Context

Welcome to the shopping world of Istanbul! Our dataset contains shopping information from 10 different shopping malls between 2021 and 2023. We have gathered data from various age groups and genders to provide a comprehensive view of shopping habits in Istanbul. The dataset includes essential information such as invoice numbers, customer IDs, age, gender, payment methods, product categories, quantity, price, order dates, and shopping mall locations. We hope that this dataset will serve as a valuable resource for researchers, data analysts, and machine learning enthusiasts who want to gain insights into shopping trends and patterns in Istanbul. Explore the dataset and discover the fascinating world of Istanbul shopping!

Importing Libraries

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

Data Processing

In [2]:
```
shops = pd.read_csv('customer_shopping_data.csv')
shops
```

Out[2]:

	invoice_no	customer_id	gender	age	category	quantity	price	payment_method	invoice_date	sho
0	I138884	C241288	Female	28	Clothing	5	1500.40	Credit Card	5/8/2022	
1	I317333	C111565	Male	21	Shoes	3	1800.51	Debit Card	12/12/2021	Fo
2	I127801	C266599	Male	20	Clothing	1	300.08	Cash	9/11/2021	
3	I173702	C988172	Female	66	Shoes	5	3000.85	Credit Card	16/05/2021	M
4	I337046	C189076	Female	53	Books	4	60.60	Cash	24/10/2021	
...	...	...	...	...	...	...	...	...	...	
99452	I219422	C441542	Female	45	Souvenir	5	58.65	Credit Card	21/09/2022	
99453	I325143	C569580	Male	27	Food & Beverage	2	10.46	Cash	22/09/2021	Fo
99454	I824010	C103292	Male	63	Food & Beverage	2	10.46	Debit Card	28/03/2021	
99455	I702964	C800631	Male	56	Technology	4	4200.00	Cash	16/03/2021	
99456	I232867	C273973	Female	36	Souvenir	3	35.19	Credit Card	15/10/2022	Ma

99457 rows × 10 columns

Display Top 10 Rows of the shops dataset

In [3]:
```
shops.head(10)
```

Out[3]:

	invoice_no	customer_id	gender	age	category	quantity	price	payment_method	invoice_date	shoppin
0	I138884	C241288	Female	28	Clothing	5	1500.40	Credit Card	5/8/2022	K
1	I317333	C111565	Male	21	Shoes	3	1800.51	Debit Card	12/12/2021	Forum Is
2	I127801	C266599	Male	20	Clothing	1	300.08	Cash	9/11/2021	Me
3	I173702	C988172	Female	66	Shoes	5	3000.85	Credit Card	16/05/2021	Metrop
4	I337046	C189076	Female	53	Books	4	60.60	Cash	24/10/2021	K
5	I227836	C657758	Female	28	Clothing	5	1500.40	Credit Card	24/05/2022	Forum Is
6	I121056	C151197	Female	49	Cosmetics	1	40.66	Cash	13/03/2022	Istiny
7	I293112	C176086	Female	32	Clothing	2	600.16	Credit Card	13/01/2021	Mall of Is
8	I293455	C159642	Male	69	Clothing	3	900.24	Credit Card	4/11/2021	Me
9	I326945	C283361	Female	60	Clothing	2	600.16	Credit Card	22/08/2021	K

Display Last 10 Rows of the shops dataset

In [4]:
```
shops.tail(10)
```

Out[4]:

	invoice_no	customer_id	gender	age	category	quantity	price	payment_method	invoice_date	sho
99447	I281214	C288090	Female	37	Toys	3	107.52	Cash	21/02/2021	M
99448	I332105	C231387	Female	65	Shoes	4	2400.68	Credit Card	29/08/2021	M
99449	I134399	C953724	Male	65	Clothing	1	300.08	Cash	1/1/2023	
99450	I170504	C226974	Female	28	Books	1	15.15	Cash	28/02/2023	
99451	I675411	C513603	Male	50	Toys	5	179.20	Cash	9/10/2021	M
99452	I219422	C441542	Female	45	Souvenir	5	58.65	Credit Card	21/09/2022	
99453	I325143	C569580	Male	27	Food & Beverage	2	10.46	Cash	22/09/2021	Fo
99454	I824010	C103292	Male	63	Food & Beverage	2	10.46	Debit Card	28/03/2021	
99455	I702964	C800631	Male	56	Technology	4	4200.00	Cash	16/03/2021	
99456	I232867	C273973	Female	36	Souvenir	3	35.19	Credit Card	15/10/2022	Ma

Check Datatype of Each Column

In [5]:
```
shops.dtypes
```

Out[5]:
```
invoice_no         object
customer_id        object
gender             object
age                 int64
category           object
quantity            int64
price             float64
payment_method     object
invoice_date       object
shopping_mall      object
dtype: object
```

Check NULL Values in shops

In [6]:
```
shops.isnull().sum()
```

Out[6]:
```
invoice_no        0
customer_id       0
gender            0
age               0
category          0
quantity          0
price             0
payment_method    0
invoice_date      0
shopping_mall     0
dtype: int64
```

How many Rows and Columns are there in our dataset

In [7]:
```
shops.shape
```

Out[7]:
```
(99457, 10)
```

In [8]:
```
shops.columns
```

Out[8]:
```
Index(['invoice_no', 'customer_id', 'gender', 'age', 'category', 'quantity',
       'price', 'payment_method', 'invoice_date', 'shopping_mall'],
      dtype='object')
```

In [9]:
```
shops.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 99457 entries, 0 to 99456
Data columns (total 10 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   invoice_no      99457 non-null  object 
 1   customer_id     99457 non-null  object 
 2   gender          99457 non-null  object 
 3   age             99457 non-null  int64  
 4   category        99457 non-null  object 
 5   quantity        99457 non-null  int64  
 6   price           99457 non-null  float64
 7   payment_method  99457 non-null  object 
 8   invoice_date    99457 non-null  object 
 9   shopping_mall   99457 non-null  object 
dtypes: float64(1), int64(2), object(7)
memory usage: 7.6+ MB
```

Let's view quick summary of each column in our dataset

In [10]:
```
shops.describe().T
```

Out[10]:

	count	mean	std	min	25%	50%	75%	max
age	99457.0	43.427089	14.990054	18.00	30.00	43.0	56.00	69.0
quantity	99457.0	3.003429	1.413025	1.00	2.00	3.0	4.00	5.0
price	99457.0	689.256321	941.184567	5.23	45.45	203.3	1200.32	5250.0

Let's check how much shopping mall in the dataset

In [11]:
```
shops['shopping_mall'].unique()
```

Out[11]:
```
array(['Kanyon', 'Forum Istanbul', 'Metrocity', 'Metropol AVM',
       'Istinye Park', 'Mall of Istanbul', 'Emaar Square Mall',
       'Cevahir AVM', 'Viaport Outlet', 'Zorlu Center'], dtype=object)
```

Let's check how much category in the dataset

In [12]:
```
shops['category'].unique()
```

Out[12]:
```
array(['Clothing', 'Shoes', 'Books', 'Cosmetics', 'Food & Beverage',
       'Toys', 'Technology', 'Souvenir'], dtype=object)
```

Let's check how much invoice_no, customer_id in the dataset

In [13]:
```
shops['invoice_no'].nunique()
```

Out[13]:
```
99457
```

In [14]:
```
shops['customer_id'].nunique()
```

Out[14]:
```
99457
```

I see that each customer only buys once at the store

Remove attributes that have no effect on the model

In [15]:
```
shops.drop(['invoice_no','customer_id'], axis=1, inplace=True)
```

In [16]:
```
shops.head()
```

Out[16]:

	gender	age	category	quantity	price	payment_method	invoice_date	shopping_mall
0	Female	28	Clothing	5	1500.40	Credit Card	5/8/2022	Kanyon
1	Male	21	Shoes	3	1800.51	Debit Card	12/12/2021	Forum Istanbul
2	Male	20	Clothing	1	300.08	Cash	9/11/2021	Metrocity
3	Female	66	Shoes	5	3000.85	Credit Card	16/05/2021	Metropol AVM
4	Female	53	Books	4	60.60	Cash	24/10/2021	Kanyon

nunique values

In [17]:
```
shops.nunique()
```

Out[17]:
```
gender              2
age                52
category            8
quantity            5
price              40
payment_method      3
invoice_date      797
shopping_mall      10
dtype: int64
```

In [18]:
```
shops.isna().sum()
```

Out[18]:
```
gender            0
age               0
category          0
quantity          0
price             0
payment_method    0
invoice_date      0
shopping_mall     0
```

EDA

Let's look at the statistics for each shopping mall

In [19]:
```
shops.groupby('shopping_mall')['age','price'].describe().T
```

```
C:\Users\nguye\AppData\Local\Temp\ipykernel_7032\2568613092.py:1: FutureWarning: Indexin
g with multiple keys (implicitly converted to a tuple of keys) will be deprecated, use a
list instead.
  shops.groupby('shopping_mall')['age','price'].describe().T
```

Out[19]:

	shopping_mall	Cevahir AVM	Emaar Square Mall	Forum Istanbul	Istinye Park	Kanyon	Mall of Istanbul	Met
age	count	4991.000000	4811.000000	4947.000000	9781.000000	19823.000000	19943.000000	15011.0
	mean	43.172511	43.561630	43.537497	43.383601	43.498966	43.440455	43.4
	std	14.911633	14.889450	15.179075	15.051176	14.952212	15.038037	14.9
	min	18.000000	18.000000	18.000000	18.000000	18.000000	18.000000	18.0
	25%	30.000000	31.000000	30.000000	30.000000	31.000000	30.000000	30.0
	50%	43.000000	43.000000	44.000000	43.000000	43.000000	43.000000	44.0
	75%	56.000000	57.000000	57.000000	56.000000	56.000000	56.500000	56.0
	max	69.000000	69.000000	69.000000	69.000000	69.000000	69.000000	69.0
price	count	4991.000000	4811.000000	4947.000000	9781.000000	19823.000000	19943.000000	15011.0
	mean	687.972719	704.720081	674.363012	686.747525	691.658944	694.566395	682.8
	std	952.002169	957.855695	916.138978	932.392502	951.995108	946.272210	930.2
	min	5.230000	5.230000	5.230000	5.230000	5.230000	5.230000	5.2
	25%	40.660000	45.450000	40.660000	40.660000	40.660000	45.450000	43.0
	50%	203.300000	300.080000	203.300000	203.300000	203.300000	203.300000	203.3
	75%	1200.320000	1200.320000	1200.320000	1200.320000	1200.320000	1200.320000	1200.3
	max	5250.000000	5250.000000	5250.000000	5250.000000	5250.000000	5250.000000	5250.0

- From the data, we can see that the most shopping is done at the Canyon shopping center, and the least shopping is done at the Mall of Istanbul shopping center. The average age of customers shopping at shopping centers is generally the same (43), and if we take the average as the basis, the most shopping is done at the Emaar Square Mall.

- (Từ dữ liệu trên ta có thể thấy: mua sắm nhiều nhất ở trung tâm mua sắm canyon , ít mua sắm nhất ở Trung tâm mua sắm Mall of Istanbul độ tuổi trung bình của khách hàng mua sắm tại các trung tâm mua sắm nói chung là như nhau (43), nếu chúng ta lấy mức trung bình làm cơ sở, thì mua sắm nhiều nhất được thực hiện tại Emaar Square Mall.)

In [20]:
```
shops.groupby('gender')['age','price'].describe().T
```

```
C:\Users\nguye\AppData\Local\Temp\ipykernel_7032\1759342618.py:1: FutureWarning: Indexin
g with multiple keys (implicitly converted to a tuple of keys) will be deprecated, use a
list instead.
  shops.groupby('gender')['age','price'].describe().T
```

Out[20]:

	gender	Female	Male
age	count	59482.000000	39975.000000
	mean	43.453515	43.387767
	std	14.965468	15.026664
	min	18.000000	18.000000
	25%	30.000000	30.000000
	50%	43.000000	43.000000
	75%	56.000000	56.000000
	max	69.000000	69.000000
price	count	59482.000000	39975.000000
	mean	688.137615	690.920933
	std	940.791874	941.777893
	min	5.230000	5.230000
	25%	45.450000	40.660000
	50%	203.300000	203.300000
	75%	1200.320000	1200.320000
	max	5250.000000	5250.000000

- Based on the gender of shoppers, the age statistics of male and female shoppers are almost the same. Women shop more than men, and if we look at the average spending, men seem to spend more than women.

- (Dựa trên giới tính của người mua sắm, thống kê độ tuổi của người mua sắm nam và nữ gần như giống nhau, phụ nữ mua sắm nhiều hơn và nếu chúng ta nhìn vào mức chi tiêu trung bình thì ta thấy nam giới có vẻ chi tiêu nhiều hơn so với phụ nữ.)

In [21]:
```
shops.groupby('payment_method')['age','price'].describe().T
```

```
C:\Users\nguye\AppData\Local\Temp\ipykernel_7032\1118872701.py:1: FutureWarning: Indexin
g with multiple keys (implicitly converted to a tuple of keys) will be deprecated, use a
list instead.
  shops.groupby('payment_method')['age','price'].describe().T
```

Out[21]:

	payment_method	Cash	Credit Card	Debit Card
age	count	44447.000000	34931.000000	20079.000000
	mean	43.457421	43.427901	43.358534
	std	15.011227	14.997443	14.930696
	min	18.000000	18.000000	18.000000
	25%	30.000000	30.000000	30.000000
	50%	43.000000	43.000000	43.000000
	75%	56.000000	56.000000	56.000000
	max	69.000000	69.000000	69.000000
price	count	44447.000000	34931.000000	20079.000000
	mean	690.823475	688.542467	687.029135
	std	942.599875	943.067511	934.789284
	min	5.230000	5.230000	5.230000
	25%	40.660000	40.660000	45.450000
	50%	203.300000	203.300000	203.300000
	75%	1200.320000	1200.320000	1200.320000
	max	5250.000000	5250.000000	5250.000000

- 44447 customers completed their purchases with cash, 34931 customers with customer credit cards and 20079 customers with debit cards, if we take the average expenditures as a basis, cash expenditures were relatively higher.

- (44447 khách hàng đã hoàn tất giao dịch mua bằng tiền mặt, 34931 khách hàng có thẻ tín dụng khách hàng và 20079 khách hàng có thẻ ghi nợ, nếu chúng ta lấy chi tiêu trung bình làm cơ sở thì chi tiêu bằng tiền mặt tương đối cao hơn.)

In [22]:
```
shops.groupby('category')['age','price'].describe().T
```

```
C:\Users\nguye\AppData\Local\Temp\ipykernel_7032\1409565860.py:1: FutureWarning: Indexin
g with multiple keys (implicitly converted to a tuple of keys) will be deprecated, use a
list instead.
  shops.groupby('category')['age','price'].describe().T
```

Out[22]:

	category	Books	Clothing	Cosmetics	Food & Beverage	Shoes	Souvenir	Technology
age	count	4981.000000	34487.000000	15097.000000	14776.000000	10034.000000	4999.000000	4996.000000
	mean	43.541859	43.409227	43.580645	43.354426	43.454953	43.393079	43.368495
	std	15.051601	14.998012	14.936556	14.981172	15.003153	15.009072	14.835330
	min	18.000000	18.000000	18.000000	18.000000	18.000000	18.000000	18.000000
	25%	30.000000	30.000000	31.000000	31.000000	30.000000	30.000000	31.000000
	50%	44.000000	43.000000	44.000000	43.000000	44.000000	43.000000	43.000000
	75%	56.000000	56.000000	56.000000	56.000000	57.000000	57.000000	56.000000
	max	69.000000	69.000000	69.000000	69.000000	69.000000	69.000000	69.000000
price	count	4981.000000	34487.000000	15097.000000	14776.000000	10034.000000	4999.000000	4996.000000
	mean	45.568621	901.084021	122.448626	15.671948	1807.388568	34.894345	3156.935548
	std	21.492662	424.225594	57.458097	7.422176	845.116269	16.564498	1481.559805
	min	15.150000	300.080000	40.660000	5.230000	600.170000	11.730000	1050.000000
	25%	30.300000	600.160000	81.320000	10.460000	1200.340000	23.460000	2100.000000
	50%	45.450000	900.240000	121.980000	15.690000	1800.510000	35.190000	3150.000000
	75%	60.600000	1200.320000	162.640000	20.920000	2400.680000	46.920000	4200.000000
	max	75.750000	1500.400000	203.300000	26.150000	3000.850000	58.650000	5250.000000

- Most customers spend a lot of money on the fashion sector, especially on clothing. The cost of clothing is quite high compared to other items, and the least amount of money is spent on shopping for books.
- (hầu hết các khách hàng đều chi nhiều tiền ở mảng thời trang đặc biệt là quần áo chi phí cho quần áo khá cao so với các mặt hàng khác, và chi phí mua sắm được dành ít nhất là sách.)

Time Series Analysis

In [39]:
```
shops['invoice_date'] = pd.to_datetime(shops['invoice_date'])
shops.dtypes
```

Out[39]:
```
gender                    object
age                        int64
category                  object
quantity                   int64
price                    float64
payment_method            object
invoice_date      datetime64[ns]
shopping_mall             object
year                       int64
month                      int64
day                        int64
dtype: object
```

In [41]:
```
shops['year'] = shops['invoice_date'].dt.year
shops['month'] = shops['invoice_date'].dt.month
shops['day'] = shops['invoice_date'].dt.day
```

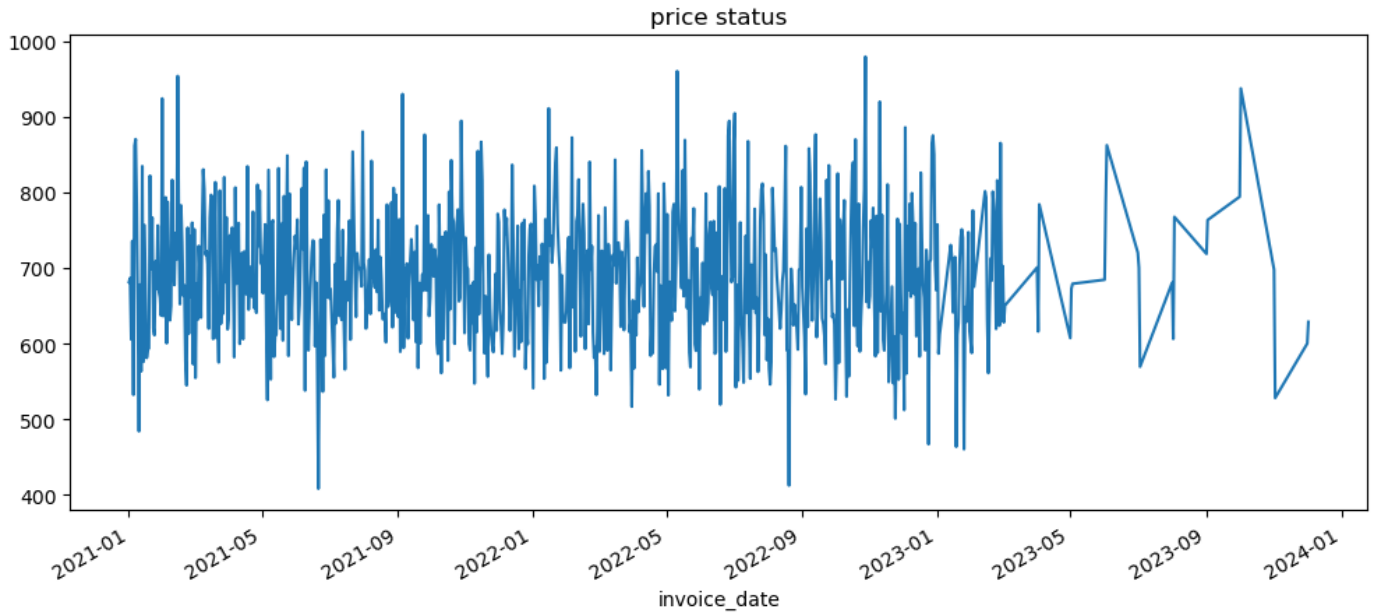
In [40]:
```
shops.head()
```

Out[40]:

	gender	age	category	quantity	price	payment_method	invoice_date	shopping_mall	year	month	day
0	Female	28	Clothing	5	1500.40	Credit Card	2022-05-08	Kanyon	2022	5	8
1	Male	21	Shoes	3	1800.51	Debit Card	2021-12-12	Forum Istanbul	2021	12	12
2	Male	20	Clothing	1	300.08	Cash	2021-09-11	Metrocity	2021	9	11
3	Female	66	Shoes	5	3000.85	Credit Card	2021-05-16	Metropol AVM	2021	5	16
4	Female	53	Books	4	60.60	Cash	2021-10-24	Kanyon	2021	10	24

Visualize

In [26]:
```
shopsgroup=shops.groupby('invoice_date').mean()
plt.figure(figsize=(12,5))
shopsgroup['price'].plot(x=shops.invoice_date)
plt.title("price status")
plt.show()
```

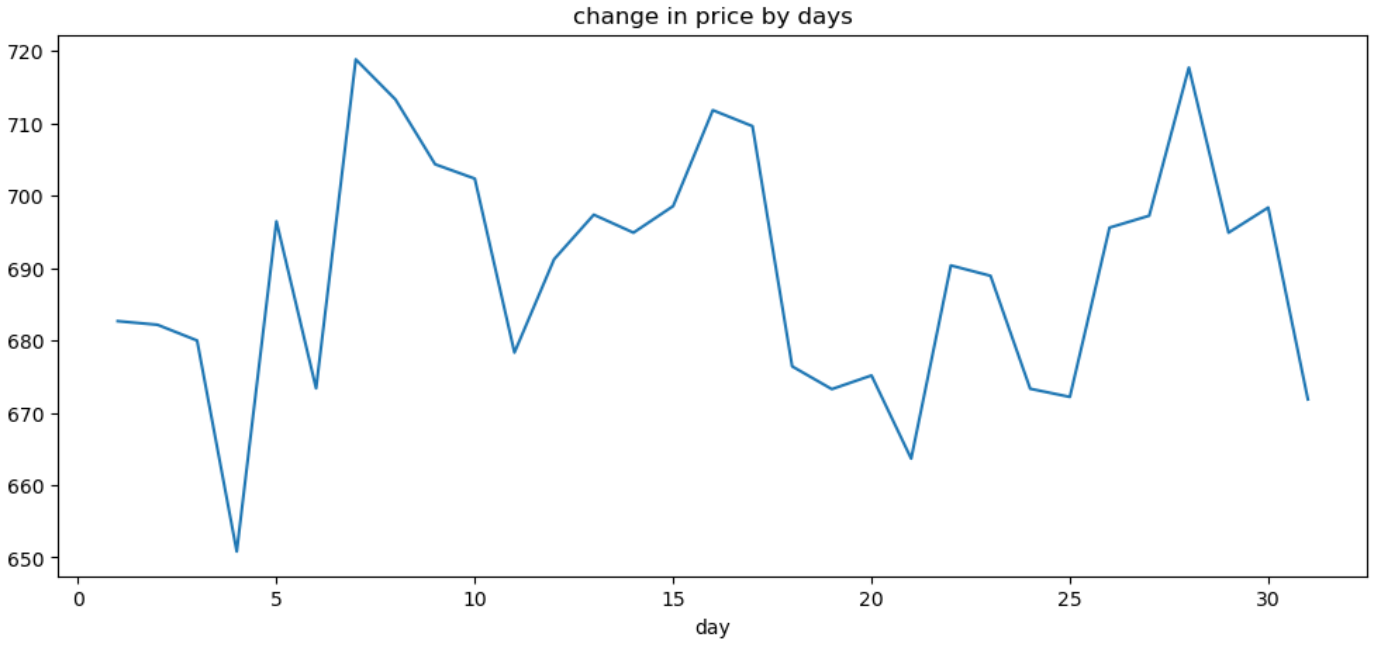


- We visualized the change of prices as a time series.
- (Chúng tôi hình dung sự thay đổi của giá theo chuỗi thời gian.)

In [27]:
```
shopsgroup= shops.groupby('month').mean()
fix,ax = plt.subplots(figsize=(12,5))
ax.xaxis.set(ticks= range(0,13))
shopsgroup['price'].plot(x=shops.invoice_date)
plt.title('Price status by month')
plt.show()
```

- We visualized the price change by month, the highest price increase was in the 10th month.
- (Chúng tôi hình dung giá thay đổi theo tháng, giá tăng cao nhất là vào tháng thứ 10.)

In [28]:
```
shopsgroup = shops.groupby('day').mean()
fix,ax = plt.subplots(figsize=(12,5))
shopsgroup['price'].plot(x = shops.invoice_date)
plt.title('change in price by days')
plt.show()
```



- We visualized the price change statuses by days.
- (Thay đổi giá theo ngày.)

Monthly Revenue Changes Over Years

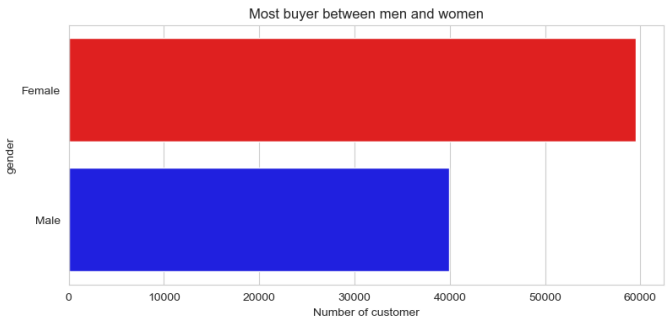
In [29]:
```
monthly_revenue = shops.groupby(['year', 'month'])['price'].sum().reset_index()
sns.set_style("whitegrid")
gsize=(10, 6))
```

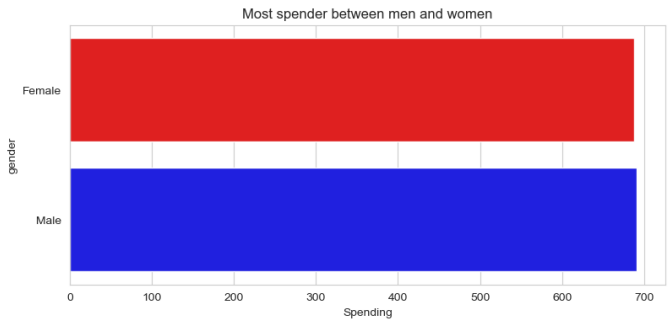
```
sns.lineplot(data=monthly_revenue, x="month", y="price", hue="year", marker='o')
plt.xlabel("Month")
plt.ylabel("Revenue")
plt.title("Monthly Revenue Changes Over Years")
plt.legend(title="Year")
plt.show()
```

What gender have the most purchased and spending

In [30]:
```
#Let's see who the most buyer between man and women
fig = plt.figure(figsize=(20,4))
ax1 = fig.add_subplot(121)


sns.countplot(y=shops['gender'],ax =ax1,palette=['red','blue'])
ax1.set_title('Most buyer between men and women')
ax1.set_xlabel('Number of customer')

#Let's see who the most spender between man and women
ax2 = fig.add_subplot(122)

shops_gender = shops.groupby('gender').mean().reset_index()

sns.barplot(y =shops_gender['gender'],x =shops_gender['price'],ax=ax2,palette=['red','bl
ax2.set_title('Most spender between men and women')
ax2.set_xlabel('Spending')
```

Out[30]:
```
Text(0.5, 0, 'Spending')
```



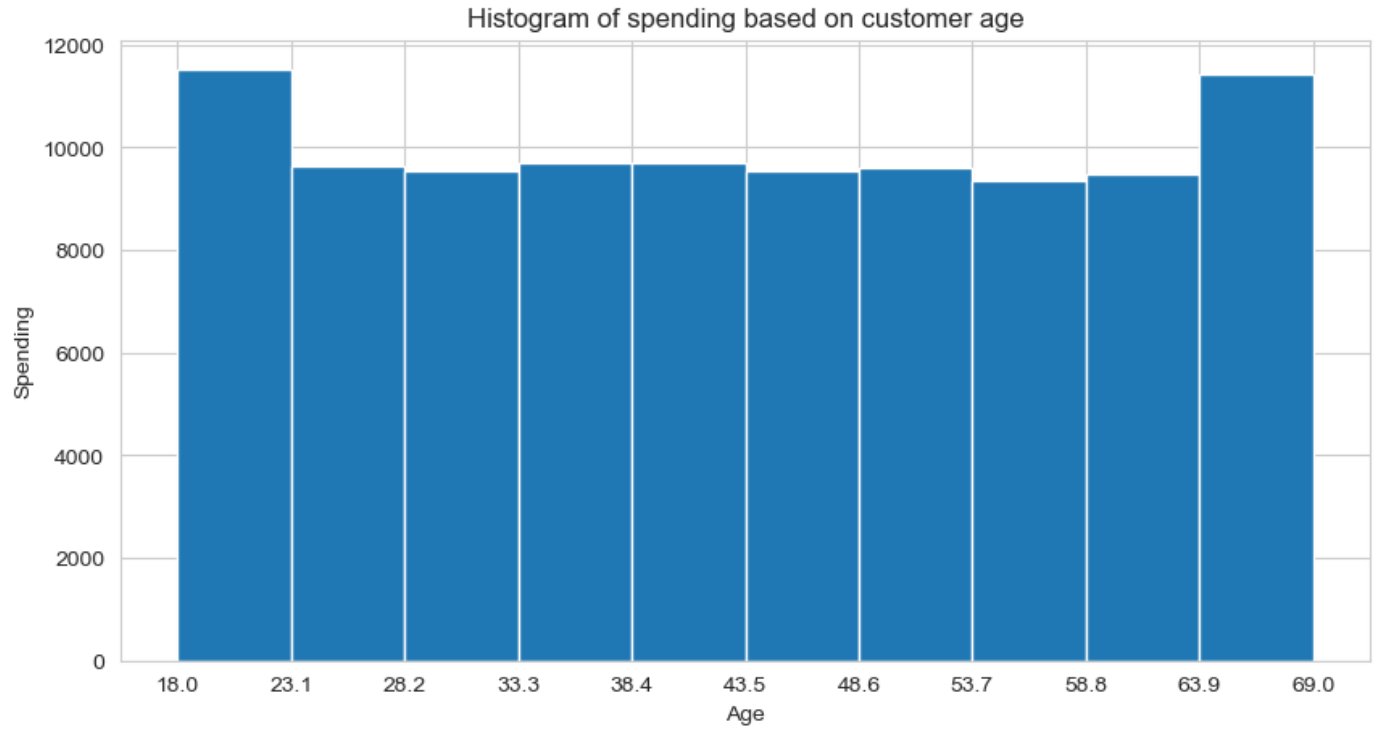


- According to the statistics, the number of female customers is higher than that of males but their spending amounts are equal. Women tend to go shopping more often, while men seem to spend more money each time they make a purchase.
- (Theo thống kê, số lượng khách hàng nữ nhiều hơn nam tuy nhiên số tiền chi tiêu của họ lại bằng nhau .Phụ nữ có xu hướng thường đi mua sắm nhiều hơn nhưng nam giới có vẻ chi nhiều tiền hơn với mỗi lần mua của họ.)

Distribution of data based on age and spending

In [31]:
```
shop_age_price = shops['age']
count,shop_age = np.histogram(shop_age_price)
shop_age_price.plot(kind ='hist',xticks= shop_age,figsize = (10,5))
plt.title('Histogram of spending based on customer age')
plt.xlabel('Age')
plt.ylabel('Spending')
```

Out[31]:
```
Text(0, 0.5, 'Spending')
```

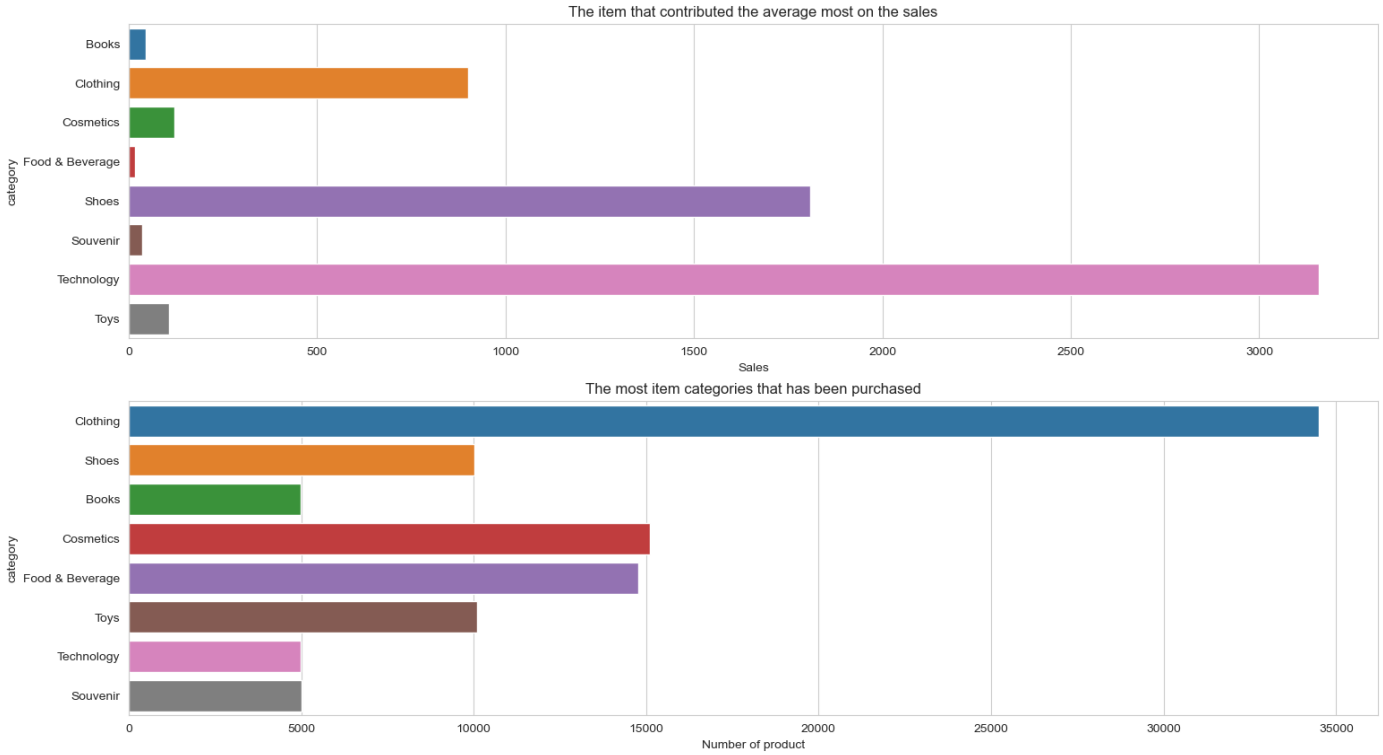


The most item in categories that has been purchased and contributed on average sales

In [32]:
```
fig = plt.figure(figsize=(18,10))
# Let's see the most item categories that has been purchased
ax1 = fig.add_subplot(212)
sns.countplot(y=shops['category'])
ax1.set_title('The most item categories that has been purchased')
```

```
ax1.set_xlabel('Number of product')
# The item that contributed the most on the sales
ax2 = fig.add_subplot(211)
most_sales = shops.groupby('category').mean().reset_index()
sns.barplot(x=most_sales['price'],y=most_sales['category'])
ax2.set_title('The item that contributed the average most on the sales')
ax2.set_xlabel('Sales')
```

Out[32]:
```
Text(0.5, 0, 'Sales')
```

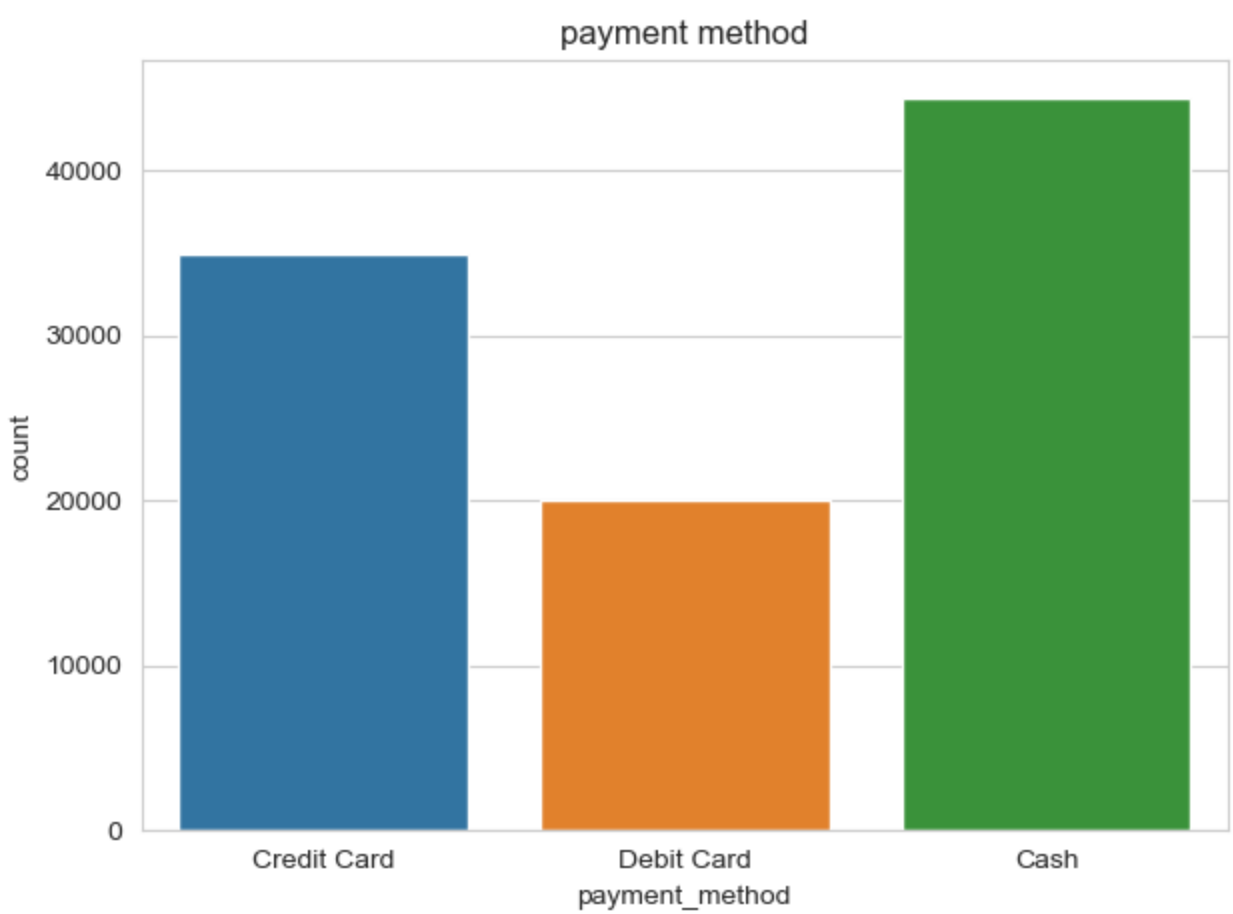


- Clothing, Shoes, and technology products contribute the most to the sales, with technology items generating exceptional high revenue.
- The category with the highest number of sales is clothing, cosmetics, food and beverage.

- (Mặt hàng quần áo, giày dép và công nghệ đóng góp vào doanh số nhiều nhất trong đó đồ công nghệ có doanh thu cao vượt trội.)
- (Danh mục có số lượng bán ra nhiều nhất là quần áo, mỹ phẩm , thức ăn và đồ uống.)

In [33]:
```
# popular payment methods
fix,ax = plt.subplots(figsize=(7,5))
sns.countplot(x=shops['payment_method'])
ax.set_title('payment method')
```

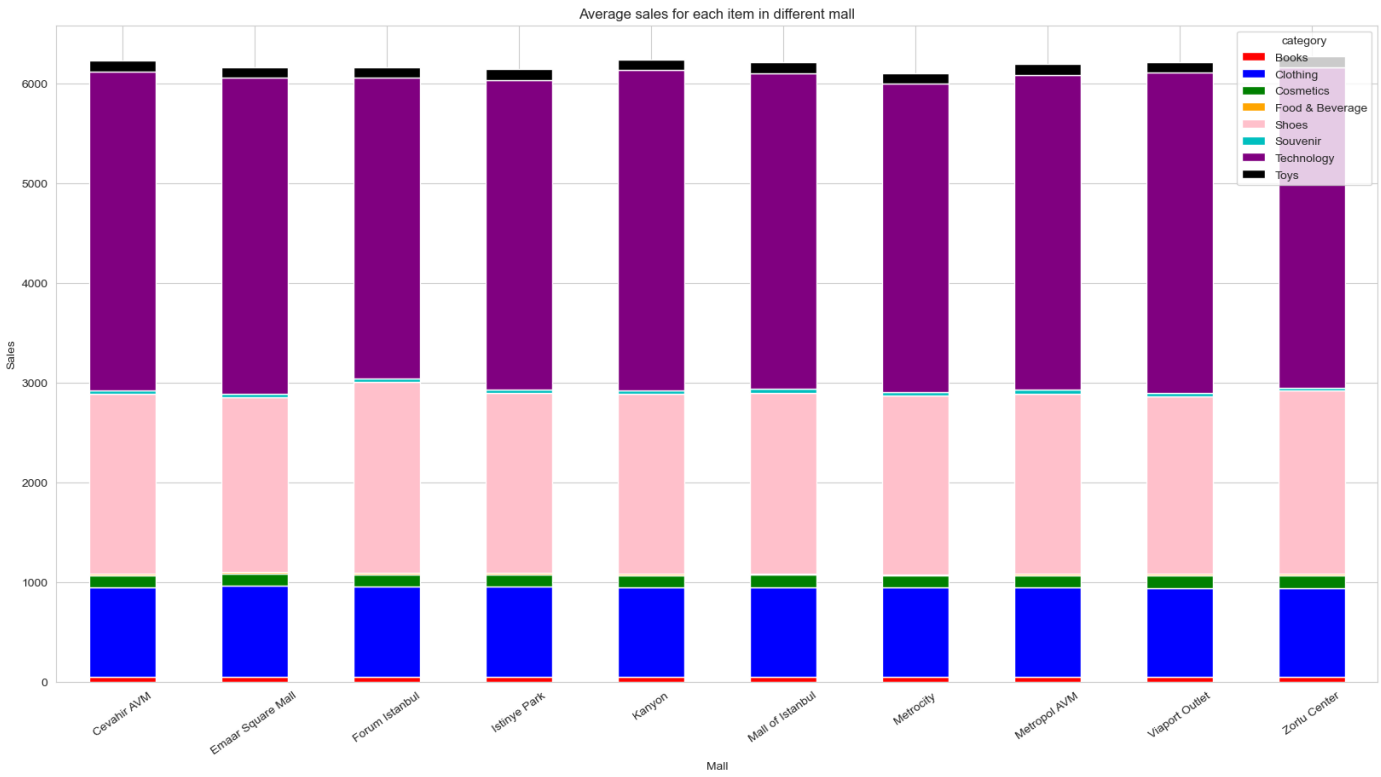
Out[33]:
```
Text(0.5, 1.0, 'payment method')
```



- Most of them prefer shopping by Cash. Priority: Cash >> Credit Card >> Debit Card.
- (Hầu hết họ thích mua sắm bằng tiền mặt hơn. Ưu tiên: Tiền mặt >> Thẻ tín dụng >> Thẻ ghi nợ.)

Average Sales for each item in different mall

In [34]:
```
colors=['red', 'blue', 'green', 'orange', 'pink','c' ,'purple', 'k']
df_mall_cat = shops.groupby(["shopping_mall",'category'])['price'].mean().unstack('categ
.plot(kind='bar',figsize=(20,10),stacked=True,color=colors)
plt.xlabel("Mall")
plt.title("Average sales for each item in different mall")
plt.ylabel("Sales")
plt.xticks(rotation=35)
plt.show()
shops.groupby(["shopping_mall",'category'])['price'].mean().unstack('category')
```

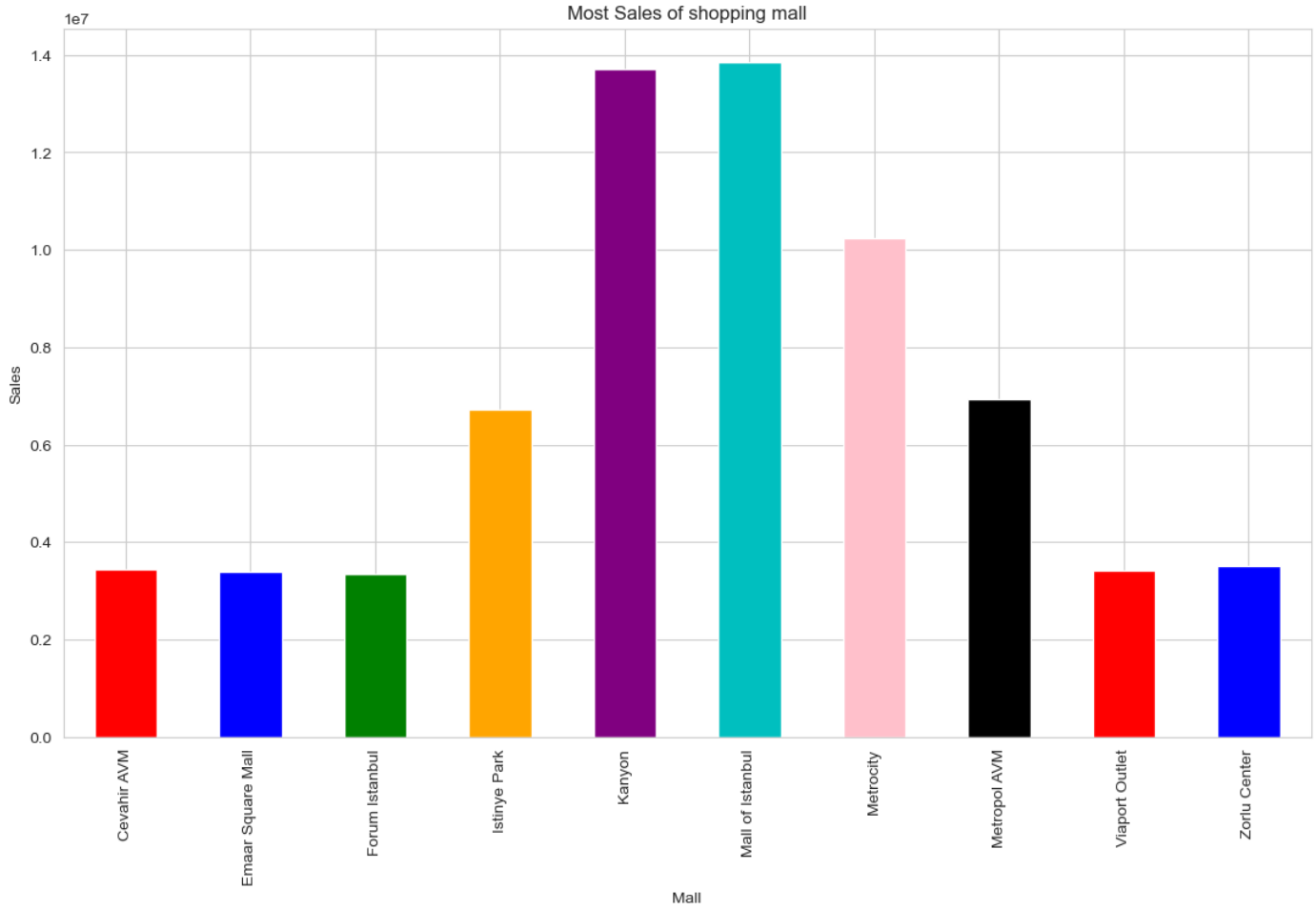


Out[34]:

shopping_mall	category	Books	Clothing	Cosmetics	Food & Beverage	Shoes	Souvenir	Technology	Toys
Cevahir AVM		46.506977	899.025101	120.757978	15.534184	1804.184510	34.603500	3199.218750	108.008093
Emaar Square Mall		46.862288	915.689303	121.552000	15.939408	1749.893253	34.477652	3173.954373	106.515862
Forum Istanbul		45.450000	909.264962	123.669766	15.219888	1911.895262	34.830460	3019.871795	103.672768
Istinye Park		44.859740	909.726573	122.593525	15.718081	1804.706993	35.668311	3100.410678	107.484690
Kanyon		44.403110	899.932901	123.136718	15.632468	1804.675781	34.217213	3212.136409	107.952894
Mall of Istanbul		45.804732	902.147196	122.553239	15.702381	1813.266950	36.142753	3166.519174	108.075520
Metrocity		45.752194	895.967790	120.274614	15.764581	1791.447721	34.919797	3091.515544	109.081347
Metropol AVM		46.723109	898.282031	120.555185	15.810666	1808.892263	35.035367	3159.051724	108.803717
Viaport Outlet		44.341463	896.199110	126.245422	15.597244	1776.358290	33.938800	3215.625000	106.058563
Zorlu Center		46.959195	890.867825	126.929450	15.452907	1837.514701	32.679689	3213.000000	106.611774

The most sales in shopping mall

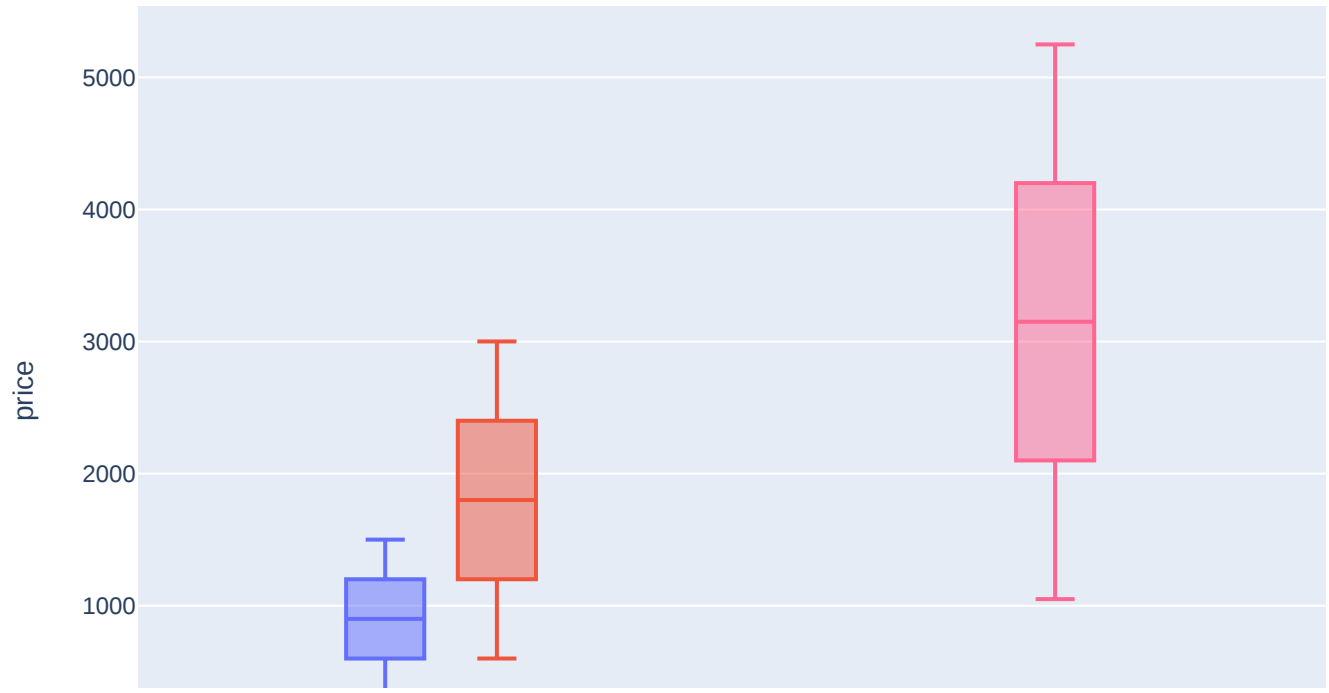
In [35]:
```
colors = ['red', 'blue', 'green', 'orange', 'purple','c' ,'pink', 'k']
shop_mall = shops.groupby('shopping_mall')['price'].sum()
shop_mall.plot(kind='bar',figsize=(14,8),color=colors)
plt.title('Most Sales of shopping mall')
plt.xlabel('Mall')
plt.ylabel('Sales')
plt.show()
```



- Highest Shopping frequency & Spending observed in Mall of Istanbul, Kanyon & Istanbul.
- (Tần suất mua sắm & Chi tiêu cao nhất được quan sát thấy ở Mall of Istanbul, Kanyon & Istanbul.)

Sales for each category

In [36]:
```
# Average sales for each category with box plot
px.box(shops,y= 'price', color='category')
```



Correlation matrix for each features

In [37]:
```
shop_corr = shops.corr() # Generate correlation matrix
shop_corr
```

Out[37]:

	age	quantity	price	year	month	day
age	1.000000	0.000667	0.001694	-0.002390	0.005692	0.008171
quantity	0.000667	1.000000	0.344880	0.000821	0.005353	-0.000529
price	0.001694	0.344880	1.000000	-0.000831	-0.002933	0.001237
year	-0.002390	0.000821	-0.000831	1.000000	-0.154802	-0.076851
month	0.005692	0.005353	-0.002933	-0.154802	1.000000	-0.028991
day	0.008171	-0.000529	0.001237	-0.076851	-0.028991	1.000000

In [38]:
```
sns.heatmap(shop_corr,  fmt=".2f",annot=True)
```

Out[38]:
```
<AxesSubplot:>
```

In [ ]: