

```
In [29]: import pandas as pd
import numpy as np
```

```
In [43]: path = "../Maven Python Course/project_transactions.csv"
```

```
In [44]: cols = ["household_key", "BASKET_ID", "DAY", "PRODUCT_ID", "QUANTITY", "SALES_VALUE"]
```

```
In [45]: dtypes = {"DAY": "Int16", "QUANTITY": "Int32", "PRODUCT_ID": "Int32"}
```

```
In [47]: transactions = pd.read_csv(path, dtype = dtypes, usecols = cols)
```

```
In [50]: transactions.describe().round()
```

	household_key	BASKET_ID	DAY	PRODUCT_ID	QUANTITY	SALES_VALUE
count	2146311.0	2.146311e+06	2146311.0	2146311.0	2146311.0	2146311.0
mean	1056.0	3.404897e+10	390.0	2884715.0	101.0	3.0
std	605.0	4.723748e+09	190.0	3831949.0	1152.0	4.0
min	1.0	2.698490e+10	1.0	25671.0	0.0	0.0
25%	548.0	3.040798e+10	229.0	917231.0	1.0	1.0
50%	1042.0	3.281176e+10	392.0	1027960.0	1.0	2.0
75%	1581.0	4.012804e+10	555.0	1132771.0	1.0	3.0
max	2099.0	4.230536e+10	711.0	18316298.0	89638.0	840.0

```
In [51]: transactions.info(memory_usage = 'deep')
```

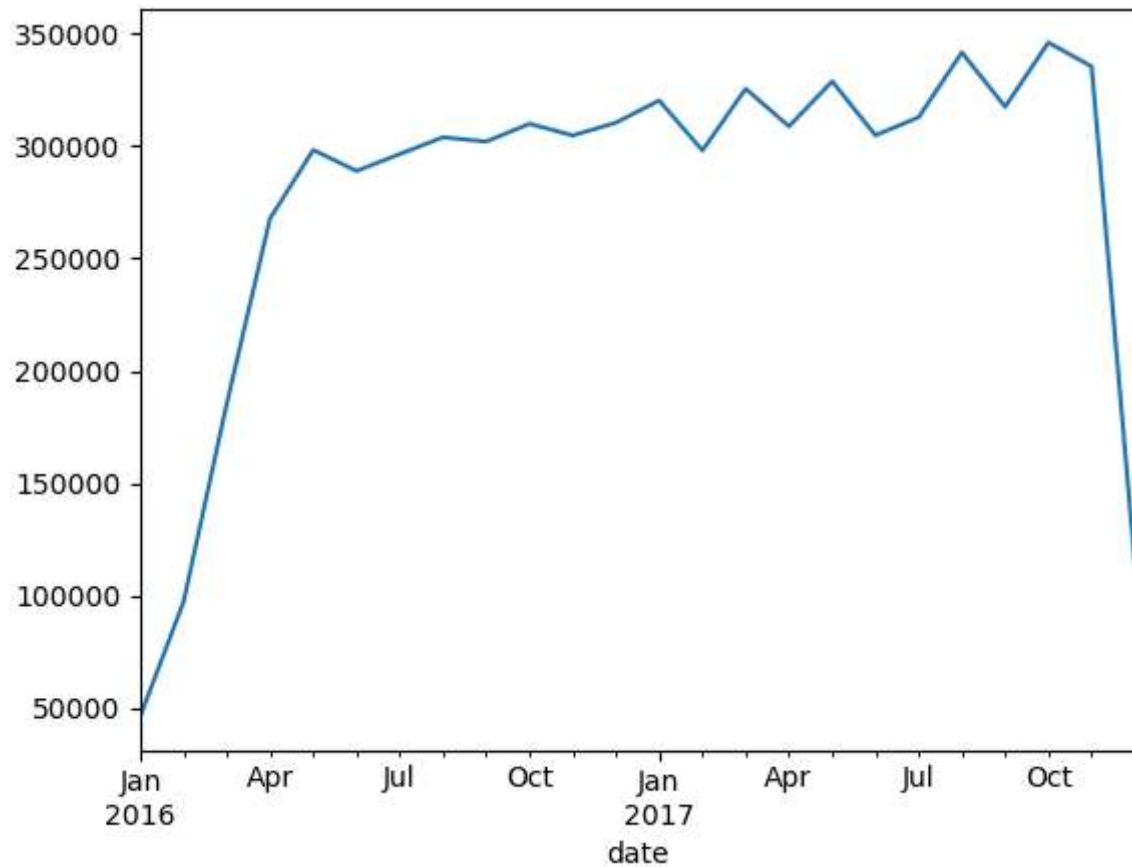
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2146311 entries, 0 to 2146310
Data columns (total 6 columns):
#   Column          Dtype
---  -
0   household_key    int64
1   BASKET_ID        int64
2   DAY              Int16
3   PRODUCT_ID       Int32
4   QUANTITY         Int32
5   SALES_VALUE      float64
dtypes: Int16(1), Int32(2), float64(1), int64(2)
memory usage: 75.7 MB
```

```
In [52]: transactions = (
    transactions
    .assign(date = (pd.to_datetime("2016", format='%Y')
                    + pd.to_timedelta(transactions["DAY"].sub(1).astype(str) + " days"
                                     )
    )
    .drop(["DAY"], axis=1)
)
```

TIME BASED ANALYSIS

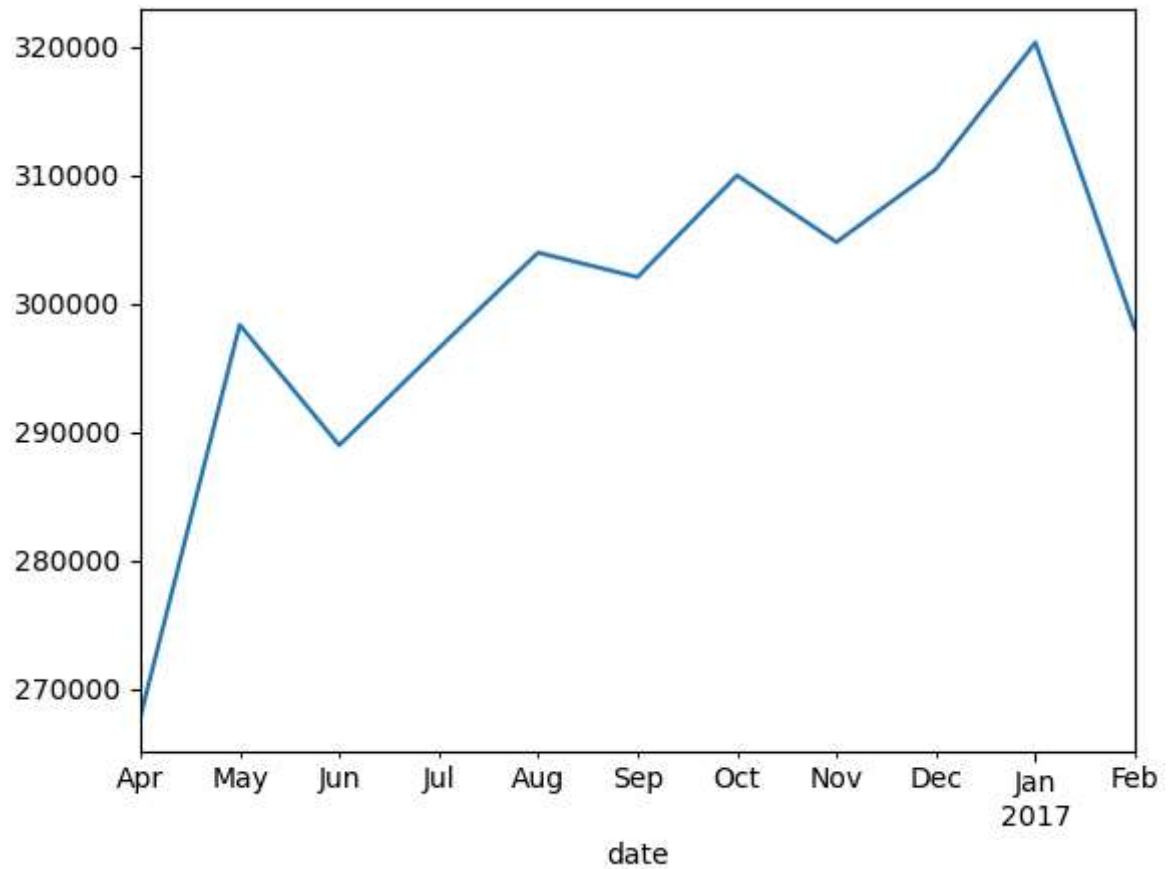
```
In [53]: (transactions.set_index("date")
          .loc[:, "SALES_VALUE"]
          .resample("M")
          .sum()
          .plot())
```

Out[53]: <Axes: xlabel='date'>



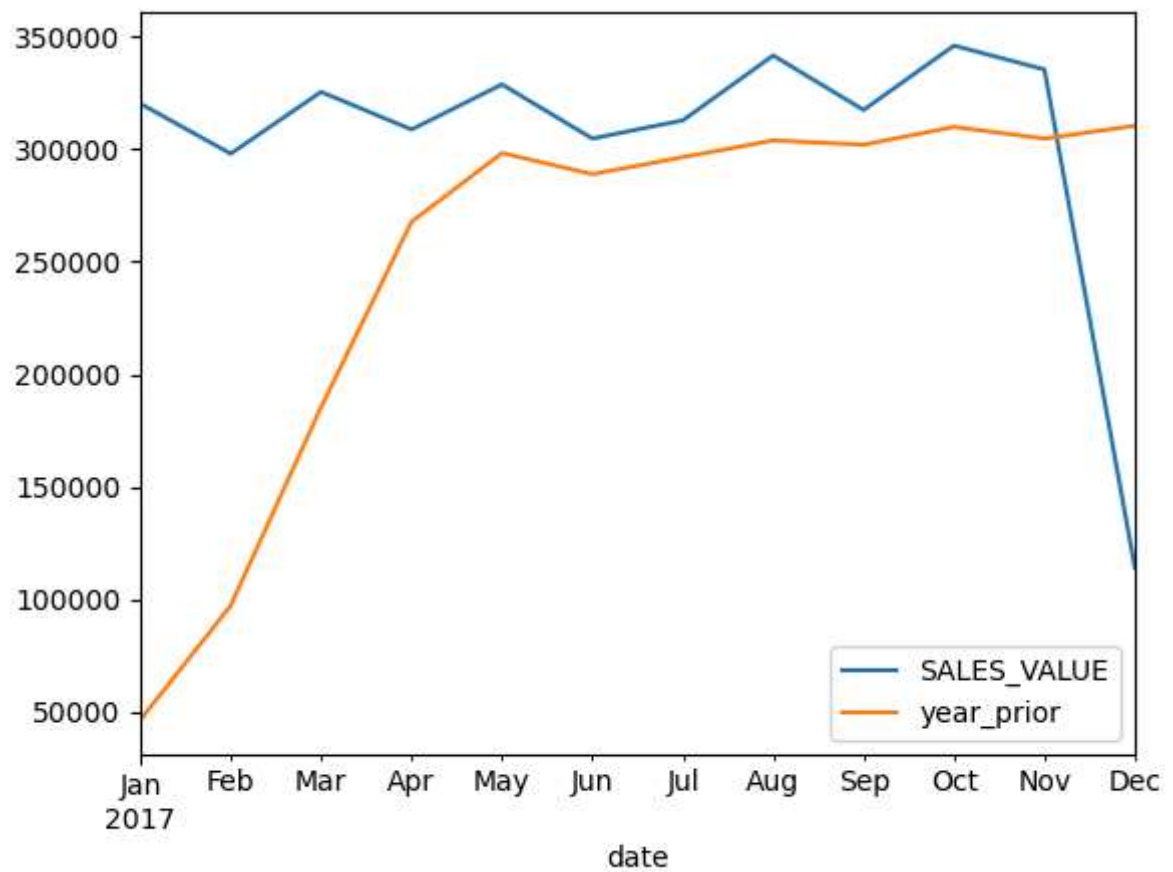
```
In [54]: (transactions.set_index("date")
          .loc["2016-04":"2017-02", "SALES_VALUE"]
          .resample("M")
          .sum()
          .plot())
```

Out[54]: <Axes: xlabel='date'>



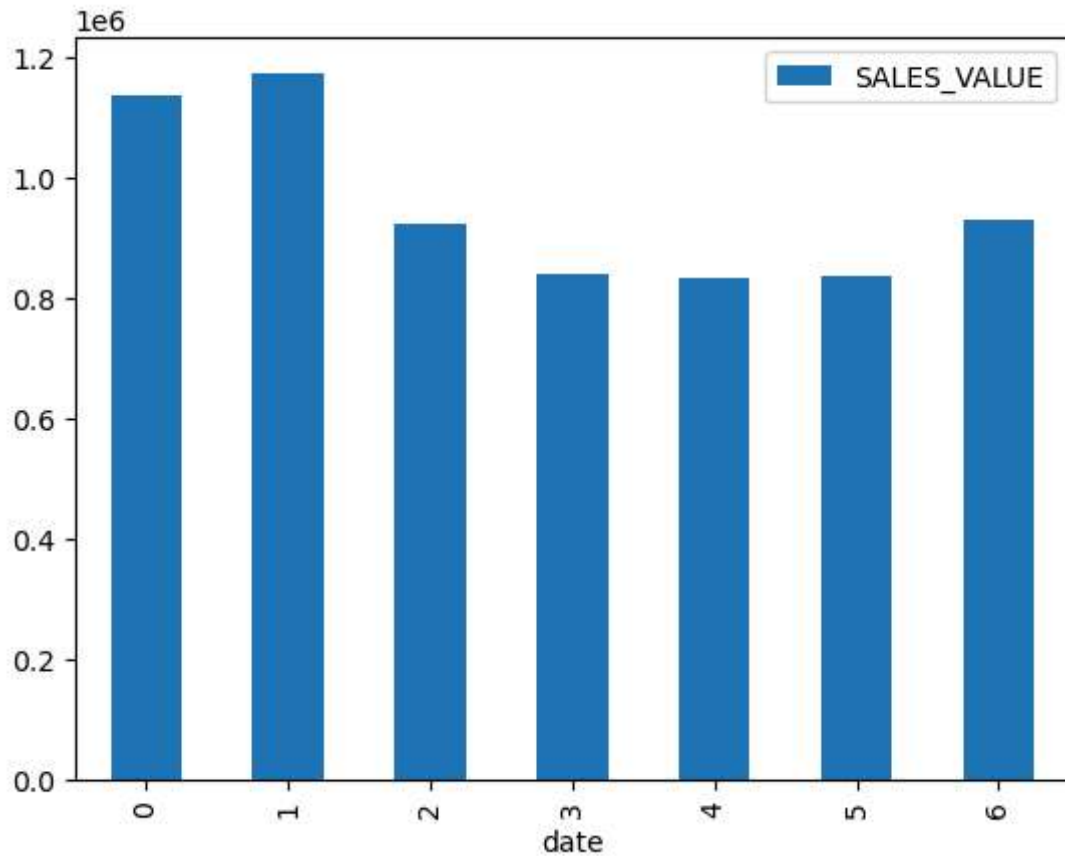
```
In [58]: (transactions.set_index("date")
          .loc[:, ["SALES_VALUE"]]
          .resample("M")
          .sum()
          .assign(year_prior = lambda x: x["SALES_VALUE"].shift(12))
          .loc["2017"]
          .plot()
          )
```

```
Out[58]: <Axes: xlabel='date'>
```



```
In [60]: (transactions.groupby(transactions["date"].dt.dayofweek)
          .agg({"SALES_VALUE": "sum"})
          .plot.bar()
          )
```

```
Out[60]: <Axes: xlabel='date'>
```



DEMOGRAPHICS

```
In [62]: dem_cols = ["AGE_DESC", "INCOME_DESC", "household_key", "HH_COMP_DESC"]
dem_dtypes = {"AGE_DESC": "category", "INCOME_DESC": "category", "HH_COMP_DESC": "category", "household_key": "int"}
demographics = pd.read_csv('../Maven Python Course/hh_demographic.csv',
                           usecols=dem_cols,
                           dtype=dem_dtypes
                           )
```

```
In [63]: demographics.head()
```

```
Out[63]:
```

	AGE_DESC	INCOME_DESC	HH_COMP_DESC	household_key
0	65+	35-49K	2 Adults No Kids	1
1	45-54	50-74K	2 Adults No Kids	7
2	25-34	25-34K	2 Adults Kids	8
3	25-34	75-99K	2 Adults Kids	13
4	45-54	50-74K	Single Female	16

```
In [64]: demographics.tail(10)
```

Out[64]:

	AGE_DESC	INCOME_DESC	HH_COMP_DESC	household_key
791	25-34	150-174K	2 Adults No Kids	2479
792	45-54	75-99K	Single Male	2483
793	35-44	25-34K	Single Male	2486
794	45-54	35-49K	Single Female	2488
795	35-44	50-74K	2 Adults No Kids	2489
796	35-44	50-74K	2 Adults No Kids	2494
797	45-54	75-99K	Unknown	2496
798	45-54	35-49K	Single Male	2497
799	25-34	50-74K	2 Adults No Kids	2498
800	25-34	Under 15K	2 Adults Kids	2499

In [65]: `demographics.info(memory_usage="deep")`

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 801 entries, 0 to 800
Data columns (total 4 columns):
#   Column          Non-Null Count  Dtype
---  -
0   AGE_DESC        801 non-null   category
1   INCOME_DESC     801 non-null   category
2   HH_COMP_DESC    801 non-null   category
3   household_key   801 non-null   int64
dtypes: category(3), int64(1)
memory usage: 10.9 KB
```

In [66]: `household_sales = (transactions.groupby("household_key").agg({"SALES_VALUE": "sum"}))`
`household_sales`

Out[66]:

SALES_VALUE	
household_key	
1	4330.16
2	1954.34
3	2653.21
4	1200.11
5	779.06
...	...
2095	3790.49
2096	1301.65
2097	8823.83
2098	682.46
2099	691.30

2099 rows × 1 columns

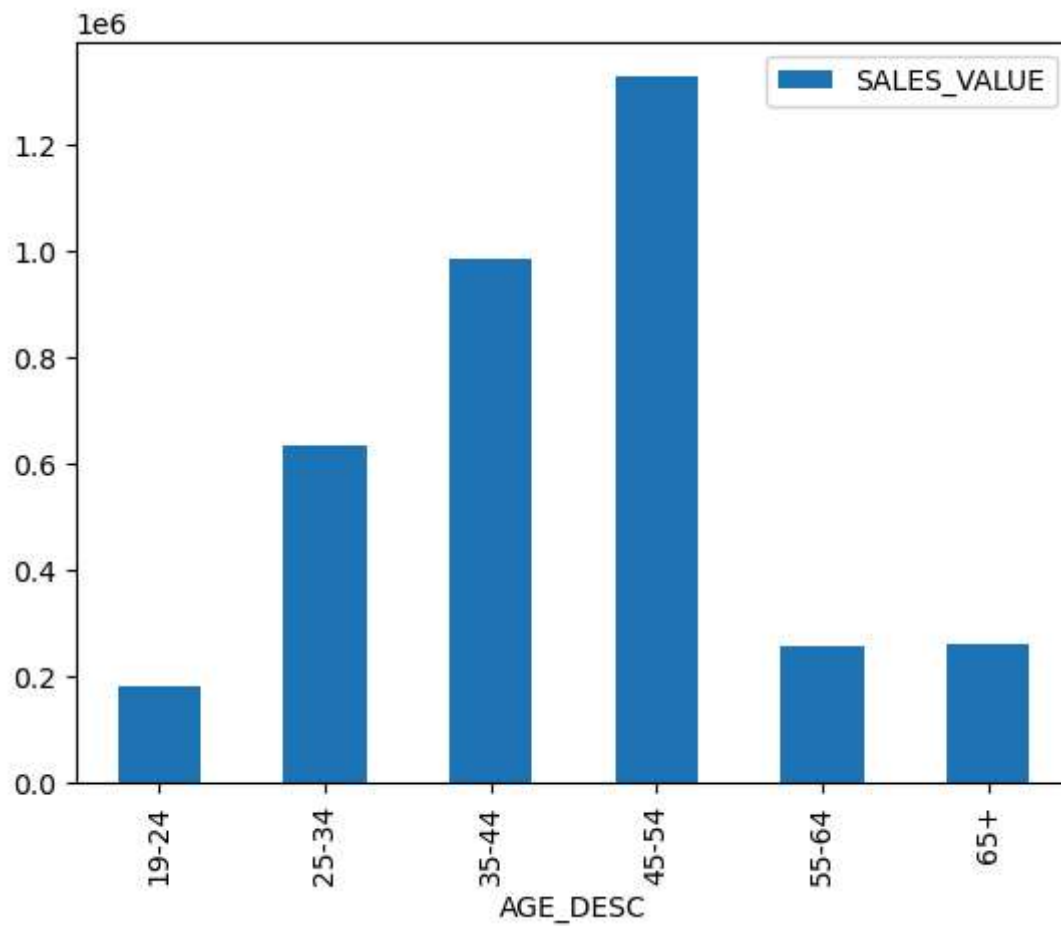
```
In [75]: household_sales_demo = (household_sales.merge(demographics,
    how="inner" ,
    left_on='household_key' ,
    right_on="household_key",
    )
    )
```

```
In [76]: household_sales_demo.info(memory_usage="deep")
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 668 entries, 0 to 667
Data columns (total 5 columns):
#   Column          Non-Null Count  Dtype
---  -
0   household_key    668 non-null   int64
1   SALES_VALUE      668 non-null   float64
2   AGE_DESC         668 non-null   category
3   INCOME_DESC      668 non-null   category
4   HH_COMP_DESC     668 non-null   category
dtypes: category(3), float64(1), int64(1)
memory usage: 14.7 KB
```

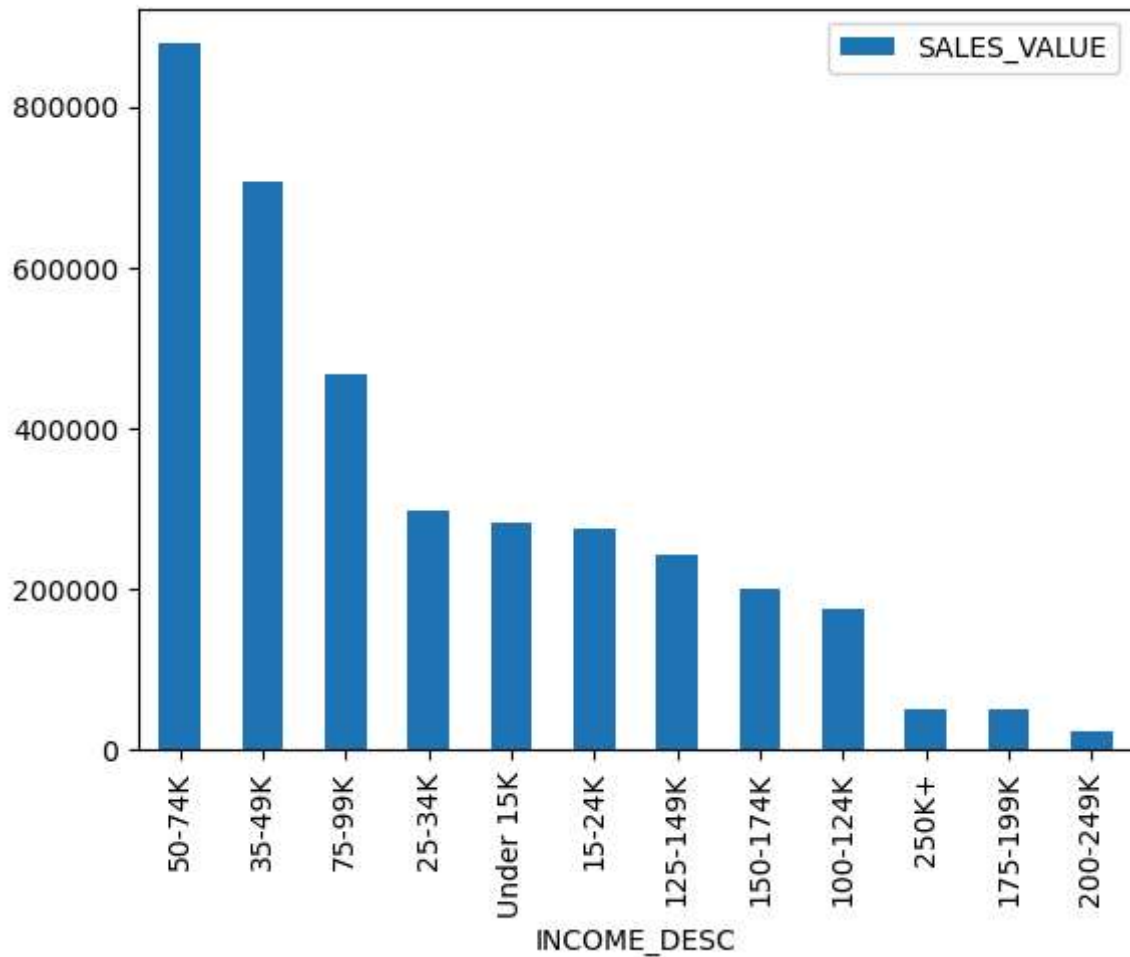
```
In [84]: (household_sales_demo
    .groupby(["AGE_DESC"])
    .agg({"SALES_VALUE": "sum"})
    .plot.bar()
    )
```

```
Out[84]: <Axes: xlabel='AGE_DESC'>
```



```
In [85]: (household_sales_demo
          .groupby(["INCOME_DESC"])
          .agg({"SALES_VALUE": "sum"})
          .sort_values("SALES_VALUE", ascending=False)
          .plot.bar())
```

```
Out[85]: <Axes: xlabel='INCOME_DESC'>
```



```
In [86]: (household_sales_demo.pivot_table(index="AGE_DESC",
      columns="HH_COMP_DESC",
      values="SALES_VALUE",
      aggfunc="mean",
      margins=True)
      .style.background_gradient(cmap="RdYlGn", axis=None)
      )
```

Out[86]:

HH_COMP_DESC	1 Adult Kids	2 Adults Kids	2 Adults No Kids	Single Female	Single Male	Unknown	
AGE_DESC							
19-24	7268.796667	5428.945000	4020.800000	4576.095556	3216.835000	4911.275000	4692.0
25-34	5512.196875	5753.973514	5638.515833	4807.440588	4909.522381	7356.270000	5435.5
35-44	6297.737778	6691.772264	6260.412444	6015.192069	4844.192000	4227.691818	6090.5
45-54	6632.569167	6610.484490	5839.527027	4549.365405	4636.637083	4843.995682	5534.8
55-64	3064.870000	4695.655000	5752.413684	4816.148462	3922.546250	7973.750000	5168.9
65+	4040.810000	5536.866667	4614.108571	4059.699412	3871.556000	2879.290000	4340.9
All	6032.802143	6280.069103	5599.857756	4895.928361	4544.646750	4936.127778	5468.3

PRODUCT DEMOGRAPHICS

```
In [88]: product_cols=["PRODUCT_ID", "DEPARTMENT"]
product_dtypes = {"PRODUCT_ID": "Int32", "DEPARTMENT": "category"}
product = pd.read_csv('../Maven Python Course/product.csv',
                      usecols=product_cols,
                      dtype =product_dtypes )
```

```
In [89]: product.dtypes
```

```
Out[89]: PRODUCT_ID      Int32
DEPARTMENT    category
dtype: object
```

```
In [92]: trans_demo_dept = (transactions
                           .merge(demographics,
                                how="inner" ,
                                left_on = "household_key" ,
                                right_on = "household_key" , )
                           .merge(product,
                                how="inner" ,
                                left_on = "PRODUCT_ID" ,
                                right_on = "PRODUCT_ID" , )
                           )
```

```
In [93]: trans_demo_dept.info(memory_usage = "deep")
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1161575 entries, 0 to 1161574
Data columns (total 10 columns):
#   Column          Non-Null Count  Dtype
---  -
0   household_key    1161575 non-null  int64
1   BASKET_ID        1161575 non-null  int64
2   PRODUCT_ID       1161575 non-null  Int32
3   QUANTITY         1161575 non-null  Int32
4   SALES_VALUE      1161575 non-null  float64
5   date             1161575 non-null  datetime64[ns]
6   AGE_DESC         1161575 non-null  category
7   INCOME_DESC      1161575 non-null  category
8   HH_COMP_DESC     1161575 non-null  category
9   DEPARTMENT       1161575 non-null  category
dtypes: Int32(2), category(4), datetime64[ns](1), float64(1), int64(2)
memory usage: 51.0 MB
```

```
In [95]: (trans_demo_dept.pivot_table(index="DEPARTMENT" ,
                                       columns="AGE_DESC" ,
                                       values = "SALES_VALUE" ,
                                       aggfunc="sum")
         .style.background_gradient(cmap="RdYlGn" , axis=1))
```

Out[95]:

AGE_DESC	19-24	25-34	35-44	45-54	55-64	6
DEPARTMENT						
	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
AUTOMOTIVE	11.640000	21.250000	72.580000	55.920000	0.000000	16.370000
CHARITABLE CONT	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
CHEF SHOPPE	81.300000	134.160000	348.530000	418.240000	80.860000	149.240000
CNTRL/STORE SUP	2.000000	0.000000	1.000000	9.950000	2.000000	0.100000
COSMETICS	698.630000	2273.030000	4362.020000	5187.570000	986.260000	600.900000
COUP/STR & MFG	7.490000	48.420000	121.200000	154.550000	40.680000	20.490000
DAIRY DELI	3.800000	3.850000	7.390000	16.750000	3.140000	1.940000
DELI	4043.300000	18181.940000	34577.290000	44334.220000	9850.540000	10462.330000
DELI/SNACK BAR	0.000000	0.000000	6.980000	1.560000	0.000000	3.310000
DRUG GM	25297.430000	85298.050000	126480.340000	177007.130000	29220.930000	32759.760000
ELECT &PLUMBING	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
FLORAL	776.990000	2355.570000	5246.600000	6835.690000	1112.690000	1160.220000
FROZEN GROCERY	1.640000	53.050000	108.960000	84.500000	54.220000	20.190000
GARDEN CENTER	41.980000	380.110000	701.830000	1487.900000	248.070000	441.810000
GM MERCH EXP	0.000000	0.000000	17.760000	30.370000	12.050000	2.950000
GRO BAKERY	0.000000	0.000000	0.000000	2.180000	0.000000	0.000000
GROCERY	99008.270000	327926.160000	490616.030000	667162.980000	127082.010000	129117.270000
HBC	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
HOUSEWARES	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
KIOSK-GAS	8465.180000	50817.910000	92614.660000	96858.440000	16329.770000	17853.990000
MEAT	11957.340000	37162.660000	61003.950000	87407.670000	20001.910000	17514.190000
MEAT-PCKGD	10453.130000	30029.690000	46499.110000	59855.460000	11891.430000	10413.650000
MEAT-WHSE	0.000000	0.000000	1.000000	4.000000	1.000000	0.000000
MISC SALES TRAN	2031.730000	8200.660000	9976.190000	23617.850000	7762.980000	2657.760000
MISC. TRANS.	73.520000	757.370000	1334.700000	859.080000	688.730000	142.630000
NUTRITION	1146.400000	11067.450000	15941.860000	16366.510000	2504.010000	3114.280000

	AGE_DESC	19-24	25-34	35-44	45-54	55-64	6
DEPARTMENT							
PASTRY	2386.730000	8161.420000	13706.810000	19534.790000	3601.860000	5162.8000	
PHARMACY SUPPLY	0.000000	5.970000	3.980000	1.990000	0.000000	0.0000	
PHOTO	4.980000	5.190000	6.980000	2.490000	0.000000	1.9900	
PORK	0.000000	0.000000	0.000000	0.000000	0.000000	0.0000	
POSTAL CENTER	0.000000	0.000000	0.000000	5.960000	1.000000	0.3300	
PROD-WHS SALES	0.000000	0.000000	2.520000	5.000000	0.000000	0.0000	
PRODUCE	10170.590000	41706.460000	67779.890000	96442.730000	21326.150000	23295.7800	
RESTAURANT	1.390000	389.040000	234.110000	540.310000	33.900000	30.6500	
RX	0.000000	0.000000	26.880000	26.780000	0.000000	10.9900	
SALAD BAR	1330.150000	2050.060000	3631.810000	4770.150000	925.770000	1677.8800	
SEAFOOD	461.180000	2080.940000	3101.910000	5551.320000	1363.950000	1341.4500	
SEAFOOD- PCKGD	1500.270000	4189.130000	6346.770000	10079.840000	2975.970000	2206.1000	
SPIRITS	2983.750000	2474.460000	1491.370000	3218.340000	263.180000	141.6400	
TOYS	0.000000	0.000000	0.000000	1.490000	0.000000	0.0000	
TRAVEL & LEISUR	50.220000	173.560000	283.190000	431.480000	81.150000	133.2000	
VIDEO	0.000000	7.990000	13.990000	0.000000	0.000000	0.0000	
VIDEO RENTAL	0.000000	0.000000	0.000000	0.000000	0.000000	0.0000	

EXPORT

```
In [96]: (trans_demo_dept.pivot_table(index="DEPARTMENT" ,
                                         columns="AGE_DESC" ,
                                         values = "SALES_VALUE" ,
                                         aggfunc="sum")
          .style.background_gradient(cmap="RdYlGn" , axis=1)
          .to_excel("demographic_category_sales.xlsx" , sheet_name="sales_pivot")
          )
```