# 1.Data Analysis Task

## 1.1Target

The dataset here is a sample of the transactions made in a retail store. The store wants to know better the customer purchase behavior against different products.

## 1.2Feature

The data has only 12 features, respectively,

| Feature | Info |
| --- | --- |
| User_ID | The ID of User |
| Product_ID | The ID of Project |
| Age | Age Info |
| Occupation | Career of people |
| City_Category | The Category of City |
| Stay_In_Current_City_Years | The years people stay in city |
| Marital_Status | Marriage ried |
| Product_Category_1 | Product Category 1 |
| Product_Category_2 | Product Category 2 |
| Product_Category_3 | Product Category 3 |
| Purchase | Purchase Amount |

The useful features are:

Age、Occupation、City_Category、Stay_In_Current_City_Years、Marital_Status、Product_Category_1、Purchase

(I dropped both features because Product_Category_2 and Product_Category_3 have a lot of missing values)

## 1.3 Problems want to solve

Purchase of of age ages

User purchases in different occupations

Purchase of Different Genders

User purchases in different cities

Purchase of users with different living years

Which category is the most popular product?

How much is the overall data discrete?

## 2.Specific implementation

## 2.1Data processing

In the process of using Dash I found it slow to render the raw data directly with dash and the data 1BlackFriday has 58w, so to preprocess the data first. Specific code is in the datainfo.py:

For example, the commodity category:

```
#商品类别处理
data=pd.DataFrame()
data['Product_Category_1']=df['Product_Category_1']
data['Purchase']=df['Purchase']
data['User_ID']=df['User_ID']
temp=data.groupby('Product_Category_1').sum()
temp.to_csv('CategoryPurchase.csv')
```

Advance through data.groupby, and then read the generated new table directly when reading

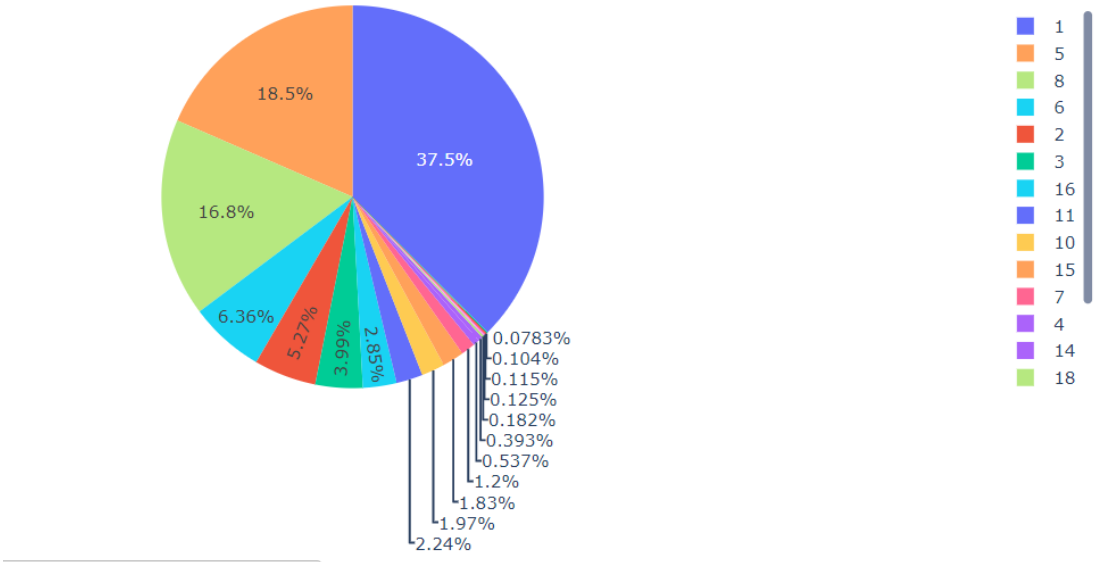
```
CategoryPurchase=pd.read_csv('./dataset/CategoryPurchase.csv')
```

## 2.2Sales volume statistics of commodity category

Show it off through a pie chart:

```
CategoryFigure=px.pie(CategoryPurchase,names="Product_Category_1",values="Purcha
se",color="Product_Category_1")
```

The effect shown follows:




You can clearly see how much of each commodity's sales share of the total.

## 2.3Statistics of age, sex and living time

Read the processed data directly

```
Stay_In_Current_City_YearsPurchase=pd.read_csv('./dataset/Stay_In_Current_City_YearsPurchase.csv')
GenderPurchase=pd.read_csv('./dataset/GenderPurchase.csv')
AgePurchase=pd.read_csv('./dataset/AgePurchase.csv')

Stay_In_Current_City_YearsFigure=px.bar(Stay_In_Current_City_YearsPurchase,x='Stay_In_Current_City_Years',y='Purchase',color="City_Category",barmode="group")
GenderFigure=px.bar(GenderPurchase,x='Gender',y='Purchase',color="City_Category",barmode="group")
AgeFigure=px.bar(AgePurchase,x='Age',y='Purchase',color="City_Category",barmode="group")
```

The effect are as follows:

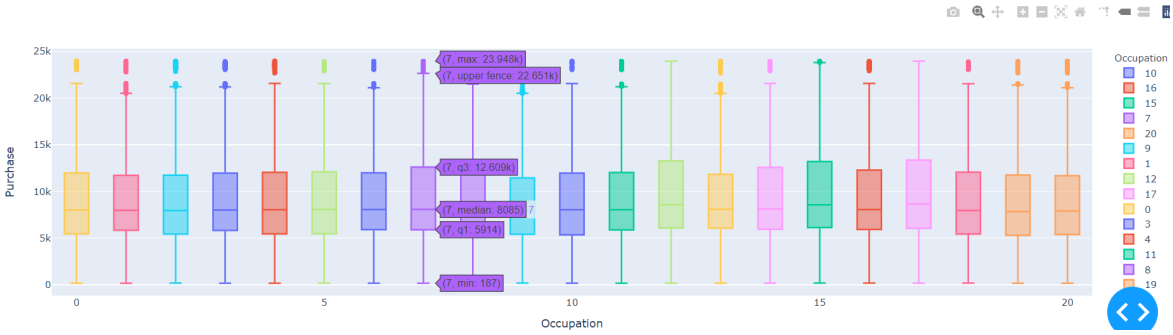
View at top left

Implementation through the callback function:

```
@app.callback(
    Output('DropDown-graph', 'figure'),
    Input('info-col', 'value')
    )
def update_graph(info_col):
    if info_col=='Age':
        DropDownFigure=AgeFigure
    elif info_col=='Gender':
        DropDownFigure=GenderFigure
    else:
        DropDownFigure=Stay_In_Current_City_YearsFigure
    return DropDownFigure
```


## 2.4Data discrete case box type diagram:

Box graphs can describe the discrete distribution of the data in a relatively stable way, and by describing the purchasing power of occupational characteristic box graphs, it is found that the discrete situation of our data is basically the same. (See the purchasing power average, maximum, minimum, 75%, 25% for each profession)


(7, max: 23.948k)
(7, upper fence: 22.651k)
(7, q3: 12.609k)
(7, median: 8085)
(7, q1: 5914)
(7, min: 187)
Purchase
Occupation
Occupation
10
16
15
7
20
9
1
12
17
0
3
4
11
8
19