99.000000	1.000000	1.000000	49.000000
75%	480.000000	2.000000	175.000000	2.000000	2.000000	49.000000
max	831.000000	58.000000	12538.000000	65.000000	84.000000	99.000000

In [5]:
```
U[U['revenue']<0].head()#found min value in revenue is minus(abnormal), so need to find out all minus values in revenue
```

Out[5]:

	store_id	city	channel	gender_group	age_group	wkd_ind	product	customer	revenue	order	quant	unit_cost
20049	91	wuhan	online	Female	55-59	Weekday	sports	1	-0.66	1	2	49

In [6]:
```
U=U[U['revenue']>0]#remove the record which contains the minus value in revenue
```

In [7]:
```
U.describe()
```

Out[7]:

	store_id	customer	revenue	order	quant	unit_cost
count	22262.000000	22262.000000	22262.000000	22262.000000	22262.000000	22262.000000
mean	335.486614	1.630357	159.753549	1.652906	1.859222	46.127841
std	230.371454	1.786694	276.382135	1.862617	2.348723	19.120825
min	19.000000	1.000000	10.000000	1.000000	1.000000	9.000000
25%	142.000000	1.000000	66.000000	1.000000	1.000000	49.000000
50%	315.000000	1.000000	99.000000	1.000000	1.000000	49.000000
75%	480.000000	2.000000	175.000000	2.000000	2.000000	49.000000
max	831.000000	58.000000	12538.000000	65.000000	84.000000	99.000000

In [8]:
```
U['city'].value_counts()#check city field got 'unknow' or not
```

Out[8]:
```
shenzhen     4364
hangzhou     3785
wuhan        3566
shnaghai     2391
guangzhou    2170
chongqing    1787
xian         1593
chengdu      1529
beijing       577
nanjing       500
Name: city, dtype: int64
```

In [9]:
```
U['channel'].value_counts()#check channel field got 'unknow' or not
```

Out[9]:
```
offline    18373
online      3889
Name: channel, dtype: int64
```

In [10]:
```
U['gender_group'].value_counts()#including Unknow,can be removed when need to study on the customer info,in question1 no need to remove
```

Out[10]:
```
Female    14186
Male       7958
Unkown      118
Name: gender_group, dtype: int64
```

In [11]:
```
U['age_group'].value_counts()#including Unknow,can be removed when need to study on the customer info,in question1 no need to remove
```

Out[11]:
```
30-34     4423
25-29     4220
35-39     3689
20-24     3339
40-44     1950
>=60      1570
45-49     1093
50-54      669
<20        659
55-59      513
Unkown     137
Name: age_group, dtype: int64
```

In [12]:
```
U.groupby(['wkd_ind'])['revenue'].describe()
```

Out[12]:

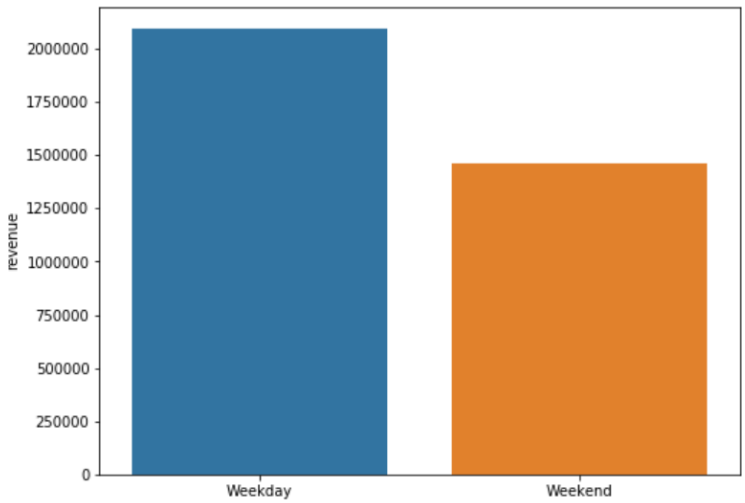
wkd_ind	count	mean	std	min	25%	50%	75%	max
Weekday	12450.0	168.188646	310.905834	10.0	66.0000	99.0	192.865	12538.0
Weekend	9812.0	149.050639	224.639689	10.0	60.3525	99.0	158.000	7919.0

In [13]:
```
U.groupby(['wkd_ind'])['revenue'].sum()#check the total revenue on weekday and weekend
```

Out[13]:
```
wkd_ind
Weekday    2093948.64
Weekend    1462484.87
Name: revenue, dtype: float64
```

In [14]:
```
fig=plt.figure(figsize=(8,6))
sns.barplot(x=np.unique(U['wkd_ind']),y=U.groupby(['wkd_ind'])['revenue'].sum(),data=U)#plot the bar chart
```

Out[14]: <matplotlib.axes._subplots.AxesSubplot at 0xc18c6d8>



In [15]:
```
U.groupby(['wkd_ind'])['order'].sum()#check the total order number during different period
```

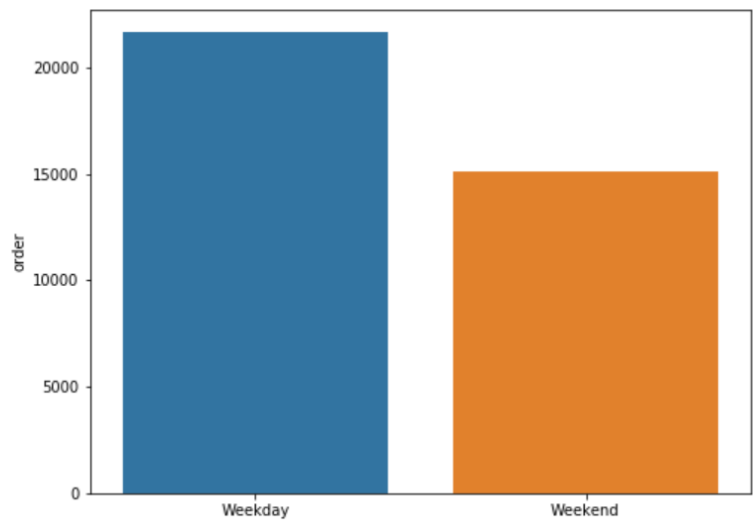
Out[15]:
```
wkd_ind
Weekday    21667
Weekend    15130
Name: order, dtype: int64
```

```
U.groupby(['wkd_ind'])['order'].describe()
```

wkd_ind	count	mean	std	min	25%	50%	75%	max
Weekday	12450.0	1.740321	2.077838	1.0	1.0	1.0	2.0	65.0
Weekend	9812.0	1.541989	1.539998	1.0	1.0	1.0	2.0	48.0

In [16]:
```
fig=plt.figure(figsize=(8,6))
sns.barplot(x=np.unique(U['wkd_ind']),y=U.groupby(['wkd_ind'])['order'].sum(),data=U)
```

Out[16]: <matplotlib.axes._subplots.AxesSubplot at 0xc0cab70>



```
In [17]: U.groupby(['wkd_ind'])['quant'].sum()#check the total quant during different period
```

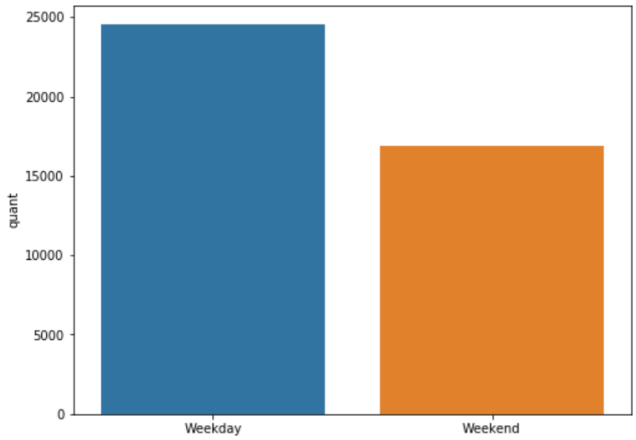
```
Out[17]: wkd_ind
         Weekday    24530
         Weekend    16860
         Name: quant, dtype: int64
```

```
U.groupby(['wkd_ind'])['quant'].describe()
```

wkd_ind	count	mean	std	min	25%	50%	75%	max
Weekday	12450.0	1.970281	2.670050	1.0	1.0	1.0	2.0	84.0
Weekend	9812.0	1.718304	1.853447	1.0	1.0	1.0	2.0	53.0

```
In [18]: fig=plt.figure(figsize=(8,6))
         sns.barplot(x=np.unique(U['wkd_ind']),y=U.groupby(['wkd_ind'])['quant'].sum(),data=U)
```

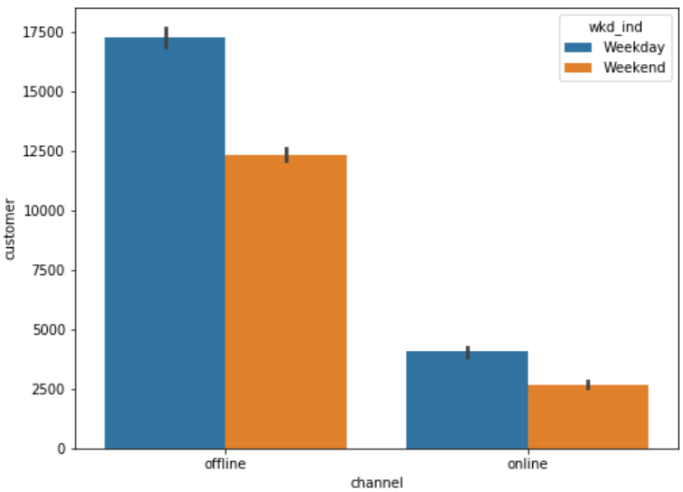
```
Out[18]: <matplotlib.axes._subplots.AxesSubplot at 0xc3dab38>
```



Conclusion：the sales in weekdays always be better than weekend no matter in total amount or average from revenue, quant and order number, the total amount with the large gap while the average in two periods got light difference.

```
In [57]: fig=plt.figure(figsize=(8,6))
         sns.barplot(x='channel',y='customer',hue='wkd_ind',data=U,estimator=sum)
```

```
Out[57]: <matplotlib.axes._subplots.AxesSubplot at 0xf0d40f0>
```



In addition：no matter from offline or online the weekdays' sales is better than weekends'.

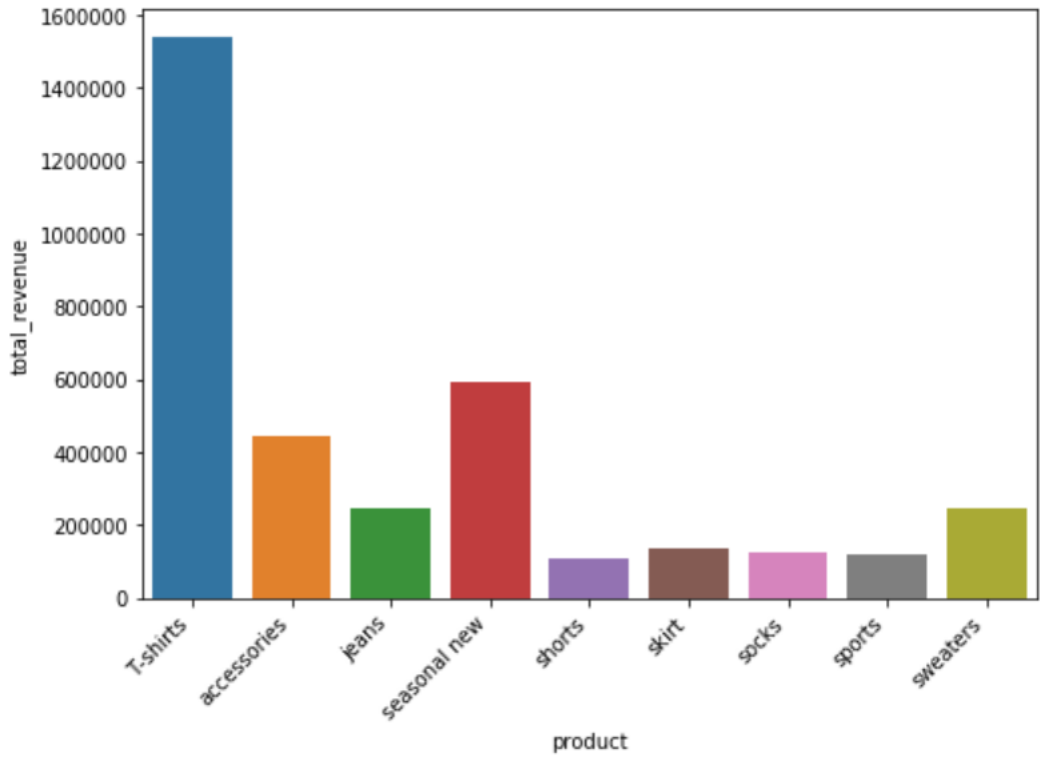
Question2:what's the sales of different kinds of product and how does customer prefer to order

In [19]:
```
U.groupby(['product'])['revenue'].sum()#check the total revenue of each product category
```

Out[19]:
```
product
T-shirts        1538744.84
accessories      444685.15
jeans            246127.48
seasonal new     590664.88
shorts           107485.88
skirt            137302.78
socks            127731.36
sports           118060.34
sweaters         245630.80
Name: revenue, dtype: float64
```

In [20]:
```
fig=plt.figure(figsize=(8,6))
product=np.unique(U['product'])
sns.barplot(x=product,y=U.groupby(['product'])['revenue'].sum(),data=U)
fig.autofmt_xdate(rotation = 45)
plt.xlabel('product')
plt.ylabel('total_revenue')
```

Out[20]:
```
Text(0,0.5,'total_revenue')
```

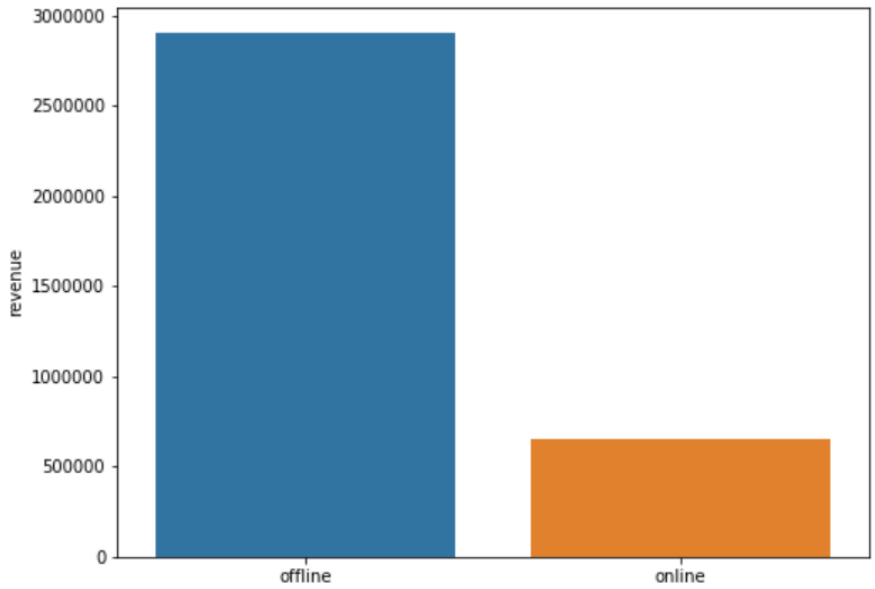


In [21]:
```
U.groupby(['channel'])['revenue'].sum()#study the sales of the different purchasing methods,which channel does customer prefer
```

Out[21]:
```
channel
offline    2903262.37
online      653171.14
Name: revenue, dtype: float64
```

In [22]:
```
fig=plt.figure(figsize=(8,6))
sns.barplot(x=np.unique(U['channel']),y=U.groupby(['channel'])['revenue'].sum(),data=U)
```

Out[22]: <matplotlib.axes._subplots.AxesSubplot at 0xc9b5e80>

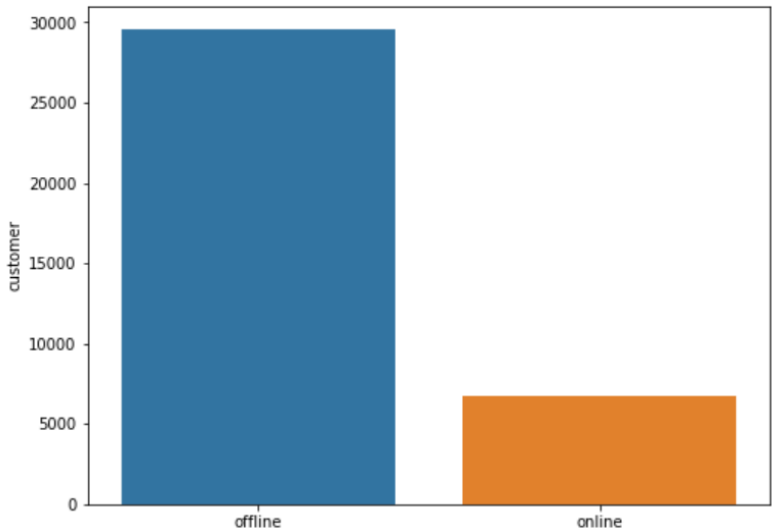


In [23]:
```
U.groupby(['channel'])['customer'].sum()#check the total customer number purchasing with different channels
```

Out[23]:
```
channel
offline    29551
online      6744
Name: customer, dtype: int64
```

In [24]:
```
fig=plt.figure(figsize=(8,6))
sns.barplot(x=np.unique(U['channel']),y=U.groupby(['channel'])['customer'].sum(),data=U)
```

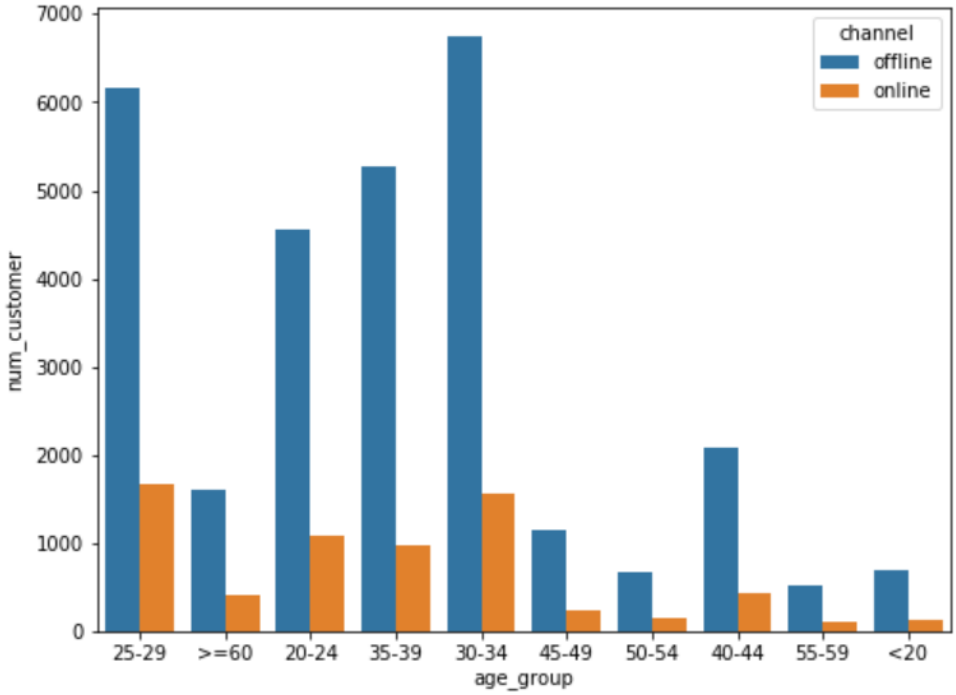
Out[24]: <matplotlib.axes._subplots.AxesSubplot at 0xcb864a8>



In [25]:
```
U_1=pd.read_csv('by age.csv')#using SQL to remove the 'Unknown' and group sum of customer_num by age and channel save as U_1
```

In [26]:
```
fig=plt.figure(figsize=(8,6))
sns.barplot(x='age_group',y='num_customer',hue='channel',data=U_1)
```

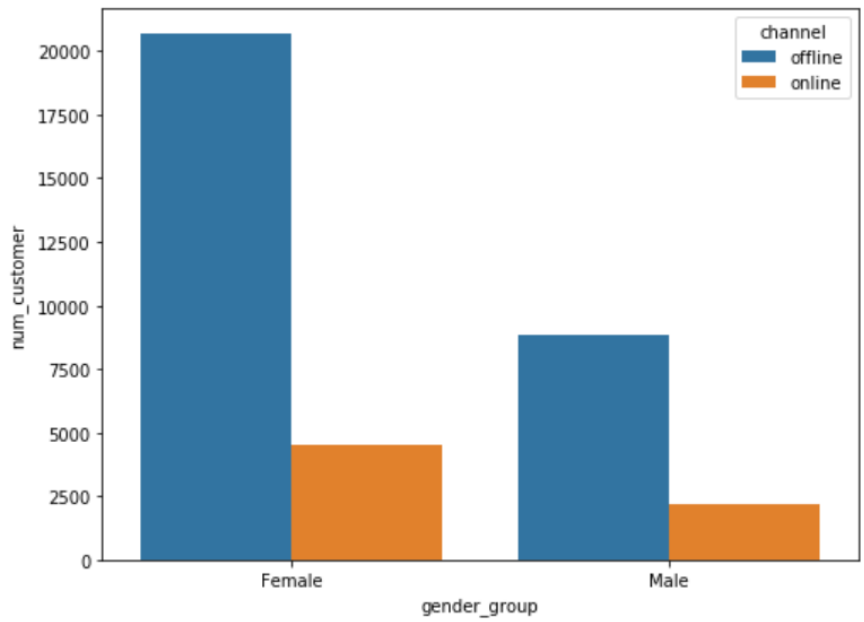
Out[26]: <matplotlib.axes._subplots.AxesSubplot at 0xca1f198>



In [27]:
```
U_2=pd.read_csv('by gender.csv')#using SQL to remove the 'Unknown' and group sum of customer_num by gender and channel save as U_2
```

In [28]:
```
fig=plt.figure(figsize=(8,6))
sns.barplot(x='gender_group',y='num_customer',hue='channel',data=U_2)
```

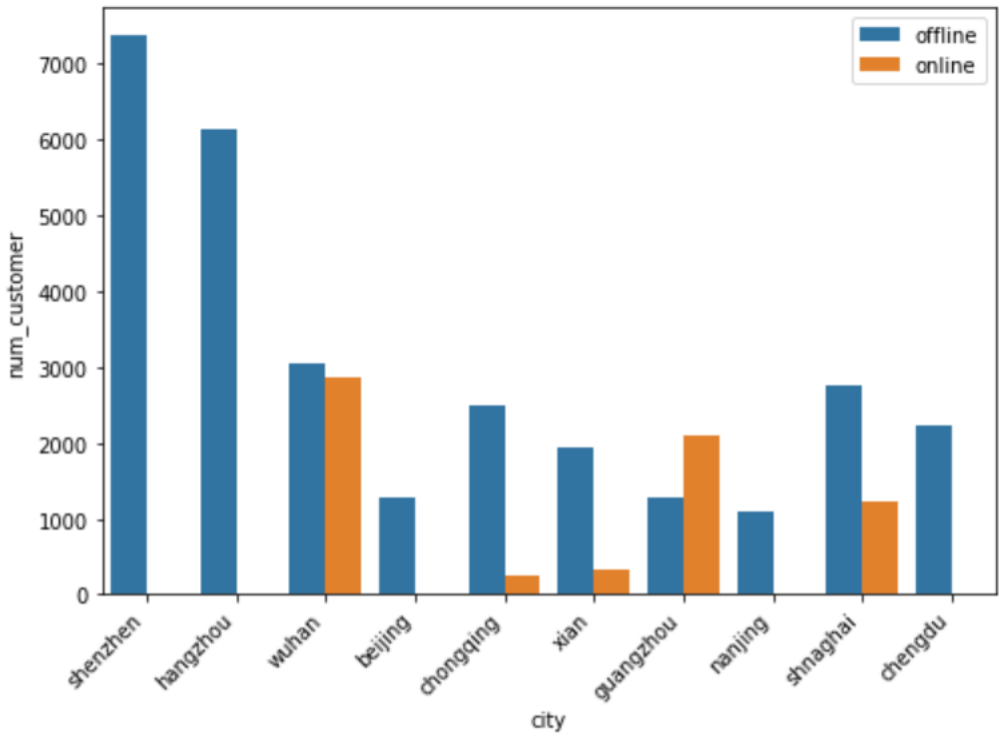
Out[28]: <matplotlib.axes._subplots.AxesSubplot at 0xcdb8b38>



In [29]:
```
U_3=pd.read_csv('by city.csv')#using SQL to group sum of customer_num by city and channel save as U_3
```

In [30]:
```
fig=plt.figure(figsize=(8,6))
sns.barplot(x='city',y='num_customer',hue='channel',data=U_3)
fig.autofmt_xdate(rotation = 45)
plt.legend(loc='upper right')
```

Out[30]: <matplotlib.legend.Legend at 0xdbfc630>



Conclusion: the most popular product is T-shirt, its revenue and customer number both are much higher than the other ones; most people in all age range no matter female or male prefer to purchase offline, but people from Guangzhou seems to like purchasing online, as well as Wuhan.

Question3:what's the correlation between revenue and cost

In [31]:
```
U['margin']=U['revenue']-(U['unit_cost']*U['quant'])#add in a new field 'margin'
```

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

Out[32]:

	store_id	city	channel	gender_group	age_group	wkd_ind	product	customer	revenue	order	quant	unit_cost	margin
0	658	shenzhen	offline	Female	25-29	Weekday	seasonal new	4	796.0	4	4	59	560.0
1	146	hangzhou	offline	Female	25-29	Weekday	sports	1	149.0	1	1	49	100.0
2	70	shenzhen	offline	Male	>=60	Weekday	T-shirts	2	178.0	2	2	49	80.0
3	658	shenzhen	offline	Female	25-29	Weekday	T-shirts	1	59.0	1	1	49	10.0
4	229	shenzhen	offline	Male	20-24	Weekend	socks	2	65.0	2	3	9	38.0

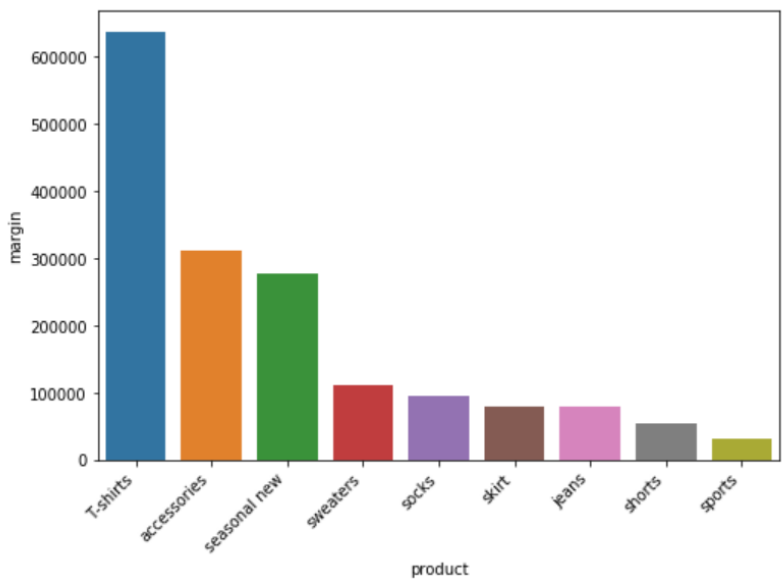
In [33]:
```
U.margin.describe()#check margin, found there's minus value
```

Out[33]:
```
count    22262.000000
mean        75.061698
std        179.888672
min       -650.000000
25%         18.000000
50%         41.000000
75%         88.000000
max       8408.000000
Name: margin, dtype: float64
```

In [34]:
```
U.groupby(['product'])['margin'].sum()#check the margin of each product
```

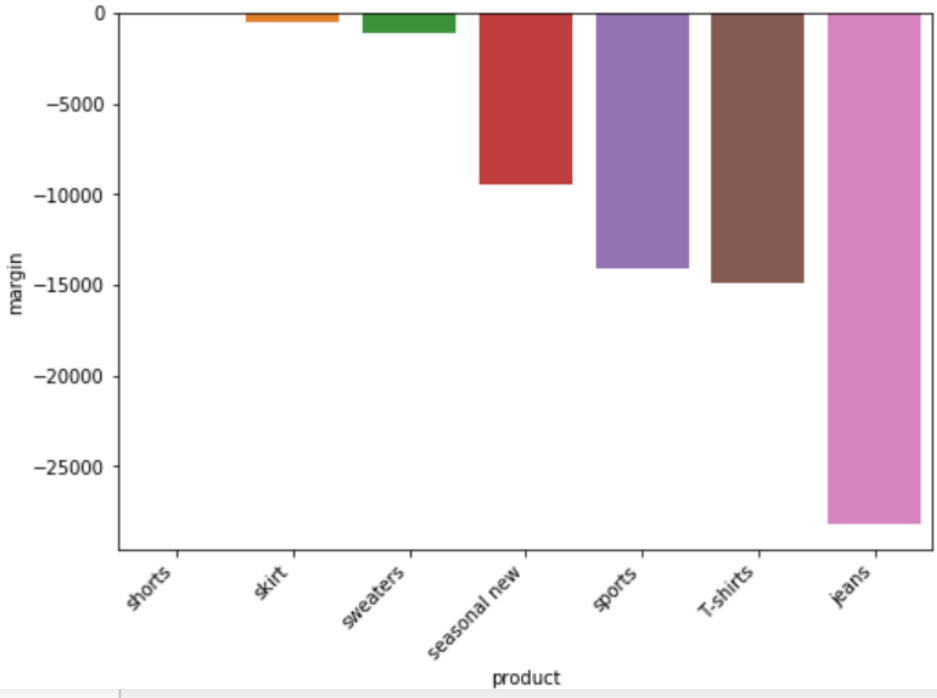
Out[34]:
```
product
T-shirts        636507.84
accessories     310676.15
jeans            78457.48
seasonal new    276076.88
shorts           53943.88
skirt            78597.78
socks            95025.36
sports           30252.34
sweaters        111485.80
Name: margin, dtype: float64
```

In [35]:
```
fig=plt.figure(figsize=(8,6))
Ua=U.groupby(['product'])['margin'].sum().reset_index()
Ub=Ua.sort_values(by='margin',ascending=False)
sns.barplot(x='product',y='margin',data=Ub)
fig.autofmt_xdate(rotation = 45)
```



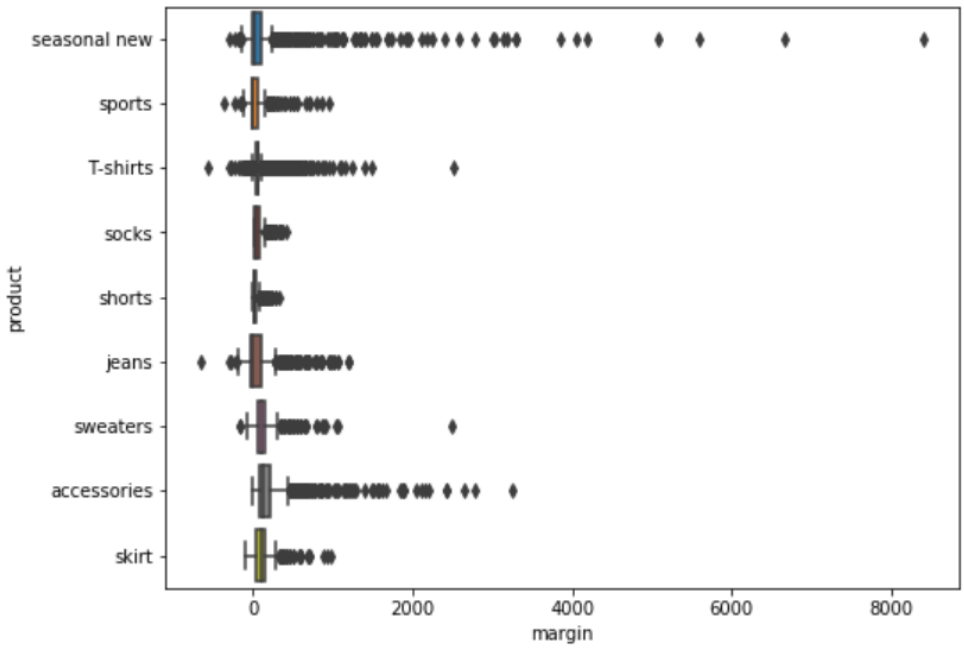
In [36]:
```
U0=U[U['margin']<0]#find that jeans got the highest deficit
```

In [37]:
```
fig=plt.figure(figsize=(8,6))
Ux=U0.groupby(['product'])['margin'].sum().reset_index()
Uy=Ux.sort_values(by='margin',ascending=False)
sns.barplot(x='product',y='margin',data=Uy)
fig.autofmt_xdate(rotation = 45)
```



In [38]:
```
fig=plt.figure(figsize=(8,6))#chexk the margin distribution of each product by boxplot
sns.boxplot(x='margin',y='product',data=U)
```

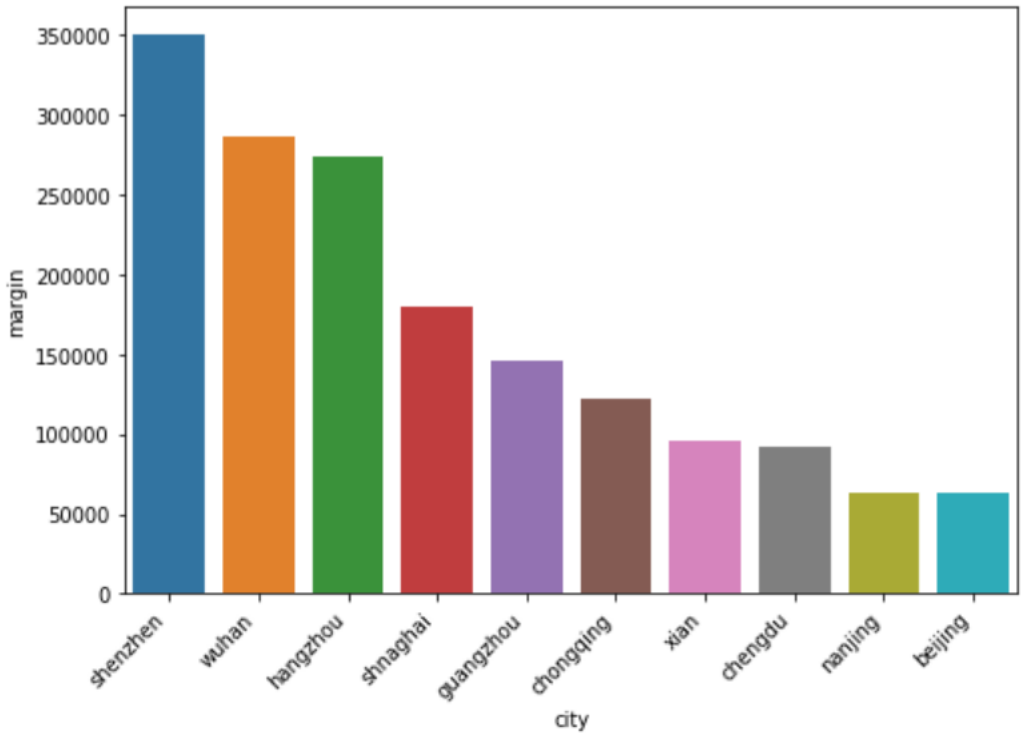
Out[38]: <matplotlib.axes._subplots.AxesSubplot at 0xe079048>



In [39]:
```
U.groupby(['city'])['margin'].sum()#check the margin of each city
```

Out[39]:
```
city
beijing       63455.62
chengdu       91763.86
chongqing    121786.65
guangzhou    146271.49
hangzhou     273753.49
nanjing       63675.93
shenzhen     349875.68
shnaghai     179946.73
wuhan        285279.44
xian         95214.62
Name: margin, dtype: float64
```

In [40]:
```
fig=plt.figure(figsize=(8,6))
U_sort=U.groupby(['city'])['margin'].sum().reset_index()
U_sort1=U_sort.sort_values(by='margin',ascending=False)
sns.barplot(x='city',y='margin',data=U_sort1)
fig.autofmt_xdate(rotation = 45)
```



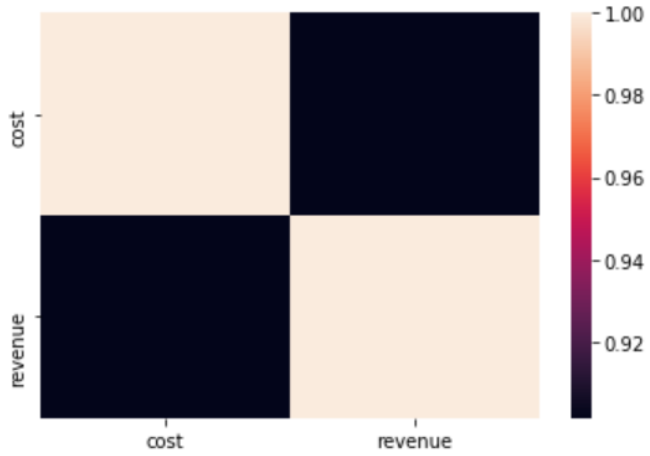
In [62]:
```
U['cost']=U['unit_cost']*U['quant']
q1=['cost','revenue']
U[q1].corr()
```

Out[62]:

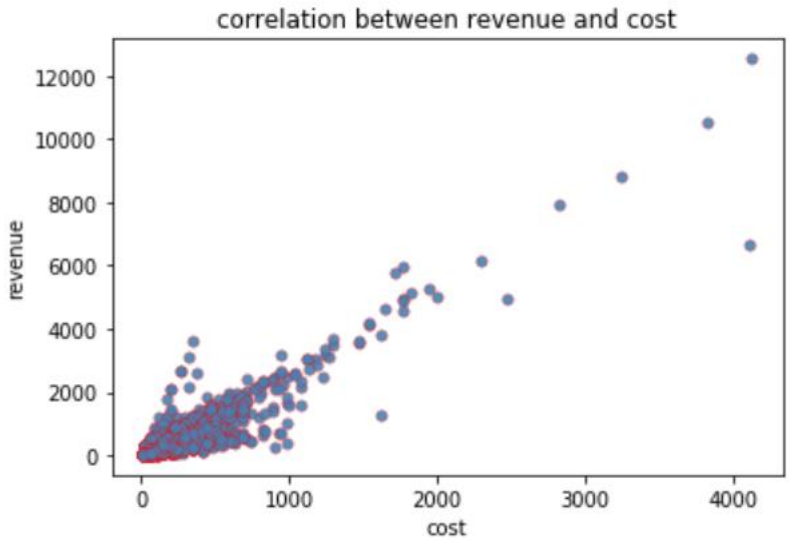
	cost	revenue
cost	1.00000	0.90142
revenue	0.90142	1.00000

In [65]:
```
sns.heatmap(U[q1].corr())
```

Out[65]: <matplotlib.axes._subplots.AxesSubplot at 0x107c9fd0>



In [66]:
```
plt.scatter(U.cost,
            U.revenue,
            s = 30,
            c = 'steelblue',
            marker = 'o',
            alpha = 0.9,
            linewidths = 0.3,
            edgecolors = 'red'
            )
plt.title('correlation between revenue and cost')
plt.xlabel('cost')
plt.ylabel('revenue')
plt.tick_params(top = 'off', right = 'off')
plt.show()
```



Conclusion: T-shirt got the highest margin, jeans lost much; from the heatmap and the scatter plot can see the revenue and cost shows the positive correlation, means the higher cost makes the higher revenue, it kindly matchs with high-end product model.