NB05

September 25, 2022

0.0.1 Import pandas and read in the csv file and set it to a dataframe called baskets

[2]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import math
from datetime import datetime, timedelta
import collections
```

- make a function for preparing data, that
  - read in data from a filename
  - clean data: remove rows with NA's, convert "id" columns into categorical type
  - make columns for date, year, month, day, hour, weekday, from the "placed_at" column
- make a function that construct a new data frame for "merchants" from the basket data, that
  - compute the total number of spending, number of orders, number of days, number of SKU's, number of top categories, number of sub categories
  - total spent = sum of (quantity times price)
  - average spent per order = total spent divided by number of orders

[3]:
```
from math import floor


def clean_data(baskets):
    baskets.dropna(inplace=True)
    for s in baskets.columns:
        if ("id" in s):
            baskets.loc[:,s] = pd.Categorical(baskets.loc[:,s].apply(lambda x:␣
 ↪floor(x)))
    return baskets

def convert_date(baskets):
    baskets['datetime'] = baskets['placed_at'].apply(lambda x: datetime.
 ↪fromisoformat(x))
    baskets['date'] = baskets['datetime'].dt.date
    baskets['year'] = baskets['datetime'].dt.year
    baskets['month'] = baskets['datetime'].dt.month
    baskets['day'] = baskets['datetime'].dt.day
```

```
    baskets['hour'] = baskets['datetime'].dt.hour
    baskets['weekday'] = baskets['datetime'].dt.weekday
    return(baskets)

def prep_data(filename):
    baskets = pd.read_csv(filename)
    baskets = clean_data(baskets)
    baskets = convert_date(baskets)
    baskets["spent"] = baskets["qty"] * baskets["price"]
    return baskets

def make_merchants(baskets):
    merchants = baskets.groupby(['merchant_id'])\
        .agg({'spent': 'sum',
                'order_id': 'nunique',
                'date': 'nunique',
                'sku_id':'nunique',
                'top_cat_id':'nunique',
                'sub_cat_id':'nunique'})\
        .reset_index()\
        .rename(columns={'spent':'total_spent', 'order_id':'num_orders', 'date':
 ↪'num_days', 'sku_id':'num_skus', 'top_cat_id':'num_top_cats', 'sub_cat_id':
 ↪'num_sub_cats'})
    merchants['avg_spent_per_order'] = merchants.total_spent / merchants.
 ↪num_orders
    return merchants

def make_skus(baskets):
    skus_by_day = baskets.groupby(['sku_id','date'])\
        .agg({'price': 'mean',
            'order_id': 'nunique',
            'merchant_id': 'nunique'})\
        .reset_index()\
        .rename(columns={'price':'avg_price_by_day', 'order_id':
 ↪'num_orders_by_day', 'merchant_id':'num_merchants_by_day'})
    return skus_by_day

def make_top_cats(baskets):
    top_cats = baskets.groupby(['top_cat_id'])\
        .agg({'price': 'mean',
            'spent': 'sum',
            'qty' : 'sum',
            'order_id': 'nunique',
            'date' : 'nunique',
            'merchant_id': 'nunique'})\
        .reset_index()\
```

```
        .rename(columns={'price':'avg_price', 'spent':'total_spent', 'qty':
↪'total_quantity','order_id':'num_orders', 'merchant_id':'num_merchants'})
    return top_cats
```

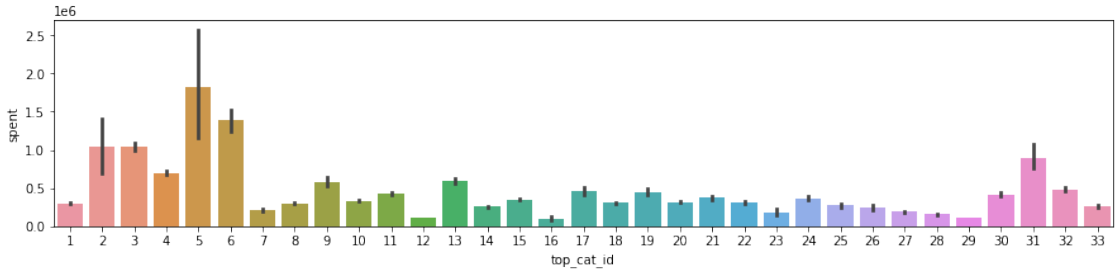
[4]:
```
filename = 'new_baskets_full.csv'
baskets = prep_data(filename)
merchants = make_merchants(baskets)
skus = make_skus(baskets)
top_cats = make_top_cats(baskets)
```

[5]:
```
top_cats.info()
```

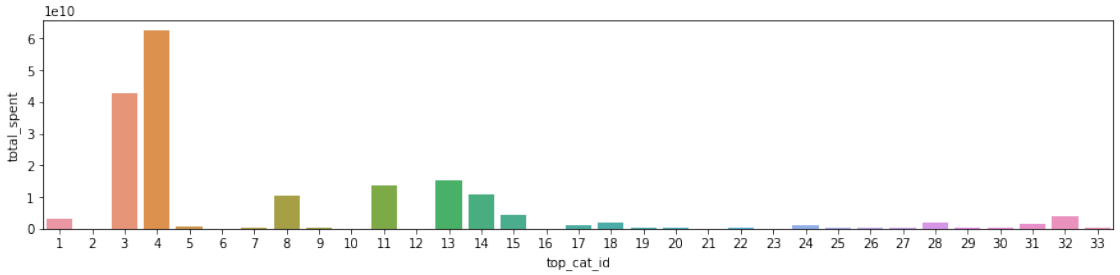
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 33 entries, 0 to 32
Data columns (total 7 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   top_cat_id      33 non-null     category
 1   avg_price       33 non-null     float64
 2   total_spent     33 non-null     float64
 3   total_quantity  33 non-null     int64
 4   num_orders      33 non-null     int64
 5   date            33 non-null     int64
 6   num_merchants   33 non-null     int64
dtypes: category(1), float64(2), int64(4)
memory usage: 3.0 KB
```

[6]:
```
filename10 = 'baskets_sample_random_10.csv'
baskets10 = prep_data(filename10)
merchants10 = make_merchants(baskets10)
skus10 = make_skus(baskets10)

filename250 = 'new_baskets_sample_top_250.csv'
baskets250 = prep_data(filename250)
merchants250 = make_merchants(baskets250)
skus250 = make_skus(baskets250)
```

[101]:
```
plt.figure(figsize=(15,3))
ax = sns.barplot(x="top_cat_id", y="spent", data=baskets)
```

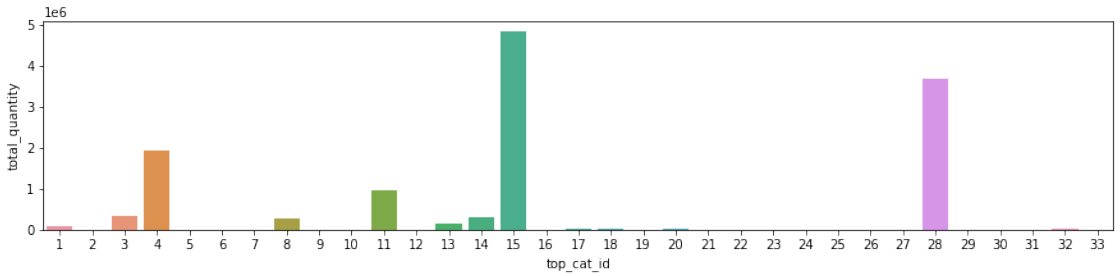


[7]:
```
plt.figure(figsize=(15,3))
ax = sns.barplot(x="top_cat_id", y="total_spent", data=top_cats)
```



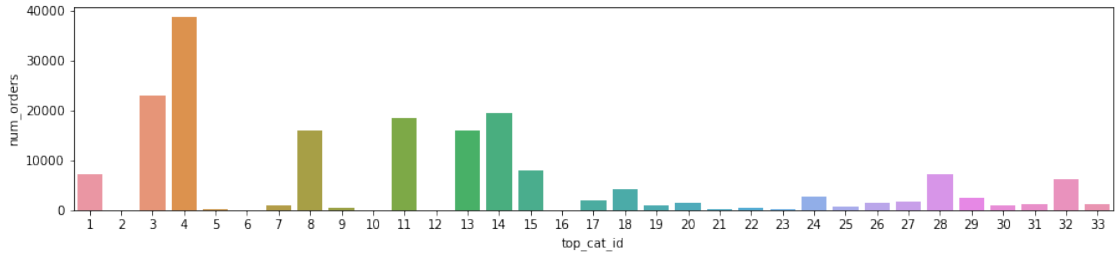
[8]:
```
plt.figure(figsize=(15,3))
sns.barplot(x='top_cat_id',y='total_quantity',data=top_cats)
```

[8]: <AxesSubplot:xlabel='top_cat_id', ylabel='total_quantity'>



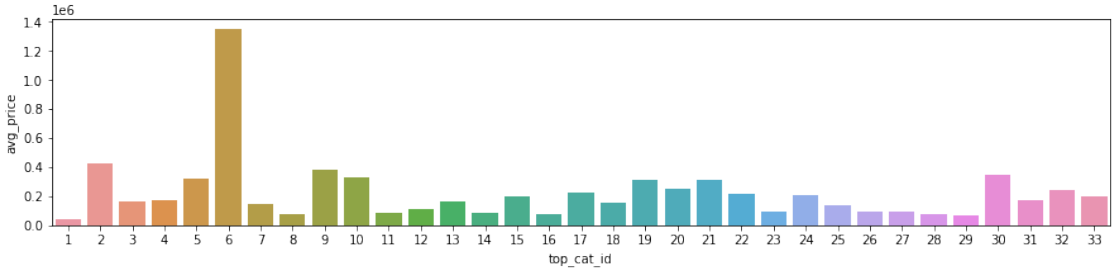
[9]:
```
plt.figure(figsize=(15,3))
sns.barplot(x='top_cat_id',y='num_orders',data=top_cats)
```

[9]: <AxesSubplot:xlabel='top_cat_id', ylabel='num_orders'>



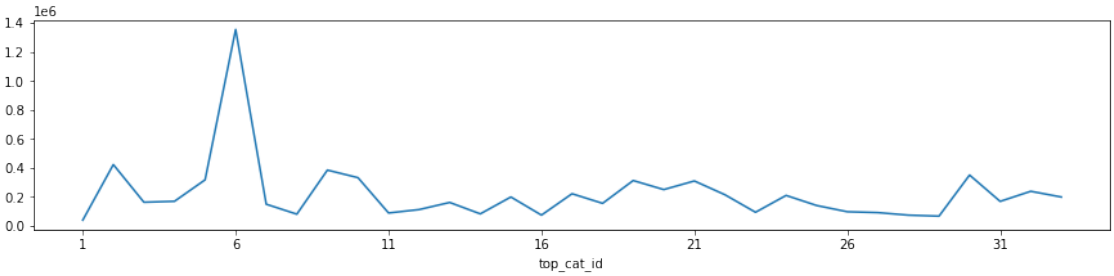
[10]:
```
plt.figure(figsize=(15,3))
sns.barplot(x='top_cat_id',y='avg_price',data=top_cats)
#baskets.groupby(by="top_cat_id").mean().spent.plot()
```

[10]: <AxesSubplot:xlabel='top_cat_id', ylabel='avg_price'>



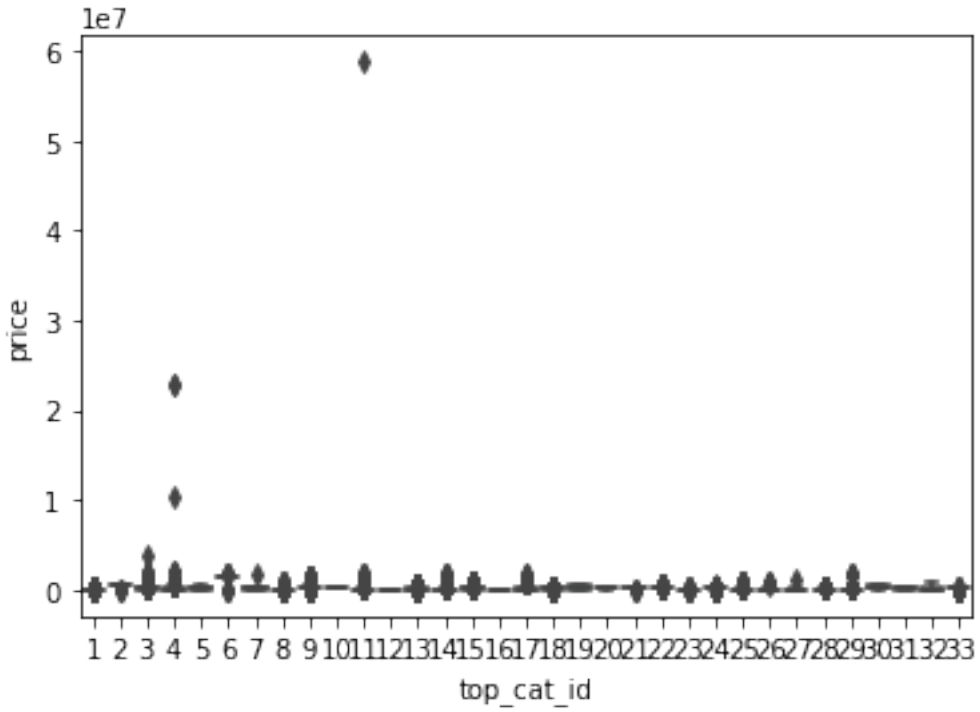
[11]:
```
plt.figure(figsize=(15,3))
baskets.groupby(by="top_cat_id").mean().price.plot()
```

[11]: <AxesSubplot:xlabel='top_cat_id'>



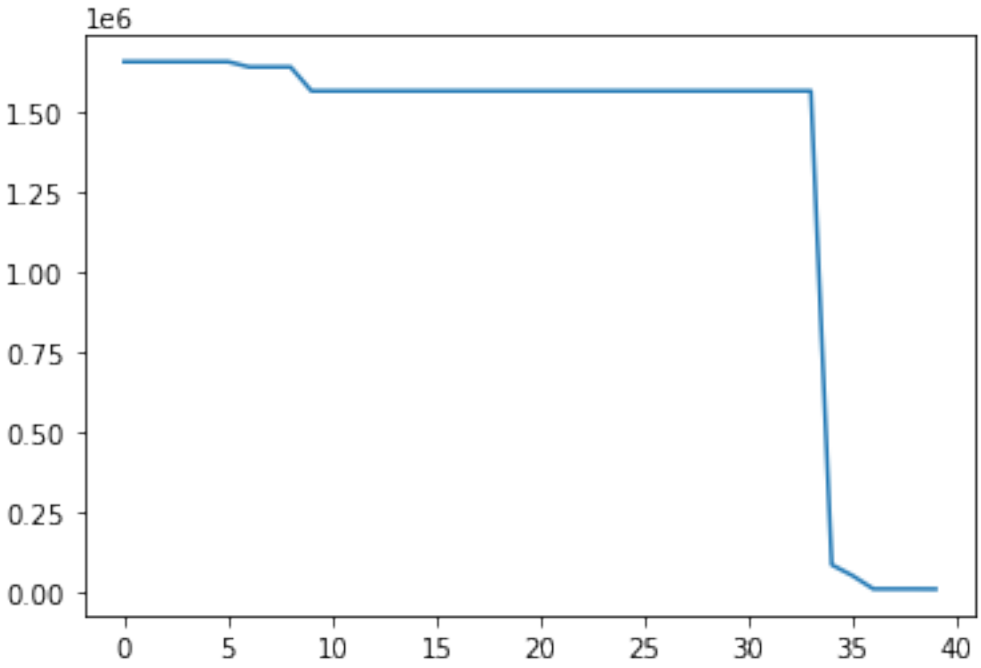
[12]:
```
sns.boxplot(x='top_cat_id', y='price', data=baskets)
```

[12]: <AxesSubplot:xlabel='top_cat_id', ylabel='price'>



[13]:
```
baskets[baskets['top_cat_id']==6].price.sort_values(ascending=False).
↪reset_index(drop = True).plot()
```

[13]: <AxesSubplot:>



[14]:
```
print(1.2e+5)
```

```
120000.0
```

[15]:
```
baskets.price.describe()
```

[15]:
```
count    3.364610e+05
mean     1.378965e+05
std      1.744712e+05
min      4.375000e-02
25%      4.600000e+04
50%      1.070000e+05
75%      1.845000e+05
max      5.875000e+07
Name: price, dtype: float64
```

[16]:
```
pd.options.display.max_rows = 100
from pprint import pprint
pprint(baskets.price.sort_values(ascending=False).reset_index(drop=True).
  ↪head(100),compact=True)
```

```
0     58750000.0
1     23010000.0
2     23010000.0
3     10550000.0
4      3825000.0
5      2222000.0
6      2175000.0
7      2175000.0
8      2125000.0
9      1925000.0
10     1925000.0
11     1925000.0
12     1925000.0
13     1925000.0
14     1925000.0
15     1925000.0
16     1925000.0
17     1890000.0
18     1870000.0
19     1849000.0
20     1849000.0
21     1849000.0
22     1849000.0
23     1849000.0
```

24	1849000.0
25	1849000.0
26	1849000.0
27	1849000.0
28	1849000.0
29	1849000.0
30	1849000.0
31	1849000.0
32	1844000.0
33	1773000.0
34	1740000.0
35	1725000.0
36	1725000.0
37	1725000.0
38	1725000.0
39	1685000.0
40	1685000.0
41	1656500.0
42	1656500.0
43	1656500.0
44	1656500.0
45	1656500.0
46	1656500.0
47	1640000.0
48	1640000.0
49	1640000.0
50	1565000.0
51	1565000.0
52	1565000.0
53	1565000.0
54	1565000.0
55	1565000.0
56	1565000.0
57	1565000.0
58	1565000.0
59	1565000.0
60	1565000.0
61	1565000.0
62	1565000.0
63	1565000.0
64	1565000.0
65	1565000.0
66	1565000.0
67	1565000.0
68	1565000.0
69	1565000.0
70	1565000.0
71	1565000.0

```
72     1565000.0
73     1565000.0
74     1565000.0
75     1460000.0
76     1460000.0
77     1460000.0
78     1460000.0
79     1460000.0
80     1460000.0
81     1460000.0
82     1460000.0
83     1460000.0
84     1460000.0
85     1460000.0
86     1460000.0
87     1460000.0
88     1460000.0
89     1425000.0
90     1425000.0
91     1425000.0
92     1425000.0
93     1387500.0
94     1387500.0
95     1375000.0
96     1375000.0
97     1375000.0
98     1375000.0
99     1375000.0
Name: price, dtype: float64
```

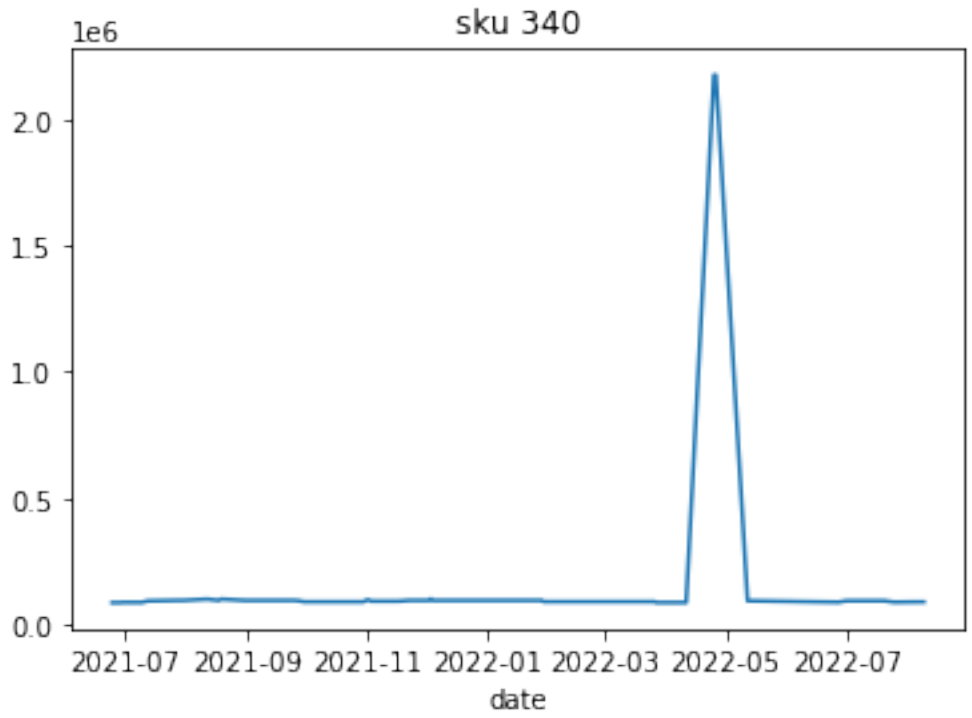
[17]:
```
pd.options.display.max_rows = 100
baskets.sort_values(by = ['price'], ascending=False).loc[:,['sku_id','price']].
  ↪head(10)
```

[17]:

	sku_id	price
78536	301	58750000.0
67262	200	23010000.0
66551	200	23010000.0
310185	432	10550000.0
5392	521	3825000.0
220150	438	2222000.0
277682	340	2175000.0
278592	340	2175000.0
2417	599	2125000.0
237889	974	1925000.0

[18]:
```
baskets[baskets.sku_id == 340].groupby('date').mean().price.plot(title="sku␣
↪340")
```

[18]:
```
<AxesSubplot:title={'center':'sku 340'}, xlabel='date'>
```



[68]:
```
pd.options.display.max_rows = 100
baskets[baskets.sku_id == 340].sort_values(by='price',ascending= False).
↪head(100)
```

[68]:

	id	order_id	placed_at	merchant_id	sku_id	\
278592	275340	47005	2022-04-26 13:52:47.626	1597	340	
277682	277828	51100	2022-04-25 21:21:43.297	608	340	
121064	121221	19000	2021-12-03 19:52:04.893	354	340	
11350	11435	1006	2021-08-12 11:37:13.699	305	340	
13163	13290	1211	2021-08-19 20:22:18.182	472	340	
70311	70427	9093	2021-11-01 16:08:50.574	32	340	
142767	142853	23378	2021-12-21 13:57:29.266	63	340	
113395	113637	17266	2021-11-29 14:58:14.928	559	340	
115434	115620	17825	2021-11-30 15:21:30.102	156	340	
118247	118408	18469	2021-12-02 12:39:15.964	754	340	
122492	122607	19279	2021-12-04 14:47:47.833	150	340	
123962	124091	19442	2021-12-06 14:05:43.417	216	340	
126169	126644	19909	2021-12-07 16:32:18.108	410	340	

136267	136420	22007	2021-12-15	14:57:43.556	13	340
137744	137868	22361	2021-12-16	15:15:54.403	1058	340
141058	141139	23010	2021-12-20	14:06:39.780	216	340
150388	150586	24898	2021-12-28	11:04:55.691	559	340
149314	149504	24844	2021-12-27	15:14:04.770	563	340
154093	154272	25797	2021-12-31	11:34:54.197	559	340
158417	158623	26511	2022-01-05	08:33:32.728	29	340
169467	169710	29212	2022-01-14	09:55:19.819	1615	340
173741	173897	30070	2022-01-18	07:05:57.745	29	340
174721	174901	30320	2022-01-18	12:37:29.841	664	340
176297	176572	30677	2022-01-19	16:56:35.198	754	340
179703	179967	29442	2022-01-22	07:42:55.071	864	340
182396	182538	31912	2022-01-24	12:34:09.609	864	340
184439	184668	32377	2022-01-26	11:11:25.013	649	340
186309	186613	32926	2022-01-27	17:17:47.984	559	340
186722	186935	33005	2022-01-28	09:01:17.781	649	340
102827	103011	15344	2021-11-22	15:08:54.644	216	340
126275	126493	15013	2021-12-07	18:49:57.471	965	340
16602	16704	1566	2021-08-31	14:04:26.386	592	340
25663	25635	2551	2021-09-25	16:55:22.024	710	340
7790	7832	672	2021-07-28	13:25:34.313	33	340
8407	8445	732	2021-07-31	11:51:58.008	29	340
12816	13123	1170	2021-08-18	16:46:22.525	13	340
22845	22911	2182	2021-09-19	22:35:38.292	380	340
115852	116079	17897	2021-12-01	08:12:41.711	29	340
27214	27259	2800	2021-09-27	19:22:17.574	32	340
287348	287578	52418	2022-05-12	21:44:03.610	437	340
314375	314458	57487	2022-07-01	21:00:28.528	290	340
319089	318817	58578	2022-07-11	10:23:03.111	1632	340
324300	324436	59513	2022-07-20	15:27:57.094	1984	340
323458	323593	59238	2022-07-19	07:20:55.147	29	340
322808	322945	59343	2022-07-17	19:32:39.329	836	340
5260	5322	433	2021-07-13	07:23:35.966	313	340
93447	93615	13322	2021-11-16	07:56:47.870	965	340
94654	94762	13666	2021-11-16	15:03:49.279	714	340
79449	79553	10900	2021-11-06	14:53:15.481	1058	340
81458	81628	11206	2021-11-08	13:09:51.569	754	340
73253	73363	9745	2021-11-03	10:44:46.786	864	340
86710	86842	12161	2021-11-11	10:43:46.987	180	340
89735	91970	12678	2021-11-12	16:08:56.448	13	340
77901	78006	10650	2021-11-05	17:00:53.015	150	340
73148	73255	9717	2021-11-03	10:18:49.036	150	340
96042	96128	13971	2021-11-17	12:51:29.788	965	340
204479	204748	36653	2022-02-09	13:19:18.214	216	340
256721	256879	47714	2022-03-26	13:16:59.537	180	340
250133	250389	46083	2022-03-21	08:07:13.735	29	340
246009	246175	44971	2022-03-16	13:35:21.886	29	340

```
239570  239537    43880  2022-03-08 16:16:03.498         380    340
238821  238947    43990  2022-03-08 07:35:30.513          29    340
229303  229496    42239  2022-02-26 10:10:25.006        1860    340
229139  229332    42185  2022-02-26 09:30:13.337        1187    340
227347  227590    41779  2022-02-25 10:20:16.798         177    340
226871  227077    41665  2022-02-25 08:18:31.756         534    340
222169  222293    40482  2022-02-22 11:32:41.624        1767    340
219839  220128    40061  2022-02-20 18:18:12.342        1781    340
217269  217479    38759  2022-02-18 13:00:14.160          13    340
215251  215442    38931  2022-02-17 10:53:50.028        1188    340
213127  213257    38430  2022-02-16 07:11:13.036          29    340
189452  189632    33622  2022-01-29 16:33:45.633          10    340
191583  192982    33955  2022-01-31 22:17:20.968         410    340
30055    30127     3160  2021-10-02 13:36:38.247         902    340
42670    43791     4714  2021-10-14 07:51:03.284         864    340
28419    28461     2947  2021-09-30 14:55:20.031         410    340
67806    67879     8907  2021-10-30 12:06:29.182         901    340
62843    62946     8053  2021-10-27 13:29:58.470         559    340
35912    36065     4074  2021-10-07 21:27:35.899         864    340
39422    39502     4555  2021-10-11 15:21:56.257         649    340
40093    40249     4671  2021-10-12 11:08:13.627         156    340
44660    44861     5373  2021-10-15 11:33:54.845          13    340
45269    45401     5453  2021-10-15 15:35:34.299         864    340
47135    47204     5727  2021-10-16 15:37:49.000         156    340
49774    49922     5173  2021-10-19 08:52:28.128         387    340
53811    53943     6728  2021-10-22 08:54:11.312         965    340
60373    60353     7676  2021-10-26 09:56:42.005         216    340
35640    35781     4019  2021-10-07 15:47:41.923         432    340
332302  332359    61300  2022-08-06 19:48:25.468         608    340
333407  333545    60880  2022-08-09 09:10:11.115        1603    340
312174  312385    57259  2022-06-27 15:24:05.211         497    340
326521  326687    58989  2022-07-25 13:24:41.172         559    340
5036      5089      406  2021-07-10 11:50:46.198          26    340
3464      3485      252  2021-06-30 16:40:59.127         109    340
269238  269349    49557  2022-04-11 14:11:54.938         864    340
265511  262777    48258  2022-04-04 08:22:13.540         608    340
262253  262408    48299  2022-03-31 11:25:27.282         864    340
258059  258206    48028  2022-03-27 22:43:24.490         636    340
3046      3046      222  2021-06-29 12:55:02.476          90    340
3143      3143      228  2021-06-29 13:41:30.625         280    340

       top_cat_id sub_cat_id  qty      price                datetime  \
278592          4         28    1  2175000.0 2022-04-26 13:52:47.626
277682          4         28    1  2175000.0 2022-04-25 21:21:43.297
121064          4         28    3   100000.0 2021-12-03 19:52:04.893
11350           4         28    1    99000.0 2021-08-12 11:37:13.699
13163           4         28    1    99000.0 2021-08-19 20:22:18.182
```

70311	4	28	5	97000.0	2021-11-01	16:08:50.574
142767	4	28	25	94000.0	2021-12-21	13:57:29.266
113395	4	28	25	94000.0	2021-11-29	14:58:14.928
115434	4	28	25	94000.0	2021-11-30	15:21:30.102
118247	4	28	25	94000.0	2021-12-02	12:39:15.964
122492	4	28	20	94000.0	2021-12-04	14:47:47.833
123962	4	28	25	94000.0	2021-12-06	14:05:43.417
126169	4	28	25	94000.0	2021-12-07	16:32:18.108
136267	4	28	100	94000.0	2021-12-15	14:57:43.556
137744	4	28	25	94000.0	2021-12-16	15:15:54.403
141058	4	28	25	94000.0	2021-12-20	14:06:39.780
150388	4	28	25	94000.0	2021-12-28	11:04:55.691
149314	4	28	25	94000.0	2021-12-27	15:14:04.770
154093	4	28	25	94000.0	2021-12-31	11:34:54.197
158417	4	28	1	94000.0	2022-01-05	08:33:32.728
169467	4	28	25	94000.0	2022-01-14	09:55:19.819
173741	4	28	25	94000.0	2022-01-18	07:05:57.745
174721	4	28	25	94000.0	2022-01-18	12:37:29.841
176297	4	28	25	94000.0	2022-01-19	16:56:35.198
179703	4	28	25	94000.0	2022-01-22	07:42:55.071
182396	4	28	25	94000.0	2022-01-24	12:34:09.609
184439	4	28	25	94000.0	2022-01-26	11:11:25.013
186309	4	28	25	94000.0	2022-01-27	17:17:47.984
186722	4	28	25	94000.0	2022-01-28	09:01:17.781
102827	4	28	25	94000.0	2021-11-22	15:08:54.644
126275	4	28	25	94000.0	2021-12-07	18:49:57.471
16602	4	28	25	94000.0	2021-08-31	14:04:26.386
25663	4	28	25	94000.0	2021-09-25	16:55:22.024
7790	4	28	25	94000.0	2021-07-28	13:25:34.313
8407	4	28	25	94000.0	2021-07-31	11:51:58.008
12816	4	28	25	94000.0	2021-08-18	16:46:22.525
22845	4	28	25	94000.0	2021-09-19	22:35:38.292
115852	4	28	25	94000.0	2021-12-01	08:12:41.711
27214	4	28	5	93000.0	2021-09-27	19:22:17.574
287348	4	28	2	92500.0	2022-05-12	21:44:03.610
314375	4	28	1	92500.0	2022-07-01	21:00:28.528
319089	4	28	1	92500.0	2022-07-11	10:23:03.111
324300	4	28	25	92500.0	2022-07-20	15:27:57.094
323458	4	28	25	92500.0	2022-07-19	07:20:55.147
322808	4	28	2	92500.0	2022-07-17	19:32:39.329
5260	4	28	2	92000.0	2021-07-13	07:23:35.966
93447	4	28	25	91000.0	2021-11-16	07:56:47.870
94654	4	28	25	91000.0	2021-11-16	15:03:49.279
79449	4	28	25	91000.0	2021-11-06	14:53:15.481
81458	4	28	25	91000.0	2021-11-08	13:09:51.569
73253	4	28	50	91000.0	2021-11-03	10:44:46.786
86710	4	28	25	91000.0	2021-11-11	10:43:46.987

89735	4	28	75	91000.0	2021-11-12 16:08:56.448
77901	4	28	25	91000.0	2021-11-05 17:00:53.015
73148	4	28	25	91000.0	2021-11-03 10:18:49.036
96042	4	28	25	91000.0	2021-11-17 12:51:29.788
204479	4	28	25	89000.0	2022-02-09 13:19:18.214
256721	4	28	25	89000.0	2022-03-26 13:16:59.537
250133	4	28	25	89000.0	2022-03-21 08:07:13.735
246009	4	28	25	89000.0	2022-03-16 13:35:21.886
239570	4	28	25	89000.0	2022-03-08 16:16:03.498
238821	4	28	25	89000.0	2022-03-08 07:35:30.513
229303	4	28	25	89000.0	2022-02-26 10:10:25.006
229139	4	28	25	89000.0	2022-02-26 09:30:13.337
227347	4	28	25	89000.0	2022-02-25 10:20:16.798
226871	4	28	25	89000.0	2022-02-25 08:18:31.756
222169	4	28	25	89000.0	2022-02-22 11:32:41.624
219839	4	28	10	89000.0	2022-02-20 18:18:12.342
217269	4	28	50	89000.0	2022-02-18 13:00:14.160
215251	4	28	25	89000.0	2022-02-17 10:53:50.028
213127	4	28	25	89000.0	2022-02-16 07:11:13.036
189452	4	28	25	89000.0	2022-01-29 16:33:45.633
191583	4	28	50	89000.0	2022-01-31 22:17:20.968
30055	4	28	25	88500.0	2021-10-02 13:36:38.247
42670	4	28	25	88500.0	2021-10-14 07:51:03.284
28419	4	28	25	88500.0	2021-09-30 14:55:20.031
67806	4	28	25	88500.0	2021-10-30 12:06:29.182
62843	4	28	25	88500.0	2021-10-27 13:29:58.470
35912	4	28	25	88500.0	2021-10-07 21:27:35.899
39422	4	28	25	88500.0	2021-10-11 15:21:56.257
40093	4	28	25	88500.0	2021-10-12 11:08:13.627
44660	4	28	50	88500.0	2021-10-15 11:33:54.845
45269	4	28	25	88500.0	2021-10-15 15:35:34.299
47135	4	28	25	88500.0	2021-10-16 15:37:49.000
49774	4	28	25	88500.0	2021-10-19 08:52:28.128
53811	4	28	25	88500.0	2021-10-22 08:54:11.312
60373	4	28	25	88500.0	2021-10-26 09:56:42.005
35640	4	28	25	88500.0	2021-10-07 15:47:41.923
332302	4	28	25	88000.0	2022-08-06 19:48:25.468
333407	4	28	1	88000.0	2022-08-09 09:10:11.115
312174	4	28	50	87000.0	2022-06-27 15:24:05.211
326521	4	28	25	87000.0	2022-07-25 13:24:41.172
5036	4	28	25	86000.0	2021-07-10 11:50:46.198
3464	4	28	62	86000.0	2021-06-30 16:40:59.127
269238	4	28	50	85500.0	2022-04-11 14:11:54.938
265511	4	28	25	85500.0	2022-04-04 08:22:13.540
262253	4	28	25	85500.0	2022-03-31 11:25:27.282
258059	4	28	2	85500.0	2022-03-27 22:43:24.490
3046	4	28	25	85000.0	2021-06-29 12:55:02.476

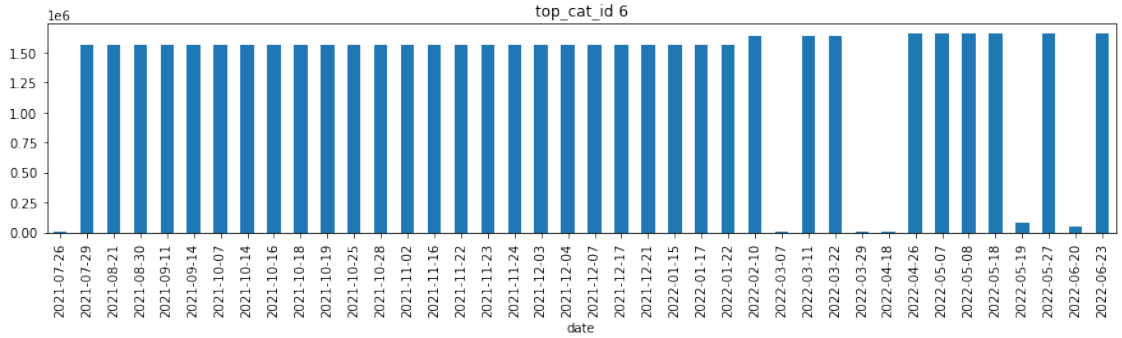
```
3143           4         28   50    85000.0 2021-06-29 13:41:30.625

             date  year  month  day  hour  weekday      spent
278592 2022-04-26  2022      4   26    13        1  2175000.0
277682 2022-04-25  2022      4   25    21        0  2175000.0
121064 2021-12-03  2021     12    3    19        4   300000.0
11350  2021-08-12  2021      8   12    11        3    99000.0
13163  2021-08-19  2021      8   19    20        3    99000.0
70311  2021-11-01  2021     11    1    16        0   485000.0
142767 2021-12-21  2021     12   21    13        1  2350000.0
113395 2021-11-29  2021     11   29    14        0  2350000.0
115434 2021-11-30  2021     11   30    15        1  2350000.0
118247 2021-12-02  2021     12    2    12        3  2350000.0
122492 2021-12-04  2021     12    4    14        5  1880000.0
123962 2021-12-06  2021     12    6    14        0  2350000.0
126169 2021-12-07  2021     12    7    16        1  2350000.0
136267 2021-12-15  2021     12   15    14        2  9400000.0
137744 2021-12-16  2021     12   16    15        3  2350000.0
141058 2021-12-20  2021     12   20    14        0  2350000.0
150388 2021-12-28  2021     12   28    11        1  2350000.0
149314 2021-12-27  2021     12   27    15        0  2350000.0
154093 2021-12-31  2021     12   31    11        4  2350000.0
158417 2022-01-05  2022      1    5     8        2    94000.0
169467 2022-01-14  2022      1   14     9        4  2350000.0
173741 2022-01-18  2022      1   18     7        1  2350000.0
174721 2022-01-18  2022      1   18    12        1  2350000.0
176297 2022-01-19  2022      1   19    16        2  2350000.0
179703 2022-01-22  2022      1   22     7        5  2350000.0
182396 2022-01-24  2022      1   24    12        0  2350000.0
184439 2022-01-26  2022      1   26    11        2  2350000.0
186309 2022-01-27  2022      1   27    17        3  2350000.0
186722 2022-01-28  2022      1   28     9        4  2350000.0
102827 2021-11-22  2021     11   22    15        0  2350000.0
126275 2021-12-07  2021     12    7    18        1  2350000.0
16602  2021-08-31  2021      8   31    14        1  2350000.0
25663  2021-09-25  2021      9   25    16        5  2350000.0
7790   2021-07-28  2021      7   28    13        2  2350000.0
8407   2021-07-31  2021      7   31    11        5  2350000.0
12816  2021-08-18  2021      8   18    16        2  2350000.0
22845  2021-09-19  2021      9   19    22        6  2350000.0
115852 2021-12-01  2021     12    1     8        2  2350000.0
27214  2021-09-27  2021      9   27    19        0   465000.0
287348 2022-05-12  2022      5   12    21        3   185000.0
314375 2022-07-01  2022      7    1    21        4    92500.0
319089 2022-07-11  2022      7   11    10        0    92500.0
324300 2022-07-20  2022      7   20    15        2  2312500.0
323458 2022-07-19  2022      7   19     7        1  2312500.0
```

322808	2022-07-17	2022	7	17	19	6	185000.0
5260	2021-07-13	2021	7	13	7	1	184000.0
93447	2021-11-16	2021	11	16	7	1	2275000.0
94654	2021-11-16	2021	11	16	15	1	2275000.0
79449	2021-11-06	2021	11	6	14	5	2275000.0
81458	2021-11-08	2021	11	8	13	0	2275000.0
73253	2021-11-03	2021	11	3	10	2	4550000.0
86710	2021-11-11	2021	11	11	10	3	2275000.0
89735	2021-11-12	2021	11	12	16	4	6825000.0
77901	2021-11-05	2021	11	5	17	4	2275000.0
73148	2021-11-03	2021	11	3	10	2	2275000.0
96042	2021-11-17	2021	11	17	12	2	2275000.0
204479	2022-02-09	2022	2	9	13	2	2225000.0
256721	2022-03-26	2022	3	26	13	5	2225000.0
250133	2022-03-21	2022	3	21	8	0	2225000.0
246009	2022-03-16	2022	3	16	13	2	2225000.0
239570	2022-03-08	2022	3	8	16	1	2225000.0
238821	2022-03-08	2022	3	8	7	1	2225000.0
229303	2022-02-26	2022	2	26	10	5	2225000.0
229139	2022-02-26	2022	2	26	9	5	2225000.0
227347	2022-02-25	2022	2	25	10	4	2225000.0
226871	2022-02-25	2022	2	25	8	4	2225000.0
222169	2022-02-22	2022	2	22	11	1	2225000.0
219839	2022-02-20	2022	2	20	18	6	890000.0
217269	2022-02-18	2022	2	18	13	4	4450000.0
215251	2022-02-17	2022	2	17	10	3	2225000.0
213127	2022-02-16	2022	2	16	7	2	2225000.0
189452	2022-01-29	2022	1	29	16	5	2225000.0
191583	2022-01-31	2022	1	31	22	0	4450000.0
30055	2021-10-02	2021	10	2	13	5	2212500.0
42670	2021-10-14	2021	10	14	7	3	2212500.0
28419	2021-09-30	2021	9	30	14	3	2212500.0
67806	2021-10-30	2021	10	30	12	5	2212500.0
62843	2021-10-27	2021	10	27	13	2	2212500.0
35912	2021-10-07	2021	10	7	21	3	2212500.0
39422	2021-10-11	2021	10	11	15	0	2212500.0
40093	2021-10-12	2021	10	12	11	1	2212500.0
44660	2021-10-15	2021	10	15	11	4	4425000.0
45269	2021-10-15	2021	10	15	15	4	2212500.0
47135	2021-10-16	2021	10	16	15	5	2212500.0
49774	2021-10-19	2021	10	19	8	1	2212500.0
53811	2021-10-22	2021	10	22	8	4	2212500.0
60373	2021-10-26	2021	10	26	9	1	2212500.0
35640	2021-10-07	2021	10	7	15	3	2212500.0
332302	2022-08-06	2022	8	6	19	5	2200000.0
333407	2022-08-09	2022	8	9	9	1	88000.0
312174	2022-06-27	2022	6	27	15	0	4350000.0

326521	2022-07-25	2022	7	25	13	0	2175000.0
5036	2021-07-10	2021	7	10	11	5	2150000.0
3464	2021-06-30	2021	6	30	16	2	5332000.0
269238	2022-04-11	2022	4	11	14	0	4275000.0
265511	2022-04-04	2022	4	4	8	0	2137500.0
262253	2022-03-31	2022	3	31	11	3	2137500.0
258059	2022-03-27	2022	3	27	22	6	171000.0
3046	2021-06-29	2021	6	29	12	1	2125000.0
3143	2021-06-29	2021	6	29	13	1	4250000.0

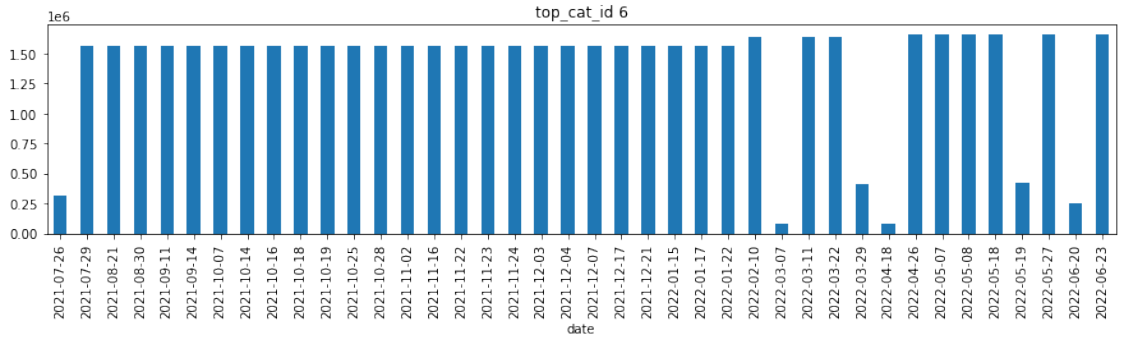
[19]:
```
plt.figure(figsize=(15,3))
baskets[baskets.top_cat_id == 6].groupby('date').mean().price.
 ↪plot(title="top_cat_id 6",kind='bar')
```

[19]: <AxesSubplot:title={'center':'top_cat_id 6'}, xlabel='date'>



[20]:
```
plt.figure(figsize=(15,3))
baskets[baskets.top_cat_id == 6].groupby('date').sum().spent.
 ↪plot(title="top_cat_id 6",kind='bar')
```

[20]: <AxesSubplot:title={'center':'top_cat_id 6'}, xlabel='date'>



[21]:
```
baskets[baskets.top_cat_id == 6.0].sort_values(by="merchant_id")
```

[21]:

	id	order_id	placed_at	merchant_id	sku_id	top_cat_id	sub_cat_id	qty	price	datetime
259394	259603	47527	2022-03-29 09:26:08.555	33	822	6	33	50	8300.0	2022-03-29 09:26:08.555
7304	7367	621	2021-07-26 13:05:45.662	35	822	6	33	40	7950.0	2021-07-26 13:05:45.662
295747	295853	54049	2022-05-27 17:34:50.006	122	822					
72515	72669	9649	2021-11-02 17:58:27.709	148	822					
59434	59428	7554	2021-10-25 15:10:53.414	148	822					
205859	206093	36999	2022-02-10 11:35:19.843	261	822					
173336	173450	29993	2022-01-17 13:46:09.700	375	822					
103623	103790	15474	2021-11-23 09:31:53.009	375	822					
103027	103178	15370	2021-11-22 15:52:03.238	375	822					
93710	93844	13293	2021-11-16 09:47:12.814	375	822					
283825	283346	51336	2022-05-07 10:12:00.341	375	822					
290630	290723	53168	2022-05-18 10:51:06.353	375	822					
48842	48856	5775	2021-10-18 12:59:05.430	375	822					
35300	35422	3957	2021-10-07 13:04:30.821	375	822					
20760	20805	2018	2021-09-14 09:21:24.078	375	822					
19953	20037	1928	2021-09-11 11:54:45.548	375	822					
16120	16167	1521	2021-08-30 12:15:14.460	375	822					
13557	13604	1213	2021-08-21 13:11:01.255	375	822					
7967	7998	678	2021-07-29 13:03:33.043	375	822					
170990	171180	29509	2022-01-15 09:58:30.842	375	822					
138527	138704	22483	2021-12-17 16:40:41.124	375	822					
46984	47043	5713	2021-10-16 15:03:34.985	450	822					
122875	121876	18900	2021-12-04 16:58:07.795	525	822					
284291	284466	51865	2022-05-08 14:39:56.370	552	822					
238603	239209	43932	2022-03-07 18:23:47.222	582	822					
43435	43476	5188	2021-10-14 14:11:29.179	592	822					
65175	65239	8453	2021-10-28 17:18:58.524	702	822					
106046	106225	16033	2021-11-24 13:44:02.823	751	822					
124787	124976	19629	2021-12-07 09:56:04.340	766	822					
141625	141735	23172	2021-12-21 09:07:18.137	983	822					
180018	180275	31432	2022-01-22 10:01:50.789	1140	822					
120677	120775	18905	2021-12-03 15:15:52.838	1140	822					
241808	242054	44482	2022-03-11 11:53:00.214	1140	822					
278721	278876	51266	2022-04-26 14:57:52.186	1140	822					
50205	50349	6179	2021-10-19 10:48:46.200	1140	822					
291311	291435	53266	2022-05-19 12:40:54.412	1549	822					
308040	308670	56433	2022-06-20 09:41:25.135	1654	822					
252287	252449	44426	2022-03-22 23:09:00.915	1817	822					
272434	273142	46191	2022-04-18 12:32:55.570	1904	822					
310209	310342	56816	2022-06-23 15:58:24.448	2053	822					

295747	6	33	1	1656500.0	2022-05-27	17:34:50.006
72515	6	33	1	1565000.0	2021-11-02	17:58:27.709
59434	6	33	1	1565000.0	2021-10-25	15:10:53.414
205859	6	33	1	1640000.0	2022-02-10	11:35:19.843
173336	6	33	1	1565000.0	2022-01-17	13:46:09.700
103623	6	33	1	1565000.0	2021-11-23	09:31:53.009
103027	6	33	1	1565000.0	2021-11-22	15:52:03.238
93710	6	33	1	1565000.0	2021-11-16	09:47:12.814
283825	6	33	1	1656500.0	2022-05-07	10:12:00.341
290630	6	33	1	1656500.0	2022-05-18	10:51:06.353
48842	6	33	1	1565000.0	2021-10-18	12:59:05.430
35300	6	33	1	1565000.0	2021-10-07	13:04:30.821
20760	6	33	1	1565000.0	2021-09-14	09:21:24.078
19953	6	33	1	1565000.0	2021-09-11	11:54:45.548
16120	6	33	1	1565000.0	2021-08-30	12:15:14.460
13557	6	33	1	1565000.0	2021-08-21	13:11:01.255
7967	6	33	1	1565000.0	2021-07-29	13:03:33.043
170990	6	33	1	1565000.0	2022-01-15	09:58:30.842
138527	6	33	1	1565000.0	2021-12-17	16:40:41.124
46984	6	33	1	1565000.0	2021-10-16	15:03:34.985
122875	6	33	1	1565000.0	2021-12-04	16:58:07.795
284291	6	33	1	1656500.0	2022-05-08	14:39:56.370
238603	6	33	10	8300.0	2022-03-07	18:23:47.222
43435	6	33	1	1565000.0	2021-10-14	14:11:29.179
65175	6	33	1	1565000.0	2021-10-28	17:18:58.524
106046	6	33	1	1565000.0	2021-11-24	13:44:02.823
124787	6	33	1	1565000.0	2021-12-07	09:56:04.340
141625	6	33	1	1565000.0	2021-12-21	09:07:18.137
180018	6	33	1	1565000.0	2022-01-22	10:01:50.789
120677	6	33	1	1565000.0	2021-12-03	15:15:52.838
241808	6	33	1	1640000.0	2022-03-11	11:53:00.214
278721	6	33	1	1656500.0	2022-04-26	14:57:52.186
50205	6	33	1	1565000.0	2021-10-19	10:48:46.200
291311	6	33	5	84000.0	2022-05-19	12:40:54.412
308040	6	33	5	50000.0	2022-06-20	09:41:25.135
252287	6	33	1	1640000.0	2022-03-22	23:09:00.915
272434	6	33	10	8400.0	2022-04-18	12:32:55.570
310209	6	33	1	1656500.0	2022-06-23	15:58:24.448

	date	year	month	day	hour	weekday	spent
259394	2022-03-29	2022	3	29	9	1	415000.0
7304	2021-07-26	2021	7	26	13	0	318000.0
295747	2022-05-27	2022	5	27	17	4	1656500.0
72515	2021-11-02	2021	11	2	17	1	1565000.0
59434	2021-10-25	2021	10	25	15	0	1565000.0
205859	2022-02-10	2022	2	10	11	3	1640000.0
173336	2022-01-17	2022	1	17	13	0	1565000.0

```
103623  2021-11-23  2021     11   23     9        1  1565000.0
103027  2021-11-22  2021     11   22    15        0  1565000.0
93710   2021-11-16  2021     11   16     9        1  1565000.0
283825  2022-05-07  2022      5    7    10        5  1656500.0
290630  2022-05-18  2022      5   18    10        2  1656500.0
48842   2021-10-18  2021     10   18    12        0  1565000.0
35300   2021-10-07  2021     10    7    13        3  1565000.0
20760   2021-09-14  2021      9   14     9        1  1565000.0
19953   2021-09-11  2021      9   11    11        5  1565000.0
16120   2021-08-30  2021      8   30    12        0  1565000.0
13557   2021-08-21  2021      8   21    13        5  1565000.0
7967    2021-07-29  2021      7   29    13        3  1565000.0
170990  2022-01-15  2022      1   15     9        5  1565000.0
138527  2021-12-17  2021     12   17    16        4  1565000.0
46984   2021-10-16  2021     10   16    15        5  1565000.0
122875  2021-12-04  2021     12    4    16        5  1565000.0
284291  2022-05-08  2022      5    8    14        6  1656500.0
238603  2022-03-07  2022      3    7    18        0    83000.0
43435   2021-10-14  2021     10   14    14        3  1565000.0
65175   2021-10-28  2021     10   28    17        3  1565000.0
106046  2021-11-24  2021     11   24    13        2  1565000.0
124787  2021-12-07  2021     12    7     9        1  1565000.0
141625  2021-12-21  2021     12   21     9        1  1565000.0
180018  2022-01-22  2022      1   22    10        5  1565000.0
120677  2021-12-03  2021     12    3    15        4  1565000.0
241808  2022-03-11  2022      3   11    11        4  1640000.0
278721  2022-04-26  2022      4   26    14        1  1656500.0
50205   2021-10-19  2021     10   19    10        1  1565000.0
291311  2022-05-19  2022      5   19    12        3   420000.0
308040  2022-06-20  2022      6   20     9        0   250000.0
252287  2022-03-22  2022      3   22    23        1  1640000.0
272434  2022-04-18  2022      4   18    12        0    84000.0
310209  2022-06-23  2022      6   23    15        3  1656500.0
```

[22]:
```
baskets10.sort_values(by = ['price'], ascending=True).loc[:,['sku_id','price']]
```

[22]:
```
       sku_id      price
0         341        0.0
807       810        0.0
808       811        0.0
809       607        0.0
810      1229        0.0
…         …          …
22218    1463  1925000.0
29189    1463  1925000.0
29188    1461  1925000.0
26619    1463  1925000.0
```

```
25568    774  2175000.0

[29298 rows x 2 columns]
```

[23]:
```
baskets250.price.sort_values(ascending=False).head(10)
```

[23]:
```
2130      3825000.0
134738    2175000.0
763       2125000.0
142500    1925000.0
142499    1925000.0
168446    1925000.0
112471    1925000.0
109325    1925000.0
164746    1925000.0
164745    1925000.0
Name: price, dtype: float64
```

[24]:
```
baskets.price.sort_values(ascending=False).head()
```

[24]:
```
78536     58750000.0
67262     23010000.0
66551     23010000.0
310185    10550000.0
5392       3825000.0
Name: price, dtype: float64
```

[25]:
```
baskets[baskets['price']==58750000.0]
```

[25]:
```
          id order_id               placed_at merchant_id sku_id top_cat_id  \
78536  78641    10750  2021-11-06 09:53:53.181        1103    301          11

      sub_cat_id  qty       price                datetime        date  year  \
78536         27    1  58750000.0 2021-11-06 09:53:53.181  2021-11-06  2021

       month  day  hour  weekday       spent
78536     11    6     9        5  58750000.0
```

[26]:
```
baskets10[baskets10['price']==2175000.0]
```

[26]:
```
           id order_id               placed_at merchant_id sku_id top_cat_id  \
25568  285542    47838  2022-04-26 13:52:47.626        2173    774           4

      sub_cat_id  qty      price                datetime        date  year  \
25568         31    1  2175000.0 2022-04-26 13:52:47.626  2022-04-26  2022

       month  day  hour  weekday      spent
```

```
25568      4   26    13        1  2175000.0
```

[27]:
```
baskets[baskets['sku_id'] == 301]
```

[27]:
```
            id order_id               placed_at merchant_id sku_id  \
917        914       83  2021-06-17 10:27:52.447         302    301
1451      1784      129  2021-06-24 12:43:03.065          22    301
1515      1514      135  2021-06-24 23:31:59.247         114    301
1529      1528      136  2021-06-24 23:58:49.239         114    301
1571      1570      137  2021-06-25 00:01:29.306         109    301
…          …       …                        …          …     …
335546  335629    61825  2022-08-13 23:23:58.111         665    301
336143  336225    62017  2022-08-15 14:37:01.641         146    301
336211  336249    61920  2022-08-15 15:15:51.713         441    301
336255  336310    61998  2022-08-15 16:51:02.716        1770    301
336389  336419    61858  2022-08-15 20:51:09.517        1842    301

       top_cat_id sub_cat_id  qty    price                datetime  \
917            11         27   50  23250.0 2021-06-17 10:27:52.447
1451           11         27   50  22000.0 2021-06-24 12:43:03.065
1515           11         27   10  22000.0 2021-06-24 23:31:59.247
1529           11         27   50  24500.0 2021-06-24 23:58:49.239
1571           11         27   50  24500.0 2021-06-25 00:01:29.306
…           …         …  …       …                       …
335546         11         27   10  25500.0 2022-08-13 23:23:58.111
336143         11         27   50  25500.0 2022-08-15 14:37:01.641
336211         11         27  150  25500.0 2022-08-15 15:15:51.713
336255         11         27   50  25500.0 2022-08-15 16:51:02.716
336389         11         27   50  25500.0 2022-08-15 20:51:09.517

              date  year  month  day  hour  weekday      spent
917     2021-06-17  2021      6   17    10        3  1162500.0
1451    2021-06-24  2021      6   24    12        3  1100000.0
1515    2021-06-24  2021      6   24    23        3   220000.0
1529    2021-06-24  2021      6   24    23        3  1225000.0
1571    2021-06-25  2021      6   25     0        4  1225000.0
…            …   …     …  …   …      …         …
335546  2022-08-13  2022      8   13    23        5   255000.0
336143  2022-08-15  2022      8   15    14        0  1275000.0
336211  2022-08-15  2022      8   15    15        0  3825000.0
336255  2022-08-15  2022      8   15    16        0  1275000.0
336389  2022-08-15  2022      8   15    20        0  1275000.0

[552 rows x 17 columns]
```

[28]:
```
baskets10[baskets10['sku_id'] == 340]
```

[28]:
```
      id order_id               placed_at merchant_id sku_id top_cat_id  \
72  1289      298  2021-05-24 11:40:06.796         223    340         12
79  2013      344  2021-06-15 09:57:31.177         349    340         12

   sub_cat_id  qty    price                datetime        date  year  month  \
72          1    5      0.0 2021-05-24 11:40:06.796  2021-05-24  2021      5
79          1    5  18500.0 2021-06-15 09:57:31.177  2021-06-15  2021      6

    day  hour  weekday    spent
72   24    11        0      0.0
79   15     9        1  92500.0
```

[29]:
```
baskets.groupby(['sku_id']).count().order_id.sort_values(ascending=False).
↪reset_index().head(100)
```

[29]:

	sku_id	order_id
0	327	7500
1	522	6919
2	390	6409
3	438	5589
4	523	4804
5	263	4548
6	185	4463
7	521	4462
8	276	3718
9	184	3480
10	432	3432
11	200	3353
12	485	3256
13	336	3201
14	511	3191
15	202	3077
16	547	3069
17	530	2881
18	321	2783
19	337	2625
20	1300	2578
21	271	2523
22	425	2352
23	567	2328
24	269	2299
25	524	2236
26	325	2141
27	267	2027
28	258	1890
29	403	1873
30	254	1847

31	1068	1765
32	419	1756
33	589	1720
34	435	1695
35	717	1691
36	311	1647
37	979	1613
38	512	1580
39	237	1567
40	508	1535
41	351	1516
42	563	1511
43	481	1483
44	980	1436
45	355	1433
46	334	1429
47	376	1374
48	458	1371
49	257	1342
50	420	1336
51	308	1333
52	501	1331
53	264	1322
54	312	1318
55	203	1310
56	543	1274
57	253	1231
58	996	1224
59	461	1218
60	329	1204
61	360	1200
62	281	1197
63	1053	1190
64	345	1167
65	431	1166
66	840	1150
67	194	1118
68	676	1115
69	204	1115
70	179	1098
71	470	1093
72	562	1086
73	375	1067
74	366	1057
75	225	1056
76	580	1045
77	970	1026

```
78    238      1003
79    274      1003
80    477      1001
81    347       989
82    328       951
83    342       948
84    399       938
85    532       909
86    332       893
87    673       892
88    393       886
89    484       879
90    513       864
91    963       849
92    455       831
93    434       823
94    236       821
95    545       800
96    343       799
97    518       788
98    473       778
99    460       777
```

[30]:
```
from pprint import pprint
pprint(list(skus[skus["sku_id"] == 438].date), compact = True)
```

```
[datetime.date(2021, 4, 9), datetime.date(2021, 4, 15),
 datetime.date(2021, 4, 16), datetime.date(2021, 4, 17),
 datetime.date(2021, 4, 19), datetime.date(2021, 4, 20),
 datetime.date(2021, 4, 22), datetime.date(2021, 4, 30),
 datetime.date(2021, 5, 3), datetime.date(2021, 5, 5),
 datetime.date(2021, 5, 6), datetime.date(2021, 5, 8),
 datetime.date(2021, 5, 9), datetime.date(2021, 5, 10),
 datetime.date(2021, 5, 11), datetime.date(2021, 5, 17),
 datetime.date(2021, 5, 18), datetime.date(2021, 5, 19),
 datetime.date(2021, 5, 20), datetime.date(2021, 5, 21),
 datetime.date(2021, 5, 22), datetime.date(2021, 5, 24),
 datetime.date(2021, 5, 26), datetime.date(2021, 5, 27),
 datetime.date(2021, 5, 31), datetime.date(2021, 6, 2),
 datetime.date(2021, 6, 3), datetime.date(2021, 6, 4),
 datetime.date(2021, 6, 7), datetime.date(2021, 6, 8),
 datetime.date(2021, 6, 9), datetime.date(2021, 6, 10),
 datetime.date(2021, 6, 11), datetime.date(2021, 6, 14),
 datetime.date(2021, 6, 15), datetime.date(2021, 6, 16),
 datetime.date(2021, 6, 17), datetime.date(2021, 6, 18),
 datetime.date(2021, 6, 19), datetime.date(2021, 6, 20),
 datetime.date(2021, 6, 21), datetime.date(2021, 6, 22),
```

```
datetime.date(2021, 6, 23), datetime.date(2021, 6, 24),
datetime.date(2021, 6, 25), datetime.date(2021, 6, 26),
datetime.date(2021, 6, 27), datetime.date(2021, 6, 28),
datetime.date(2021, 6, 29), datetime.date(2021, 6, 30),
datetime.date(2021, 7, 1), datetime.date(2021, 7, 2),
datetime.date(2021, 7, 3), datetime.date(2021, 7, 4),
datetime.date(2021, 7, 5), datetime.date(2021, 7, 6),
datetime.date(2021, 7, 7), datetime.date(2021, 7, 8),
datetime.date(2021, 7, 9), datetime.date(2021, 7, 10),
datetime.date(2021, 7, 11), datetime.date(2021, 7, 12),
datetime.date(2021, 7, 13), datetime.date(2021, 7, 14),
datetime.date(2021, 7, 15), datetime.date(2021, 7, 16),
datetime.date(2021, 7, 17), datetime.date(2021, 7, 18),
datetime.date(2021, 7, 19), datetime.date(2021, 7, 20),
datetime.date(2021, 7, 21), datetime.date(2021, 7, 22),
datetime.date(2021, 7, 23), datetime.date(2021, 7, 24),
datetime.date(2021, 7, 25), datetime.date(2021, 7, 26),
datetime.date(2021, 7, 27), datetime.date(2021, 7, 28),
datetime.date(2021, 7, 29), datetime.date(2021, 7, 30),
datetime.date(2021, 7, 31), datetime.date(2021, 8, 1),
datetime.date(2021, 8, 2), datetime.date(2021, 8, 3),
datetime.date(2021, 8, 4), datetime.date(2021, 8, 5),
datetime.date(2021, 8, 6), datetime.date(2021, 8, 7),
datetime.date(2021, 8, 8), datetime.date(2021, 8, 9),
datetime.date(2021, 8, 10), datetime.date(2021, 8, 11),
datetime.date(2021, 8, 12), datetime.date(2021, 8, 13),
datetime.date(2021, 8, 14), datetime.date(2021, 8, 15),
datetime.date(2021, 8, 16), datetime.date(2021, 8, 17),
datetime.date(2021, 8, 18), datetime.date(2021, 8, 19),
datetime.date(2021, 8, 20), datetime.date(2021, 8, 21),
datetime.date(2021, 8, 22), datetime.date(2021, 8, 23),
datetime.date(2021, 8, 24), datetime.date(2021, 8, 25),
datetime.date(2021, 8, 26), datetime.date(2021, 8, 27),
datetime.date(2021, 8, 28), datetime.date(2021, 8, 29),
datetime.date(2021, 8, 30), datetime.date(2021, 8, 31),
datetime.date(2021, 9, 1), datetime.date(2021, 9, 2),
datetime.date(2021, 9, 3), datetime.date(2021, 9, 4),
datetime.date(2021, 9, 5), datetime.date(2021, 9, 6),
datetime.date(2021, 9, 7), datetime.date(2021, 9, 8),
datetime.date(2021, 9, 9), datetime.date(2021, 9, 10),
datetime.date(2021, 9, 11), datetime.date(2021, 9, 12),
datetime.date(2021, 9, 13), datetime.date(2021, 9, 14),
datetime.date(2021, 9, 15), datetime.date(2021, 9, 16),
datetime.date(2021, 9, 17), datetime.date(2021, 9, 18),
datetime.date(2021, 9, 19), datetime.date(2021, 9, 20),
datetime.date(2021, 9, 21), datetime.date(2021, 9, 22),
datetime.date(2021, 9, 23), datetime.date(2021, 9, 24),
datetime.date(2021, 9, 25), datetime.date(2021, 9, 26),
```

```
datetime.date(2021, 9, 27), datetime.date(2021, 9, 28),
datetime.date(2021, 9, 29), datetime.date(2021, 9, 30),
datetime.date(2021, 10, 1), datetime.date(2021, 10, 2),
datetime.date(2021, 10, 3), datetime.date(2021, 10, 4),
datetime.date(2021, 10, 5), datetime.date(2021, 10, 6),
datetime.date(2021, 10, 7), datetime.date(2021, 10, 8),
datetime.date(2021, 10, 9), datetime.date(2021, 10, 10),
datetime.date(2021, 10, 11), datetime.date(2021, 10, 12),
datetime.date(2021, 10, 13), datetime.date(2021, 10, 14),
datetime.date(2021, 10, 15), datetime.date(2021, 10, 16),
datetime.date(2021, 10, 17), datetime.date(2021, 10, 18),
datetime.date(2021, 10, 19), datetime.date(2021, 10, 20),
datetime.date(2021, 10, 21), datetime.date(2021, 10, 22),
datetime.date(2021, 10, 23), datetime.date(2021, 10, 24),
datetime.date(2021, 10, 25), datetime.date(2021, 10, 26),
datetime.date(2021, 10, 27), datetime.date(2021, 10, 28),
datetime.date(2021, 10, 29), datetime.date(2021, 10, 30),
datetime.date(2021, 10, 31), datetime.date(2021, 11, 1),
datetime.date(2021, 11, 2), datetime.date(2021, 11, 3),
datetime.date(2021, 11, 4), datetime.date(2021, 11, 5),
datetime.date(2021, 11, 6), datetime.date(2021, 11, 7),
datetime.date(2021, 11, 8), datetime.date(2021, 11, 9),
datetime.date(2021, 11, 10), datetime.date(2021, 11, 11),
datetime.date(2021, 11, 12), datetime.date(2021, 11, 13),
datetime.date(2021, 11, 14), datetime.date(2021, 11, 15),
datetime.date(2021, 11, 16), datetime.date(2021, 11, 17),
datetime.date(2021, 11, 18), datetime.date(2021, 11, 19),
datetime.date(2021, 11, 20), datetime.date(2021, 11, 21),
datetime.date(2021, 11, 22), datetime.date(2021, 11, 23),
datetime.date(2021, 11, 24), datetime.date(2021, 11, 25),
datetime.date(2021, 11, 26), datetime.date(2021, 11, 27),
datetime.date(2021, 11, 28), datetime.date(2021, 11, 29),
datetime.date(2021, 11, 30), datetime.date(2021, 12, 1),
datetime.date(2021, 12, 2), datetime.date(2021, 12, 3),
datetime.date(2021, 12, 4), datetime.date(2021, 12, 5),
datetime.date(2021, 12, 6), datetime.date(2021, 12, 7),
datetime.date(2021, 12, 8), datetime.date(2021, 12, 9),
datetime.date(2021, 12, 10), datetime.date(2021, 12, 11),
datetime.date(2021, 12, 12), datetime.date(2021, 12, 13),
datetime.date(2021, 12, 14), datetime.date(2021, 12, 15),
datetime.date(2021, 12, 16), datetime.date(2021, 12, 17),
datetime.date(2021, 12, 18), datetime.date(2021, 12, 19),
datetime.date(2021, 12, 20), datetime.date(2021, 12, 21),
datetime.date(2021, 12, 22), datetime.date(2021, 12, 23),
datetime.date(2021, 12, 24), datetime.date(2021, 12, 25),
datetime.date(2021, 12, 26), datetime.date(2021, 12, 27),
datetime.date(2021, 12, 28), datetime.date(2021, 12, 29),
datetime.date(2021, 12, 30), datetime.date(2021, 12, 31),
```

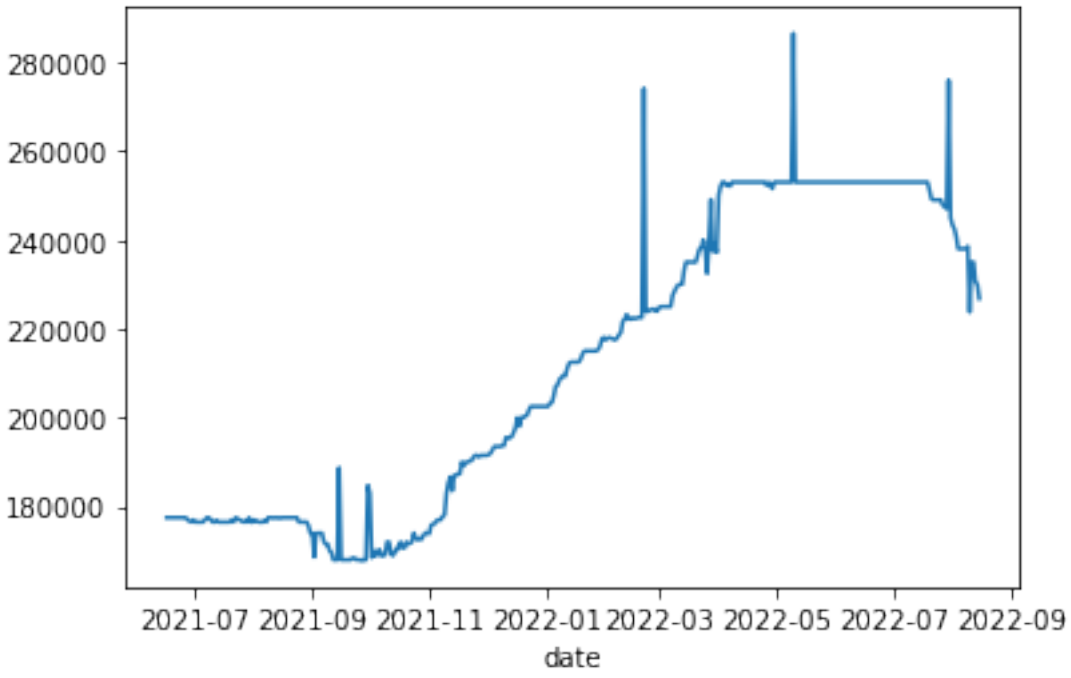
```
datetime.date(2022, 1, 1), datetime.date(2022, 1, 2),
datetime.date(2022, 1, 3), datetime.date(2022, 1, 4),
datetime.date(2022, 1, 5), datetime.date(2022, 1, 6),
datetime.date(2022, 1, 7), datetime.date(2022, 1, 8),
datetime.date(2022, 1, 9), datetime.date(2022, 1, 10),
datetime.date(2022, 1, 11), datetime.date(2022, 1, 12),
datetime.date(2022, 1, 13), datetime.date(2022, 1, 14),
datetime.date(2022, 1, 15), datetime.date(2022, 1, 16),
datetime.date(2022, 1, 17), datetime.date(2022, 1, 18),
datetime.date(2022, 1, 19), datetime.date(2022, 1, 20),
datetime.date(2022, 1, 21), datetime.date(2022, 1, 22),
datetime.date(2022, 1, 23), datetime.date(2022, 1, 24),
datetime.date(2022, 1, 25), datetime.date(2022, 1, 26),
datetime.date(2022, 1, 27), datetime.date(2022, 1, 28),
datetime.date(2022, 1, 29), datetime.date(2022, 1, 30),
datetime.date(2022, 1, 31), datetime.date(2022, 2, 1),
datetime.date(2022, 2, 2), datetime.date(2022, 2, 3),
datetime.date(2022, 2, 4), datetime.date(2022, 2, 5),
datetime.date(2022, 2, 6), datetime.date(2022, 2, 7),
datetime.date(2022, 2, 8), datetime.date(2022, 2, 9),
datetime.date(2022, 2, 10), datetime.date(2022, 2, 11),
datetime.date(2022, 2, 12), datetime.date(2022, 2, 13),
datetime.date(2022, 2, 14), datetime.date(2022, 2, 15),
datetime.date(2022, 2, 16), datetime.date(2022, 2, 17),
datetime.date(2022, 2, 18), datetime.date(2022, 2, 19),
datetime.date(2022, 2, 20), datetime.date(2022, 2, 21),
datetime.date(2022, 2, 22), datetime.date(2022, 2, 23),
datetime.date(2022, 2, 24), datetime.date(2022, 2, 25),
datetime.date(2022, 2, 26), datetime.date(2022, 2, 27),
datetime.date(2022, 2, 28), datetime.date(2022, 3, 1),
datetime.date(2022, 3, 2), datetime.date(2022, 3, 3),
datetime.date(2022, 3, 4), datetime.date(2022, 3, 5),
datetime.date(2022, 3, 6), datetime.date(2022, 3, 7),
datetime.date(2022, 3, 8), datetime.date(2022, 3, 9),
datetime.date(2022, 3, 10), datetime.date(2022, 3, 11),
datetime.date(2022, 3, 12), datetime.date(2022, 3, 13),
datetime.date(2022, 3, 14), datetime.date(2022, 3, 15),
datetime.date(2022, 3, 16), datetime.date(2022, 3, 17),
datetime.date(2022, 3, 18), datetime.date(2022, 3, 19),
datetime.date(2022, 3, 20), datetime.date(2022, 3, 21),
datetime.date(2022, 3, 22), datetime.date(2022, 3, 23),
datetime.date(2022, 3, 24), datetime.date(2022, 3, 25),
datetime.date(2022, 3, 26), datetime.date(2022, 3, 27),
datetime.date(2022, 3, 28), datetime.date(2022, 3, 29),
datetime.date(2022, 3, 30), datetime.date(2022, 3, 31),
datetime.date(2022, 4, 1), datetime.date(2022, 4, 2),
datetime.date(2022, 4, 3), datetime.date(2022, 4, 4),
datetime.date(2022, 4, 5), datetime.date(2022, 4, 6),
```

```
datetime.date(2022, 4, 7), datetime.date(2022, 4, 8),
datetime.date(2022, 4, 9), datetime.date(2022, 4, 10),
datetime.date(2022, 4, 11), datetime.date(2022, 4, 12),
datetime.date(2022, 4, 13), datetime.date(2022, 4, 14),
datetime.date(2022, 4, 15), datetime.date(2022, 4, 16),
datetime.date(2022, 4, 17), datetime.date(2022, 4, 18),
datetime.date(2022, 4, 19), datetime.date(2022, 4, 20),
datetime.date(2022, 4, 21), datetime.date(2022, 4, 22),
datetime.date(2022, 4, 23), datetime.date(2022, 4, 24),
datetime.date(2022, 4, 25), datetime.date(2022, 4, 26),
datetime.date(2022, 4, 27), datetime.date(2022, 4, 28),
datetime.date(2022, 4, 29), datetime.date(2022, 4, 30),
datetime.date(2022, 5, 3), datetime.date(2022, 5, 4),
datetime.date(2022, 5, 5), datetime.date(2022, 5, 6),
datetime.date(2022, 5, 7), datetime.date(2022, 5, 8),
datetime.date(2022, 5, 9), datetime.date(2022, 5, 10),
datetime.date(2022, 5, 11), datetime.date(2022, 5, 12),
datetime.date(2022, 5, 13), datetime.date(2022, 5, 14),
datetime.date(2022, 5, 15), datetime.date(2022, 5, 16),
datetime.date(2022, 5, 17), datetime.date(2022, 5, 18),
datetime.date(2022, 5, 19), datetime.date(2022, 5, 20),
datetime.date(2022, 5, 21), datetime.date(2022, 5, 22),
datetime.date(2022, 5, 23), datetime.date(2022, 5, 24),
datetime.date(2022, 5, 25), datetime.date(2022, 5, 26),
datetime.date(2022, 5, 27), datetime.date(2022, 5, 28),
datetime.date(2022, 5, 29), datetime.date(2022, 5, 30),
datetime.date(2022, 5, 31), datetime.date(2022, 6, 1),
datetime.date(2022, 6, 2), datetime.date(2022, 6, 3),
datetime.date(2022, 6, 4), datetime.date(2022, 6, 5),
datetime.date(2022, 6, 6), datetime.date(2022, 6, 7),
datetime.date(2022, 6, 8), datetime.date(2022, 6, 9),
datetime.date(2022, 6, 10), datetime.date(2022, 6, 11),
datetime.date(2022, 6, 12), datetime.date(2022, 6, 13),
datetime.date(2022, 6, 14), datetime.date(2022, 6, 15),
datetime.date(2022, 6, 16), datetime.date(2022, 6, 17),
datetime.date(2022, 6, 18), datetime.date(2022, 6, 19),
datetime.date(2022, 6, 20), datetime.date(2022, 6, 21),
datetime.date(2022, 6, 22), datetime.date(2022, 6, 23),
datetime.date(2022, 6, 24), datetime.date(2022, 6, 25),
datetime.date(2022, 6, 26), datetime.date(2022, 6, 27),
datetime.date(2022, 6, 28), datetime.date(2022, 6, 29),
datetime.date(2022, 6, 30), datetime.date(2022, 7, 1),
datetime.date(2022, 7, 2), datetime.date(2022, 7, 3),
datetime.date(2022, 7, 4), datetime.date(2022, 7, 5),
datetime.date(2022, 7, 6), datetime.date(2022, 7, 7),
datetime.date(2022, 7, 8), datetime.date(2022, 7, 9),
datetime.date(2022, 7, 10), datetime.date(2022, 7, 11),
datetime.date(2022, 7, 12), datetime.date(2022, 7, 13),
```

```
datetime.date(2022, 7, 14), datetime.date(2022, 7, 15),
datetime.date(2022, 7, 16), datetime.date(2022, 7, 17),
datetime.date(2022, 7, 18), datetime.date(2022, 7, 19),
datetime.date(2022, 7, 20), datetime.date(2022, 7, 21),
datetime.date(2022, 7, 22), datetime.date(2022, 7, 23),
datetime.date(2022, 7, 24), datetime.date(2022, 7, 25),
datetime.date(2022, 7, 26), datetime.date(2022, 7, 27),
datetime.date(2022, 7, 28), datetime.date(2022, 7, 29),
datetime.date(2022, 7, 30), datetime.date(2022, 7, 31),
datetime.date(2022, 8, 1), datetime.date(2022, 8, 2),
datetime.date(2022, 8, 3), datetime.date(2022, 8, 4),
datetime.date(2022, 8, 5), datetime.date(2022, 8, 6),
datetime.date(2022, 8, 7), datetime.date(2022, 8, 8),
datetime.date(2022, 8, 9), datetime.date(2022, 8, 10),
datetime.date(2022, 8, 11), datetime.date(2022, 8, 12),
datetime.date(2022, 8, 13), datetime.date(2022, 8, 14),
datetime.date(2022, 8, 15)]
```

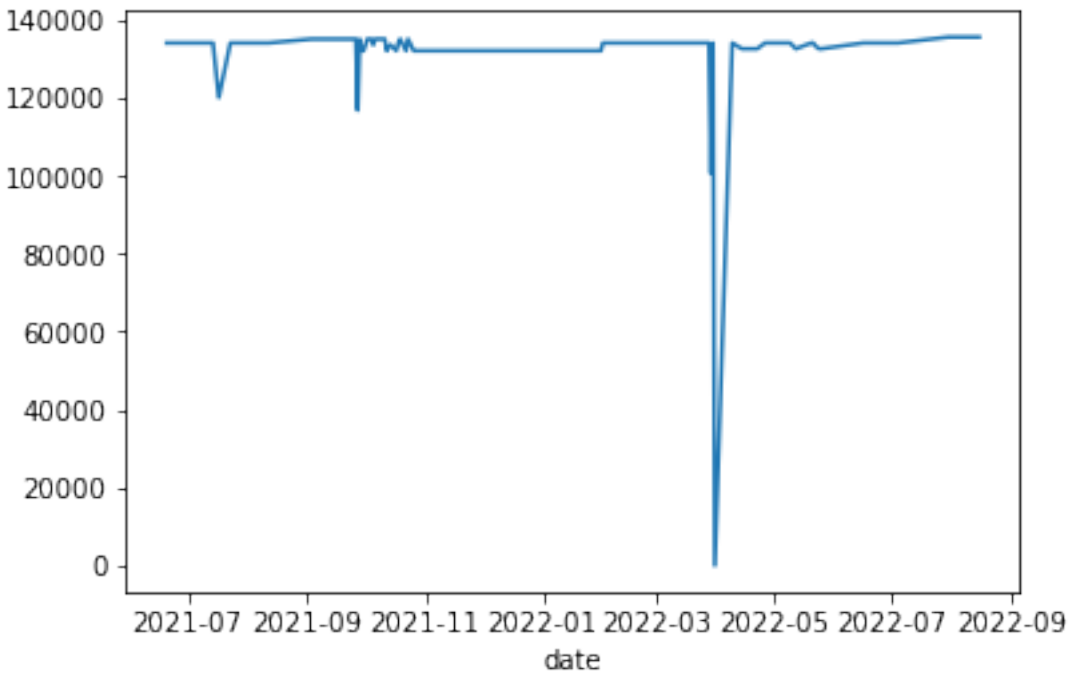
[31]:
```
baskets[baskets.sku_id == 438].groupby('date').mean().price.plot()
```

[31]:
```
<AxesSubplot:xlabel='date'>
```



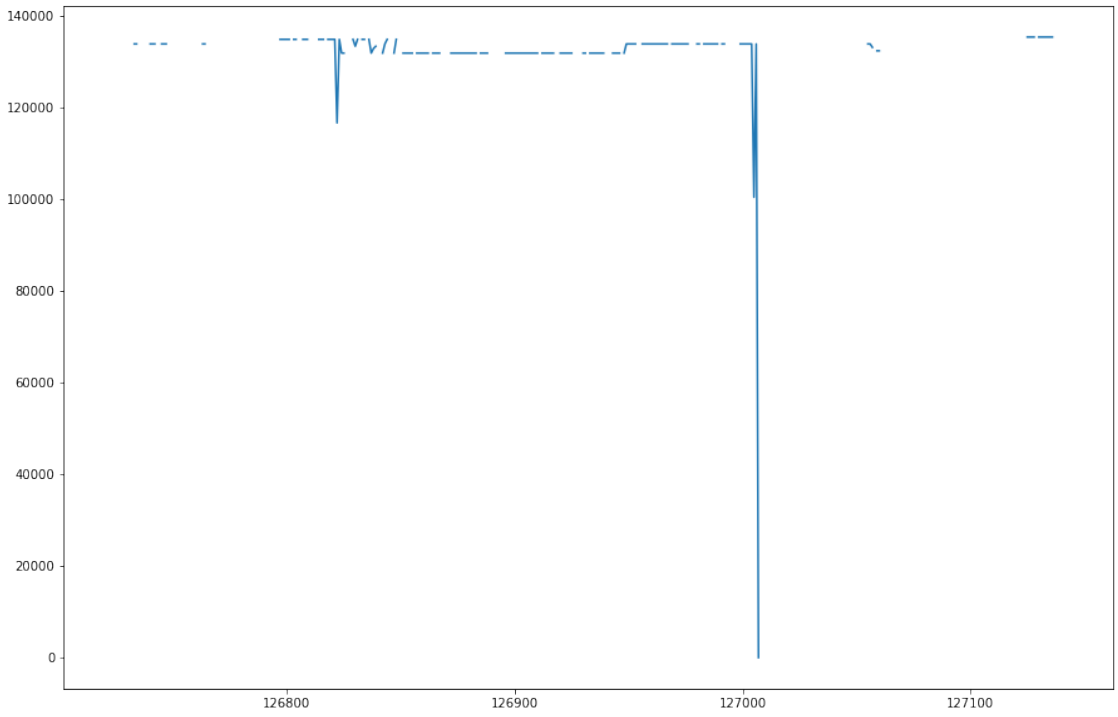
[32]:
```
baskets[baskets.sku_id == 277].groupby('date').mean().price.plot()
```

[32]:
```
<AxesSubplot:xlabel='date'>
```



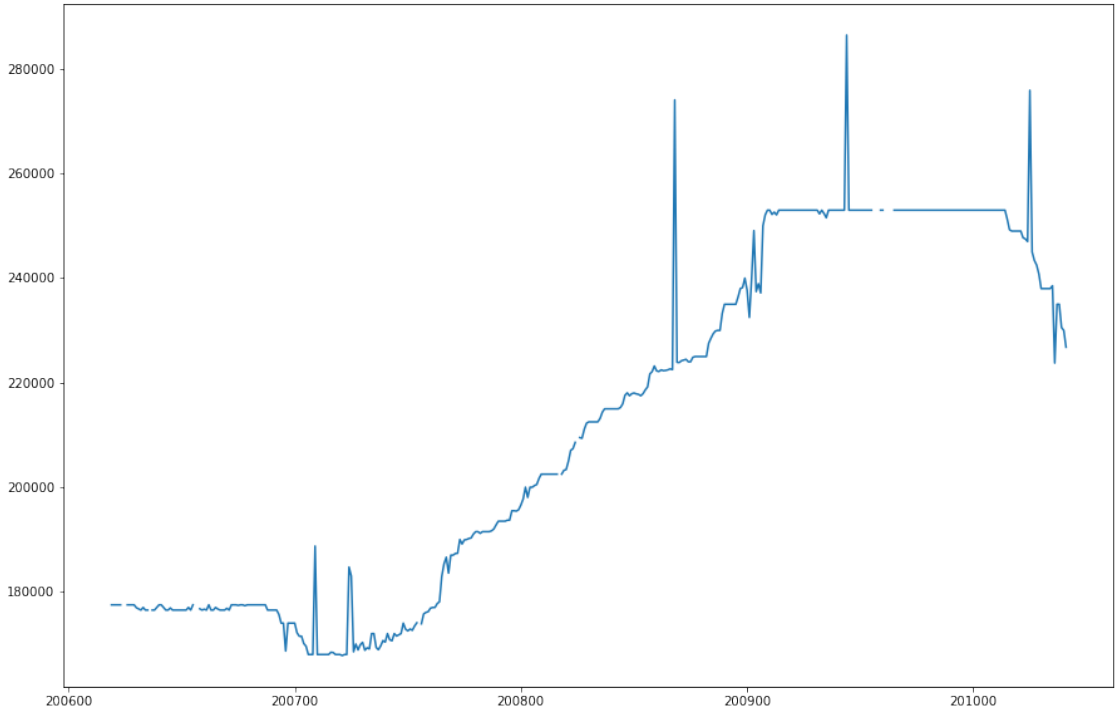
[33]:
```
plt.figure(figsize=(15,10))
skus[skus["sku_id"] == 277].avg_price_by_day.plot()
```

[33]: <AxesSubplot:>



[34]:
```
plt.figure(figsize=(15,10))
skus[skus["sku_id"] == 438].avg_price_by_day.plot()
```

[34]: <AxesSubplot:>

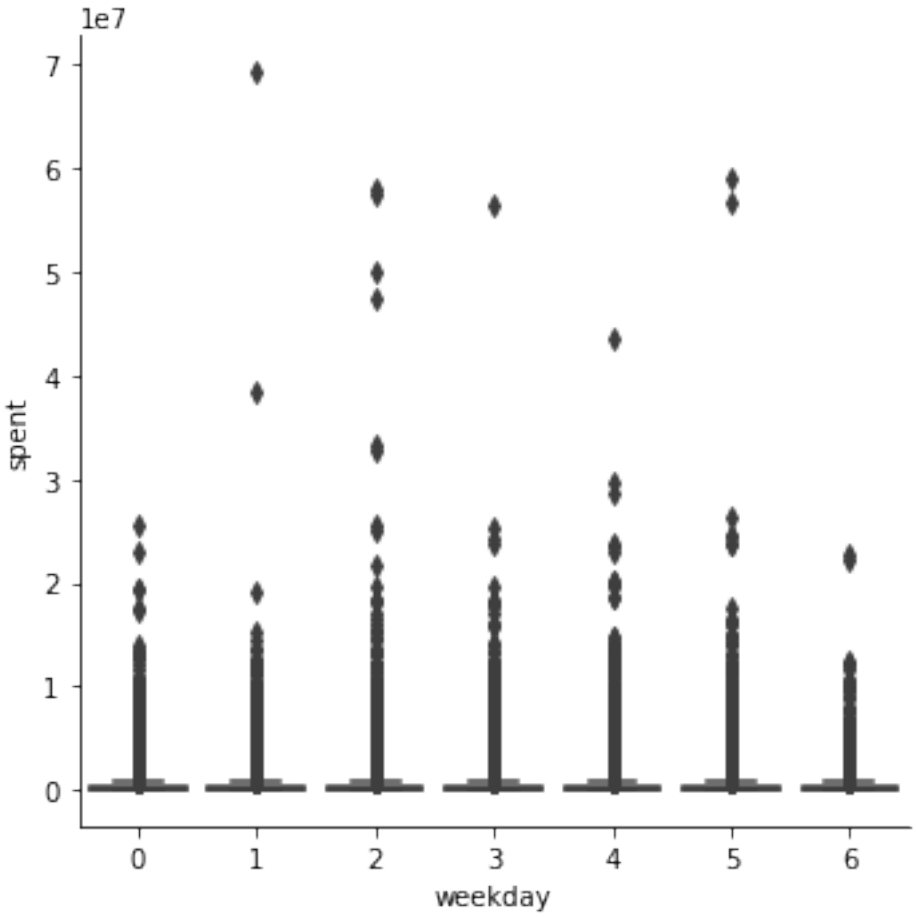


[35]:
```
skus10.groupby(['sku_id']).count().date.sort_values()
```

[35]:
```
sku_id
5       396
1516    396
1515    396
1514    396
1513    396
       ...
944     396
943     396
942     396
952     396
2383    396
Name: date, Length: 1348, dtype: int64
```

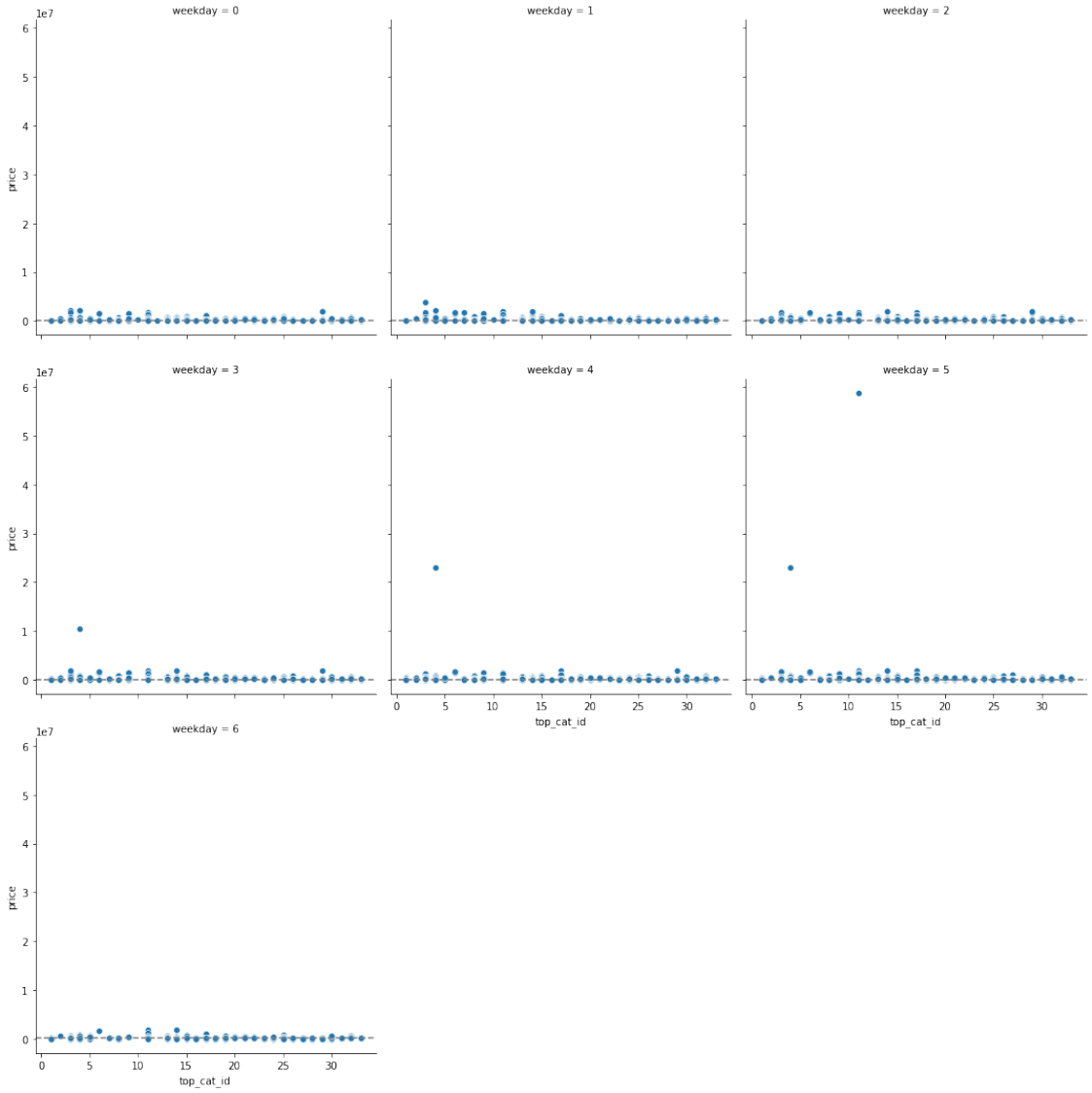
[36]:
```
plt.figure(figsize=(15,10))
ax = sns.catplot(x="weekday", y = "spent",data=baskets250,␣
↪palette="Set1",kind="box")
```

```
<Figure size 1080x720 with 0 Axes>
```



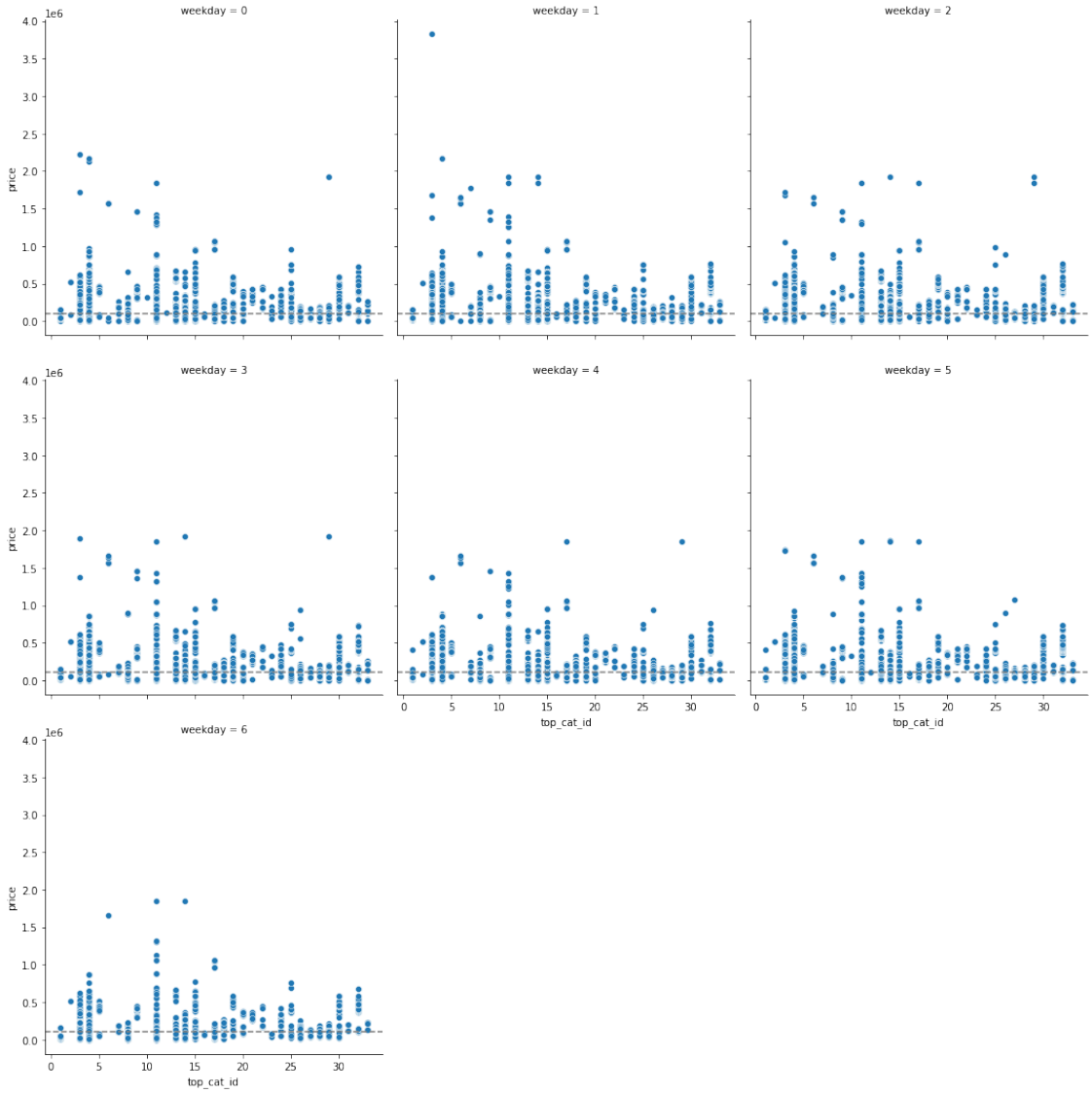
[80]:
```
plt.figure(figsize=(15,30))
g = sns.FacetGrid(baskets, col="weekday", height=5, col_wrap=3)
g.map_dataframe(sns.scatterplot, x="top_cat_id",y="price")
g.refline(y=baskets["price"].median())
```

```
<Figure size 1080x2160 with 0 Axes>
```



[90]:
```
baskets1 = baskets.drop(baskets[baskets['price']> 10000000].index,axis=0)
plt.figure(figsize=(15,30))
g = sns.FacetGrid(baskets1, col="weekday", height=5, col_wrap=3)
g.map_dataframe(sns.scatterplot, x="top_cat_id",y="price")
g.refline(y=baskets1["price"].median())
```

```
<Figure size 1080x2160 with 0 Axes>
```



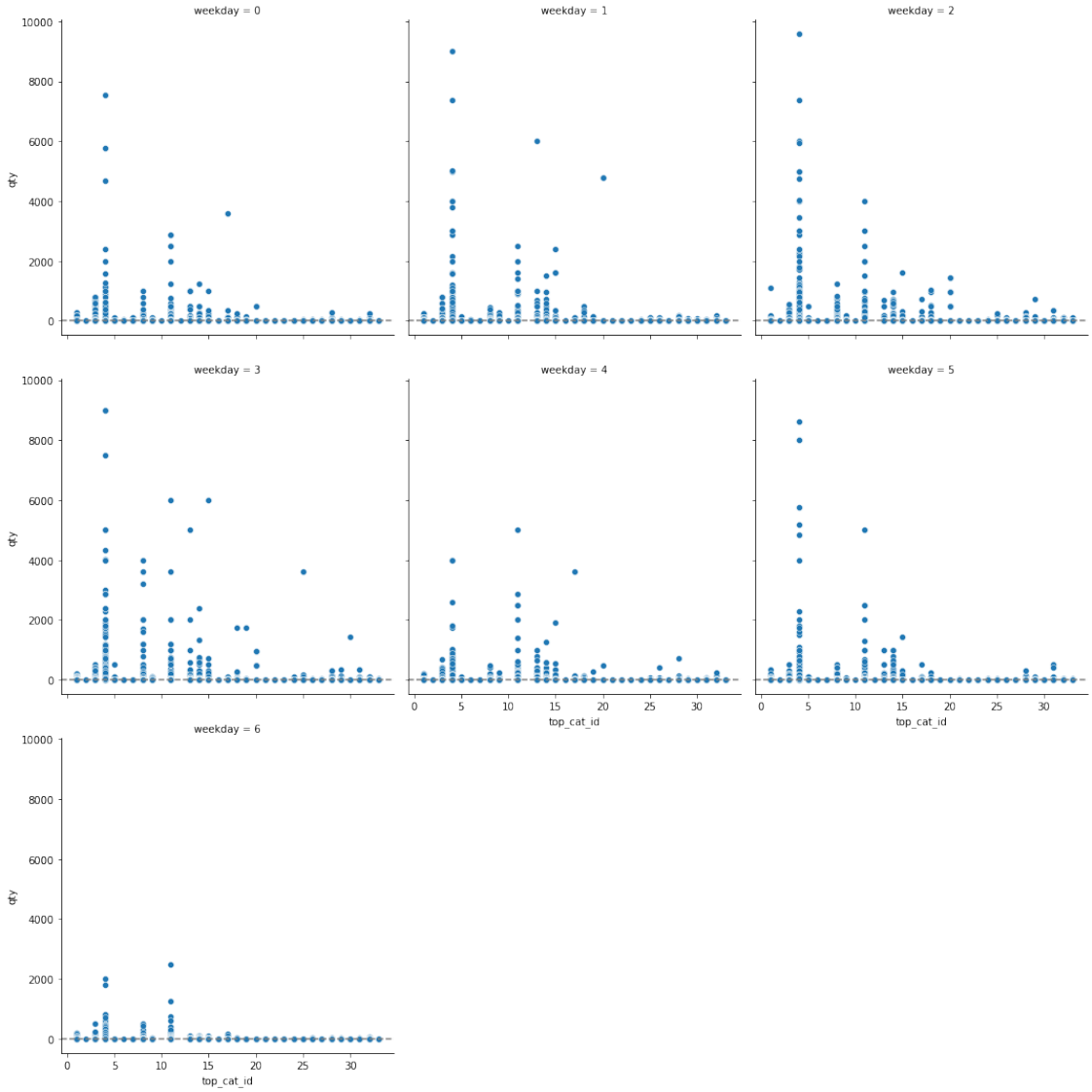
[102]:
```
baskets1 = baskets.drop(baskets[baskets['price']> 1000000].index,axis=0)
plt.figure(figsize=(15,30))
g = sns.FacetGrid(baskets1, col="month", hue='weekday', height=5, col_wrap=3)

g.map_dataframe(sns.scatterplot, x="top_cat_id",y="price")
g.refline(y=baskets1["price"].median())
```

```
<Figure size 1080x2160 with 0 Axes>
```

[103]:
```
baskets1 = baskets.drop(baskets[baskets['qty']> 10000].index,axis=0)
plt.figure(figsize=(15,30))
```

```
g = sns.FacetGrid(baskets1, col="weekday", height=5, col_wrap=3)
g.map_dataframe(sns.scatterplot, x="top_cat_id",y="qty")
g.refline(y=baskets1["qty"].median())
```

```
<Figure size 1080x2160 with 0 Axes>
```

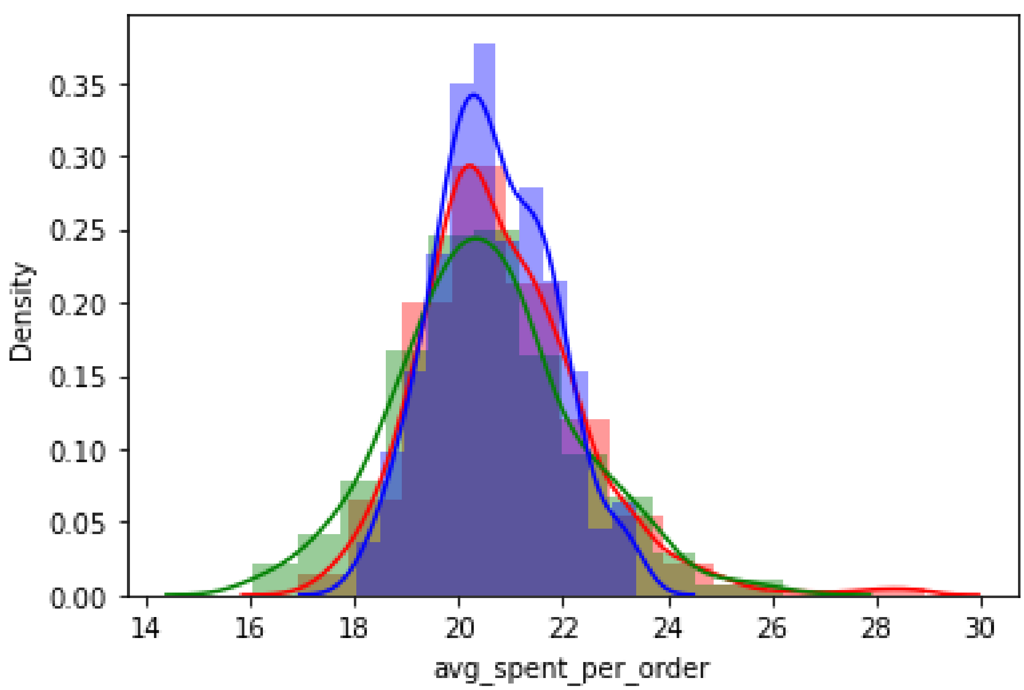


[40]:
```
from turtle import color


sns.distplot(merchants.avg_spent_per_order.apply(lambda x: math.
 ↪log(x,2)),bins=12,color='red')
sns.distplot(merchants10.avg_spent_per_order.apply(lambda x: math.
 ↪log(x,2)),bins=12,color='green')
```

```
sns.distplot(merchants250.avg_spent_per_order.apply(lambda x: math.
 ↪log(x,2)),bins=12,color='blue')
```

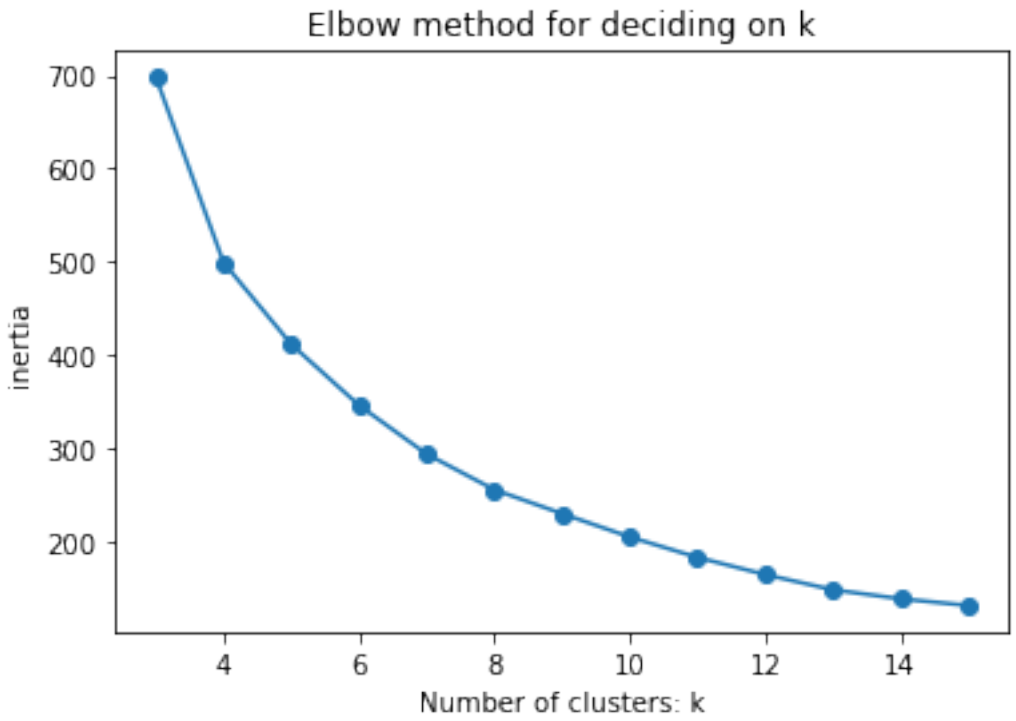
```
/Users/yingli/opt/anaconda3/lib/python3.9/site-
packages/seaborn/distributions.py:2619: FutureWarning: `distplot` is a
deprecated function and will be removed in a future version. Please adapt your
code to use either `displot` (a figure-level function with similar flexibility)
or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
/Users/yingli/opt/anaconda3/lib/python3.9/site-
packages/seaborn/distributions.py:2619: FutureWarning: `distplot` is a
deprecated function and will be removed in a future version. Please adapt your
code to use either `displot` (a figure-level function with similar flexibility)
or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
/Users/yingli/opt/anaconda3/lib/python3.9/site-
packages/seaborn/distributions.py:2619: FutureWarning: `distplot` is a
deprecated function and will be removed in a future version. Please adapt your
code to use either `displot` (a figure-level function with similar flexibility)
or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

[40]: <AxesSubplot:xlabel='avg_spent_per_order', ylabel='Density'>



[45]:
```
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler, normalize
from sklearn.metrics import silhouette_score



def find_elbow(df, colnames, clusters_range):
    df_for_cluster = df.loc[:,colnames]
    stscaler = StandardScaler().fit(df_for_cluster)
    normalized_df = stscaler.transform(df_for_cluster)

    inertias = [] # wcss: Within Cluster Sum of Squares
    for k in clusters_range:
        kmeans = KMeans(init='k-means++',n_clusters=k,n_init=100, max_iter=300,␣
 ↪random_state=0).fit(normalized_df)
        inertias.append(kmeans.inertia_)
    plt.figure()
    plt.plot(clusters_range,inertias, marker='o')
    plt.title('Elbow method for deciding on k')
    plt.xlabel('Number of clusters: k')
    plt.ylabel('inertia')
    plt.show()
    return

def run_kmeans(df, colnames, k):
    df_for_cluster = df.loc[:,colnames]
    stscaler = StandardScaler().fit(df_for_cluster)
    normalized_df = stscaler.transform(df_for_cluster)

    kmeans = KMeans(init='k-means++',n_clusters=k,n_init=100, max_iter=300,␣
 ↪random_state=0).fit(normalized_df)
    df['cluster'] = kmeans.labels_
    return df
```

[46]:
```
df = merchants10
colnames = merchants10.columns[2:]
clusters_range = [3,4,5,6,7,8,9,10,11,12,13,14,15]
find_elbow(df, colnames,clusters_range)

#kmeans.cluster_centers_
```



[47]:
```
df = merchants10
colnames = merchants10.columns[2:]
k = 6
merchants10_kmeans = run_kmeans(df, colnames,k)
merchants10_kmeans.groupby("cluster").size()
```

[47]:
```
cluster
0     55
1    151
2      5
3     10
4     14
5     82
dtype: int64
```

[48]:
```
df = merchants10
colnames = merchants10.columns[2:]
k = 5
merchants10_kmeans = run_kmeans(df, colnames,k)
merchants10_kmeans.groupby("cluster").size()
```

[48]:
```
cluster
0     82
```

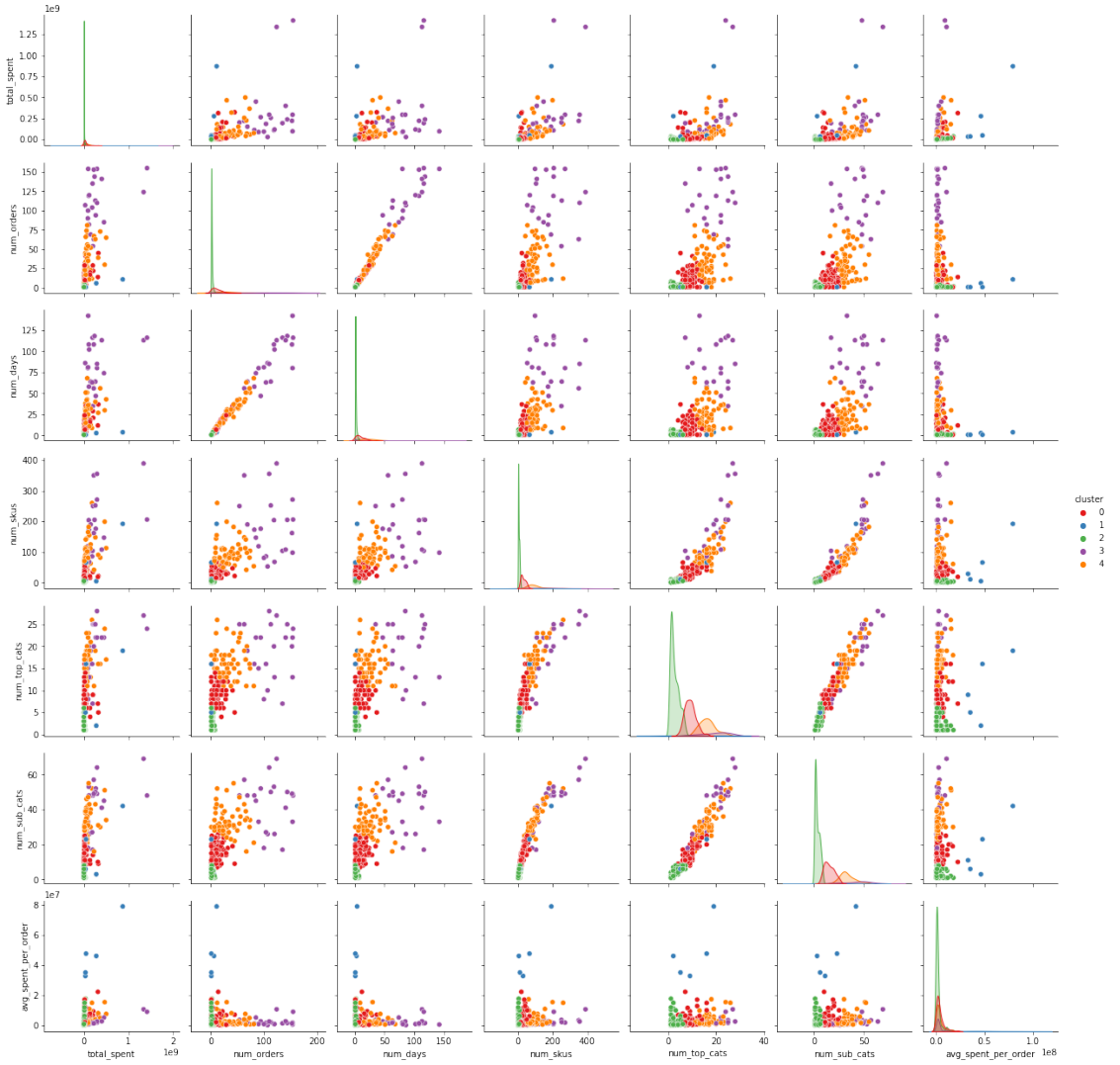
```
1      5
2    151
3     22
4     57
dtype: int64
```

[49]:
```
df = merchants
colnames = merchants.columns[2:]
k = 5
merchants_kmeans = run_kmeans(df, colnames,k)
merchants_kmeans.groupby("cluster").size()
```

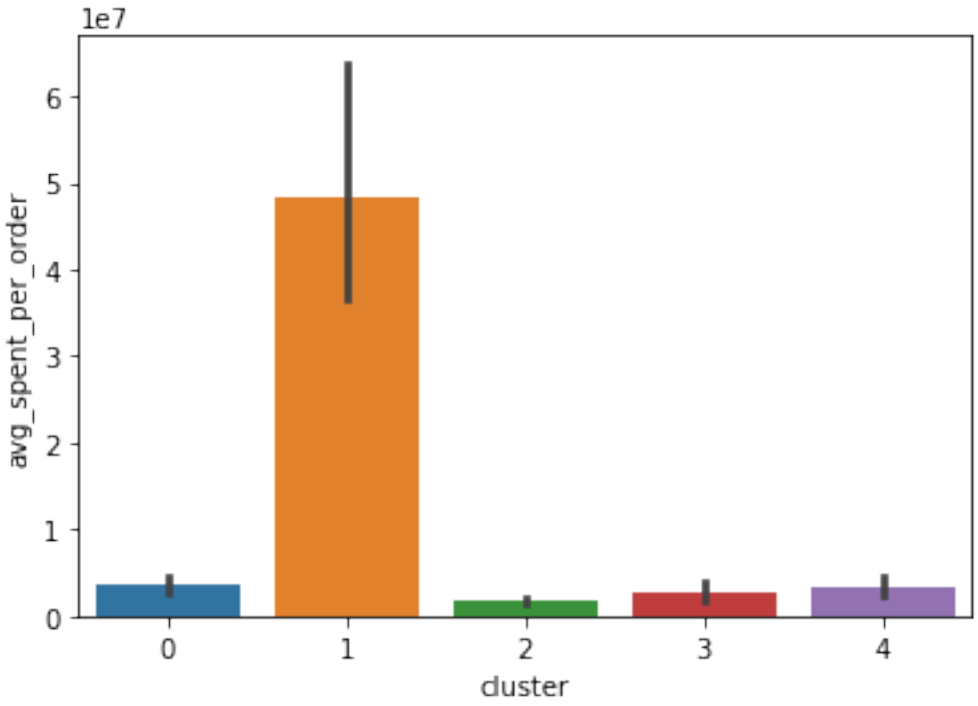
[49]:
```
cluster
0     205
1    1193
2       1
3     725
4      14
dtype: int64
```

[51]:
```
sns.pairplot(data=merchants10_kmeans, hue="cluster", palette="Set1")
```

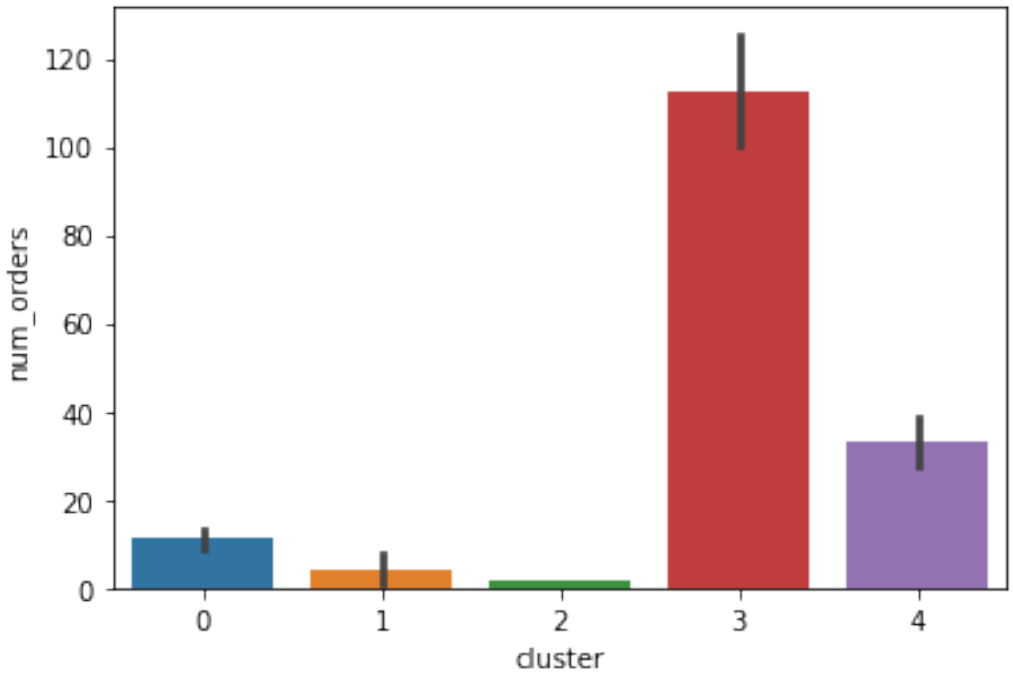
[51]:
```
<seaborn.axisgrid.PairGrid at 0x7fed7a33c070>
```



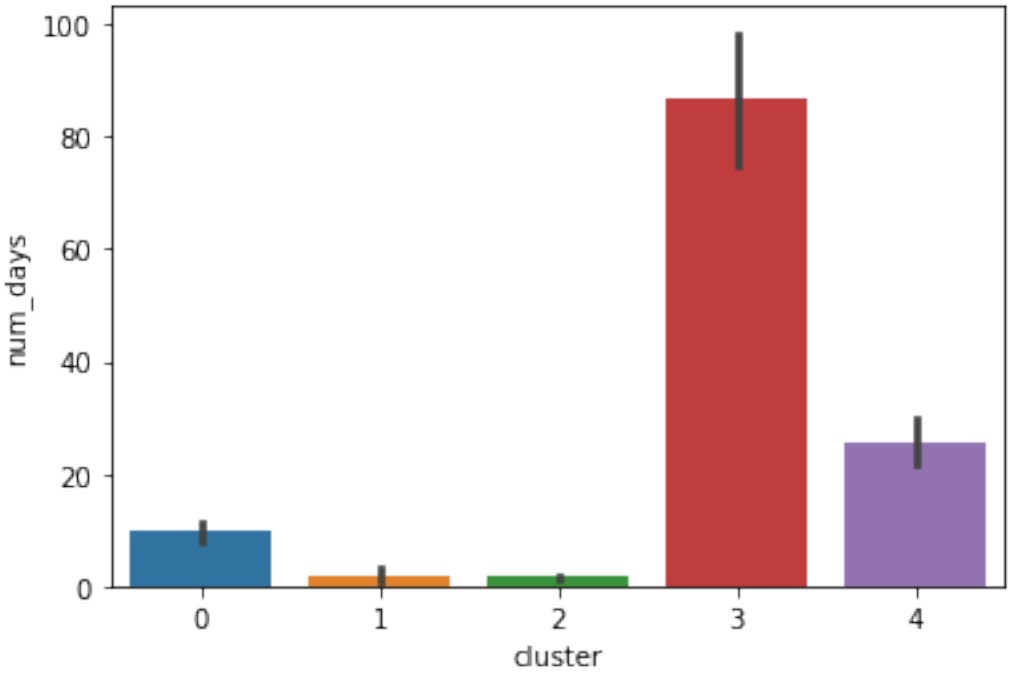
[52]:
```
ax = sns.barplot(x="cluster", y="avg_spent_per_order", data=merchants10_kmeans)
```



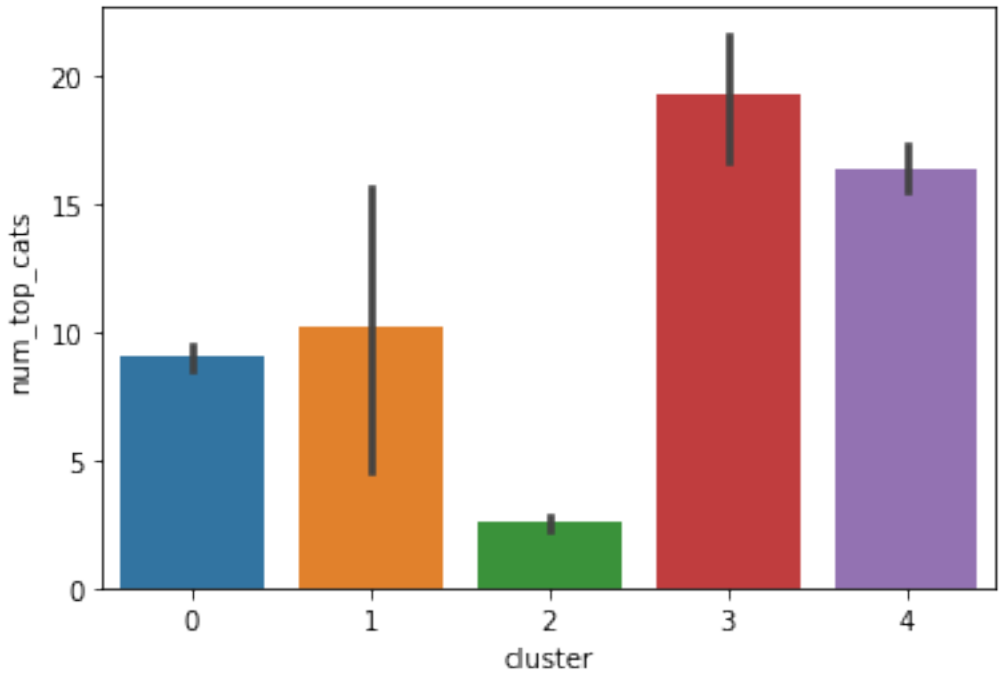
[53]:
```
ax = sns.barplot(x="cluster", y="num_orders", data=merchants10_kmeans)
```



[54]:
```
ax = sns.barplot(x="cluster", y="num_days", data=merchants10_kmeans)
```



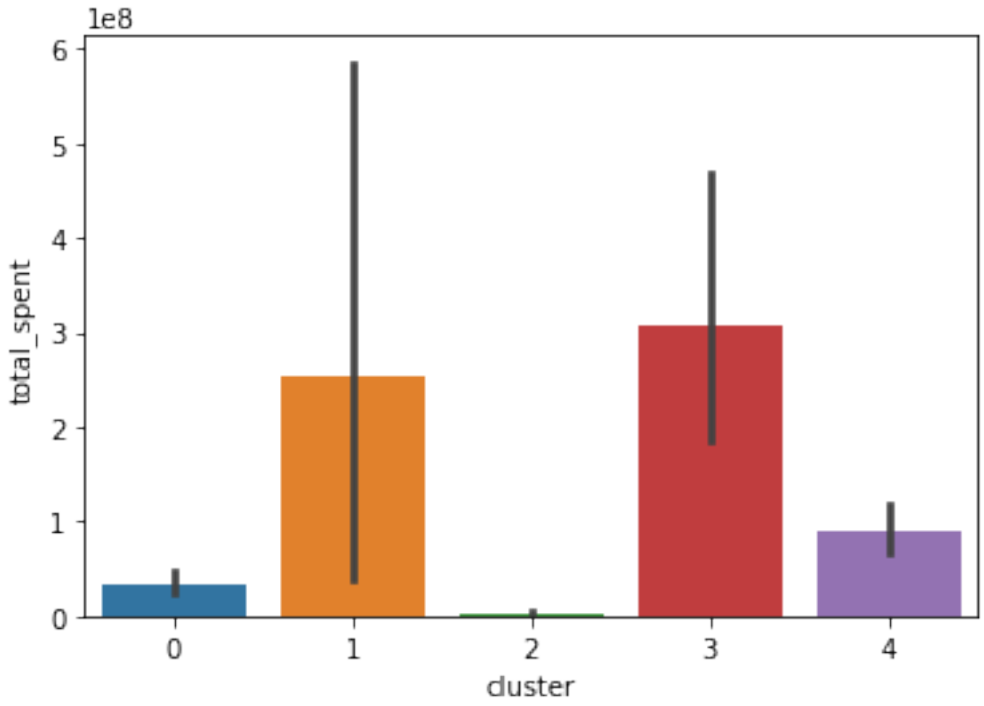
[55]:
```
ax = sns.barplot(x="cluster", y="num_top_cats", data=merchants10_kmeans)
```



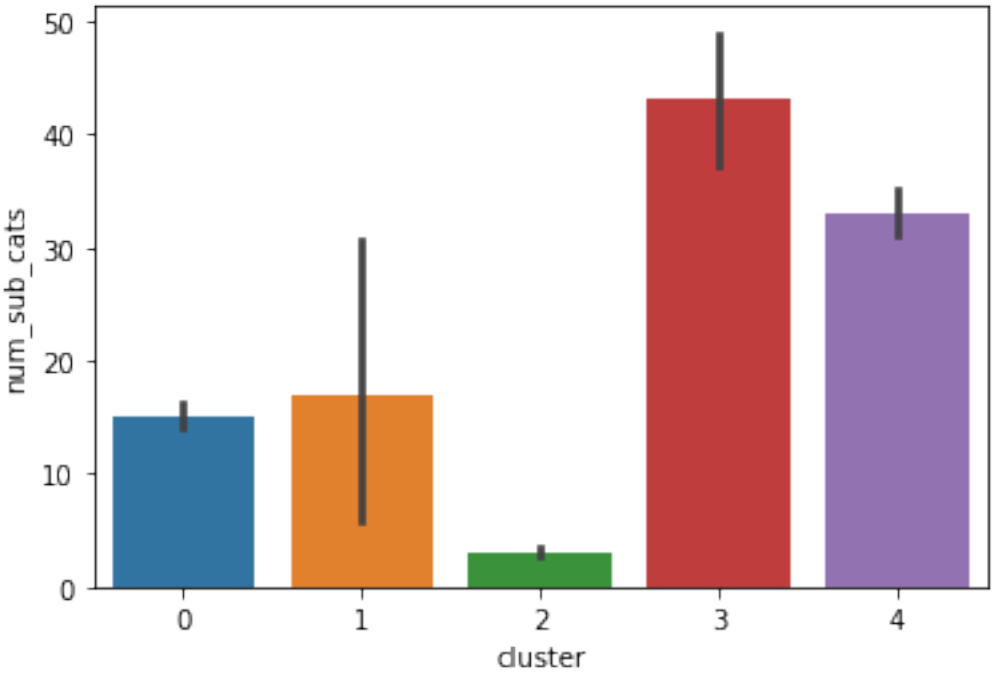
[56]:
```
merchants10_kmeans.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 317 entries, 0 to 316
Data columns (total 9 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   merchant_id          317 non-null    category
 1   total_spent          317 non-null    float64
 2   num_orders           317 non-null    int64
 3   num_days             317 non-null    int64
 4   num_skus             317 non-null    int64
 5   num_top_cats         317 non-null    int64
 6   num_sub_cats         317 non-null    int64
 7   avg_spent_per_order  317 non-null    float64
 8   cluster              317 non-null    int32
dtypes: category(1), float64(2), int32(1), int64(5)
memory usage: 29.9 KB
```

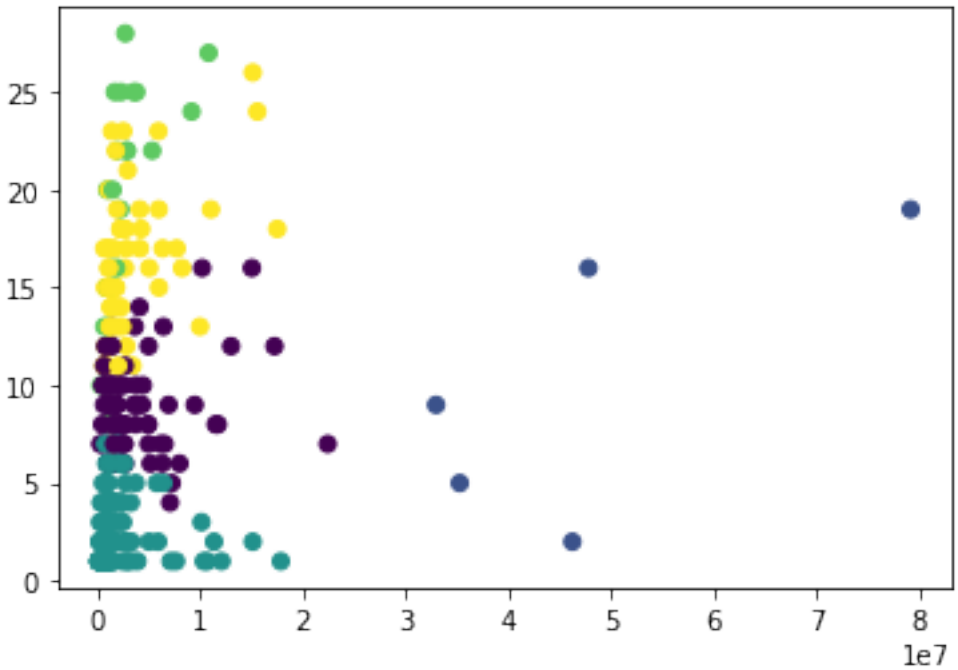
[57]:
```
ax = sns.barplot(x="cluster", y="total_spent", data=merchants10_kmeans)
```



[58]:
```
ax = sns.barplot(x="cluster", y="num_sub_cats", data=merchants10_kmeans)
```



[59]:
```
plt.scatter(merchants10["avg_spent_per_order"],␣
↪merchants10["num_top_cats"],c=merchants10.cluster)
plt.show()
```



[ ]:
```
#Counting the frequency
freq = baskets['merchant_id'].value_counts()
print(freq)
```

[ ]:
```
frequency_df = baskets.groupby(
    by=['merchant_id'], as_index=False)['date'].count()
frequency_df.columns = ['merchant_id', 'Frequency']
frequency_df.head()
```

[ ]:
```
merchants.avg_spent_per_order.sort_values(ascending=False).
 ↪reset_index(drop=True).head(33)
```

[ ]:
```
merchants.num_days.plot()
```

[ ]:
```
merchants.total_spent.plot()
```

[ ]:
```
wcss = []
for i in range(1, 11):
    kmeans = KMeans(n_clusters=i, init='k-means++', max_iter=300, n_init=10,␣
 ↪random_state=0)
    kmeans.fit(merchants_data)
    wcss.append(kmeans.inertia_)
plt.plot(range(1, 11), wcss)
plt.title('Elbow Method')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS')
plt.show()
```

[ ]:
```
ax = sns.barplot(x="cluster", y="avg_spent_per_order", data=merchants_data)
```

0.0.2 cluster 3 looks like containing merchants who have high average spending per orders.

[ ]:
```
ax = sns.barplot(x="cluster", y="num_orders", data=merchants_data)
```

[ ]:
```
plt.figure(figsize=(15,10))
ax = sns.catplot(x="num_days", y = "num_top_cats",data=merchants_data,␣
 ↪hue="cluster", palette="Set1",ci="sd")
```

[ ]:
```
plt.figure(figsize=(15,10))
ax = sns.catplot(x="num_days", y = "num_sub_cats",data=merchants_data,␣
 ↪hue="cluster", palette="Set1",ci="sd")
```

[ ]:
```
plt.figure(figsize=(15,10))
```

```
ax = sns.catplot(x="num_days", y = "num_orders",data=merchants_data,␣
 ↪hue="cluster", palette="Set1",ci="sd")
```

[ ]:
```
plt.figure(figsize=(15,10))
ax = sns.catplot(x="avg_spent_per_order", y = "num_days",data=merchants_data,␣
 ↪hue="cluster", palette="Set1",ci="sd")
```

0.0.3 cluster 4 consists of the merchants that have high number of days, 4 of them all above 100?

[ ]:
```

```