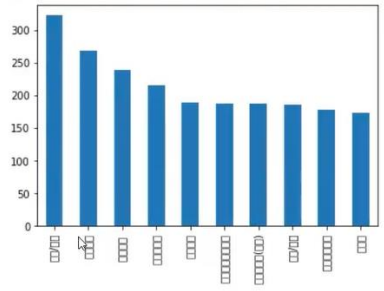
本次实验主要针对了不同维度进行了数据分析

针对订单 orderid:包括了什么菜最受欢迎

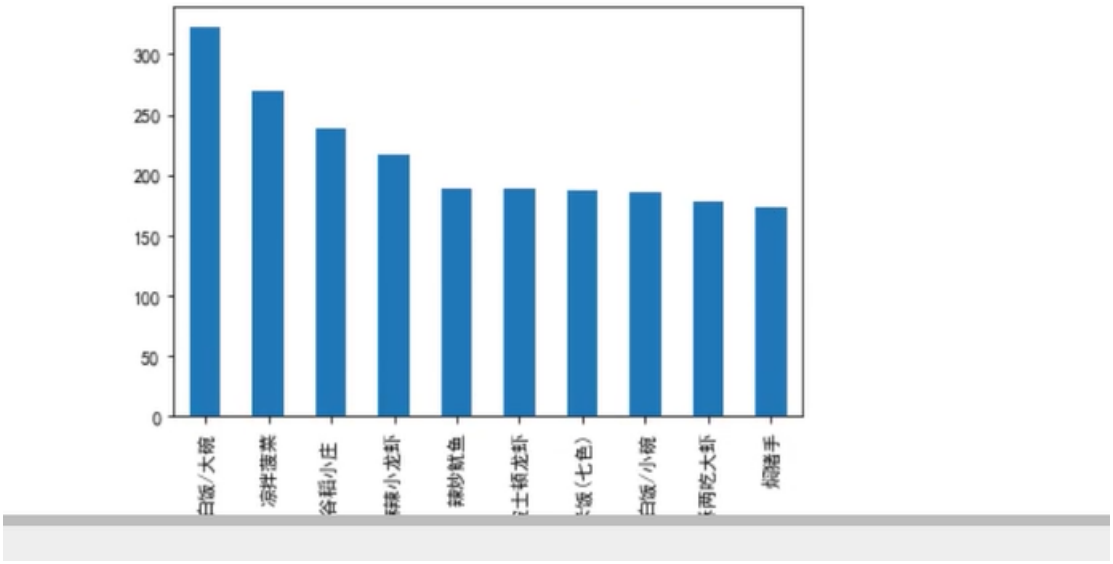
```
                                  1
[18]: 白饭/大碗                   323
      凉拌菠菜                    269
      谷稻小庄                    239
      麻辣小龙虾                  216
      辣炒鱿鱼                    189
                                 ...
      铁板牛肉                      3
      张裕葡萄酒张裕赤霞珠干红\r\n      2
      五香酱驴肉\r\n                2
      蒙古烤羊腿                    1
      香辣爆羊肝\r\n                1
      Name: dishes_name, Length: 158, dtype: int64
```

遇到问题：画统计图时出现错误，并且数据部分名称缺失，解决办法使用指令%matplotlib inline

```
  font.set_text(s, 0, flags=flags)
D:\ProgramData\Anaconda3\lib\site-packages\matplotlib\backends\backend_agg.py:180: RuntimeWarning: Glyph 21507 missing from current font.
  font.set_text(s, 0, flags=flags)
D:\ProgramData\Anaconda3\lib\site-packages\matplotlib\backends\backend_agg.py:180: RuntimeWarning: Glyph 28950 missing from current font.
  font.set_text(s, 0, flags=flags)
D:\ProgramData\Anaconda3\lib\site-packages\matplotlib\backends\backend_agg.py:180: RuntimeWarning: Glyph 29482 missing from current font.
  font.set_text(s, 0, flags=flags)
D:\ProgramData\Anaconda3\lib\site-packages\matplotlib\backends\backend_agg.py:180: RuntimeWarning: Glyph 25163 missing from current font.
  font.set_text(s, 0, flags=flags)
```

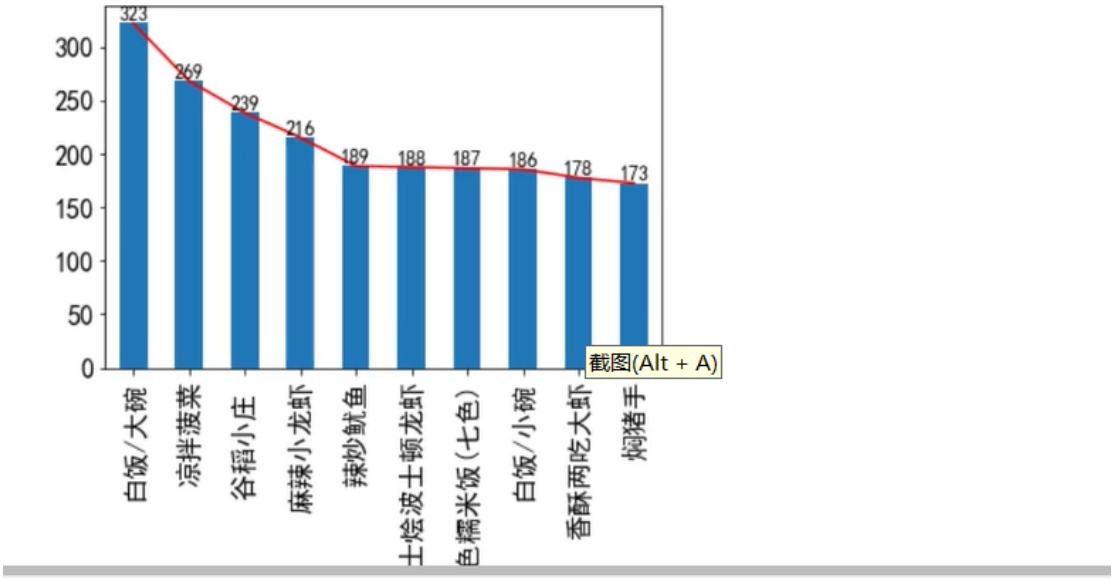


```
[24]: <matplotlib.axes._subplots.AxesSubplot at 0x24c6fc32e08>
```



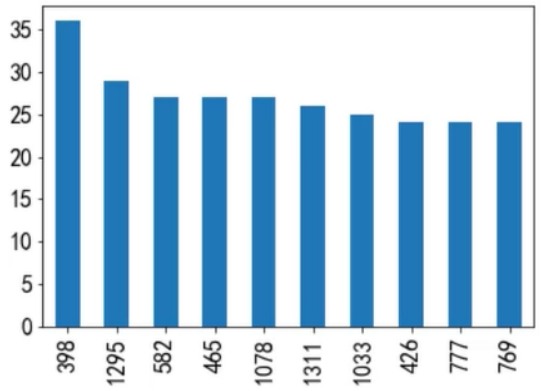
点菜的种类

```
0 323
1 269
2 239
3 216
4 189
5 188
6 187
7 186
8 178
9 173
```

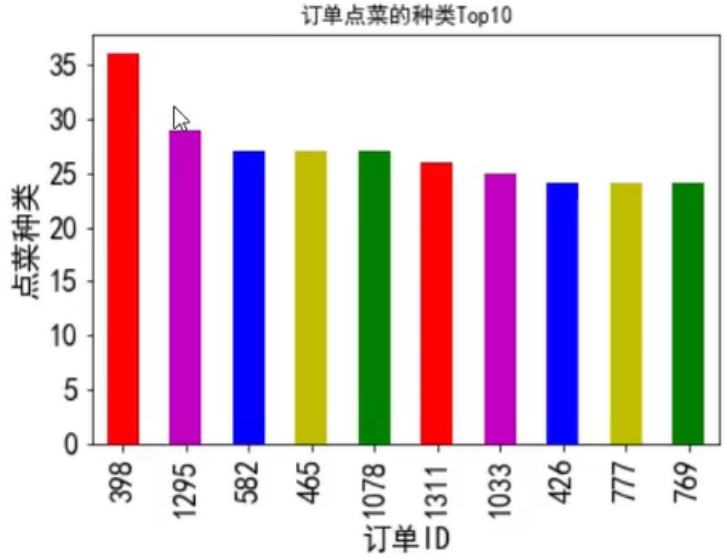


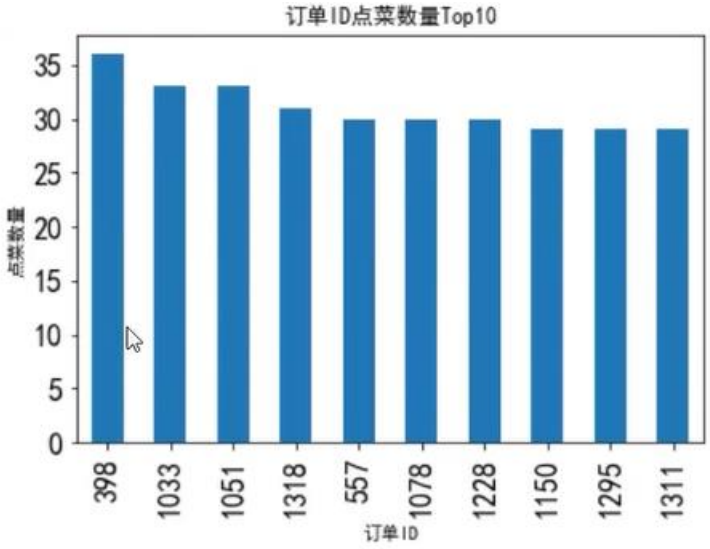
点菜的数量，消费金额最大以及平均消费金额最大等问题

```
[7]: <matplotlib.axes._subplots.AxesSubplot at 0x1dff91ffa88>
```




```
[12]: Text(0, 0.5, '点菜种类')
```



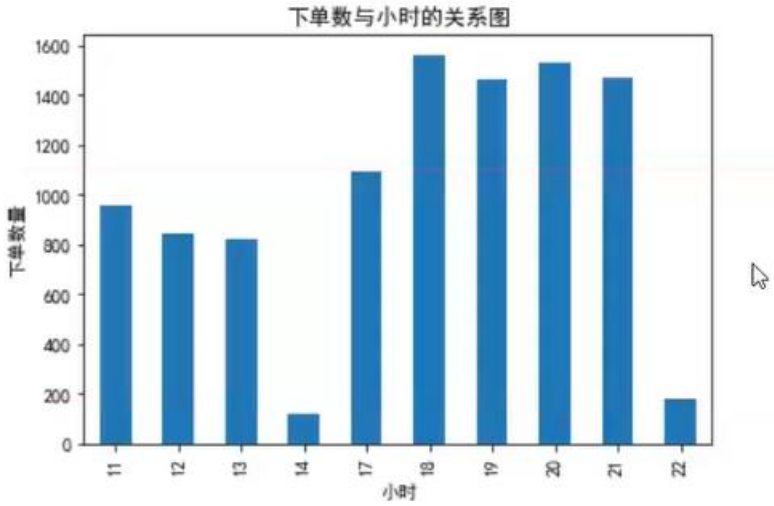


并且针对时间维度进行分析，分析了在一天中哪个时间段点菜量最多和一个月中点菜量最多的日期

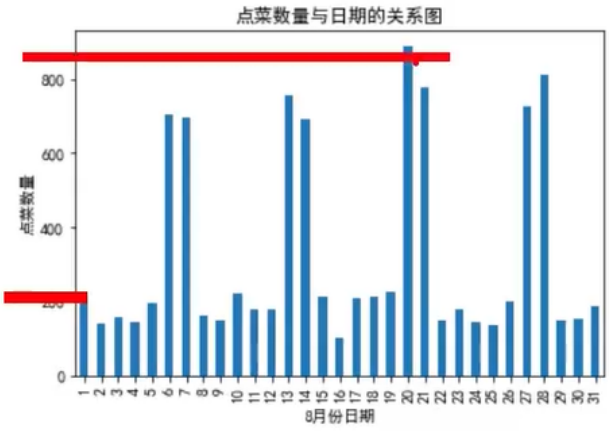
[9]:

	detail_id	order_id	dishes_id	dishes_name	itemis_add	counts	amounts	place_order_time	add_inprice	picture_file	emp_id	total_amounts	hourcount	time	hour
0	2956	417	610062	蒜香生蚝	0	1	49	2016-08-01 11:05:36	0	caipu/104001.jpg	1442	49	1	2016-08-01 11:05:36	11
1	2958	417	609957	蒙古烤羊腿	0	1	48	2016-08-01 11:07:07	0	caipu/202003.jpg	1442	48	1	2016-08-01 11:07:07	11
2	2961	417	609950	大蒜苋菜	0	1	30	2016-08-01 11:07:40	0	caipu/303001.jpg	1442	30	1	2016-08-01 11:07:40	11
3	2966	417	610038	芝麻烤紫菜	0	1	25	2016-08-01 11:11:11	0	caipu/105002.jpg	1442	25	1	2016-08-01 11:11:11	11
4	2968	417	610003	蒜香包	0	1	13	2016-08-01 11:11:30	0	caipu/503002.jpg	1442	13	1	2016-08-01 11:11:30	11
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
3606	5683	672	610049	爆炒双丝	0	1	35	2016-08-31 21:53:30	0	caipu/301003.jpg	1089	35	1	2016-08-31 21:53:30	21
3607	5686	672	609959	小炒羊腰\r\n\r\n\r\n	0	1	36	2016-08-31 21:54:40	0	caipu/202005.jpg	1089	36	1	2016-08-31 21:54:40	21
3608	5379	647	610012	香菇鹌鹑蛋	0	1	39	2016-08-31 21:54:44	0	caipu/302001.jpg	1094	39	1	2016-08-31 21:54:44	21
3609	5380	647	610054	不加一滴油的酸奶蛋糕	0	1	7	2016-08-31 21:55:24	0	caipu/501003.jpg	1094	7	1	2016-08-31 21:55:24	21
3610	5688	672	609953	凉拌菠菜	0	1	27	2016-08-31 21:56:54	0	caipu/303004.jpg	1089	27	1	2016-08-31 21:56:54	21

[12]: Text(0.5, 1.0, '下单数与小时的关系图')



[16]: Text(0.5, 1.0, '点菜数量与日期的关系图')



技术点：拼接数据，pdconcat

分组进行统计求和

排序，切片 top10

绘制柱状图的走势与高度

源代码：
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
plt.rcParams['font.sans-serif']= 'SimHei'
%matplotlib inline
# 1.加载数据
data1 =
pd.read_excel('meal_order_detail.xlsx',sheet_name='meal_order_detail1
')
data2 =
pd.read_excel('meal_order_detail.xlsx',sheet_name='meal_order_detail2
')
data3 =
pd.read_excel('meal_order_detail.xlsx',sheet_name='meal_order_detail3
')
# 2.数据预处理（NA）等处理，分析数据
data = pd.concat([data1,data2,data3],axis=0)
data.info()
data.dropna(axis=1,inplace=True)
#统计卖出菜品的平均价格
round(data['amounts'].mean(),2)
#频数统计 什么菜最受欢迎（对菜名进行频数统计，排出前十名）
dishes_count = data['dishes_name'].value_counts()

# 3.数据可视化 matplotlib
dishes_count.plot(kind='bar')
for x,y in enumerate(dishes_count):
    print(x,y)
    plt.text(x,y,y,ha='center',fontsize=12)

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
plt.rcParams['font.sans-serif']= 'SimHei'
%matplotlib inline
data1 =
pd.read_excel('meal_order_detail.xlsx',sheet_name='meal_order_detail1
')
data2 =
pd.read_excel('meal_order_detail.xlsx',sheet_name='meal_order_detail2
```

```
')
data3 =
pd.read_excel('meal_order_detail.xlsx',sheet_name='meal_order_detail3
')
# 2.数据预处理（NA）等处理，分析数据
data = pd.concat([data1,data2,data3],axis=0)
data.info()
data.dropna(axis=1,inplace=True)
#4.订单点菜的种类最多
data_group= data['order_id'].value_counts()[:10]
data_group.plot(kind='bar',fontsize=16,color=['r','m','b','y','g'])
plt.title('订单点菜的种类 Top10')
plt.xlabel('订单 Id',fontsize=16)
plt.ylabel('点菜种类',fontsize=16)
#5.订单 id 点菜数量 top10
data['total_amounts'] =data['counts']*data['amounts']
dataGroup =
data[['order_id','counts','amounts','total_amounts']].groupby(by='ord
er_id')
Group_sum = dataGroup.sum()
sort_counts= Group_sum.sort_values(by='counts',ascending=False)
sort_counts['counts'][:10].plot(kind = 'bar', fontsize=16)
plt.xlabel('订单 Id',fontsize=16)
plt.ylabel('点菜数量',fontsize=16)
plt.title('订单 id 点菜数量 top10')
#6.消费金额前十名
sort_total_amounts=
Group_sum.sort_values(by='total_amounts',ascending=False)
sort_total_amounts['total_amounts'][:10].plot(kind = 'bar',
fontsize=16)
import matplotlib.pyplot as plt
plt.rcParams['font.sans-serif']= 'SimHei'
%matplotlib inline
data1 =
pd.read_excel('meal_order_detail.xlsx',sheet_name='meal_order_detail1
')
data2 =
pd.read_excel('meal_order_detail.xlsx',sheet_name='meal_order_detail2
')
data3 =
pd.read_excel('meal_order_detail.xlsx',sheet_name='meal_order_detail3
')
# 2.数据预处理（NA）等处理，分析数据
data = pd.concat([data1,data2,data3],axis=0)
```

```
data.info()
data.dropna(axis=1,inplace=True)
#7.一天中什么时间段点菜梁集中
data['hourcount']= 1
data['time']=pd.to_datatime(data['place_order_time'])
data['hour'] = data['time'].map(lambda x:x.hour)
gp_by_hour = data.groupby(by='hour').count()['hourcount']
gp_by_hour.plot(kind='bar')
plt.xlabel('小时')
plt.ylabel('下单数量')
plt.title('下单量与小时数的关系')
#8.哪一天订餐数量多
data['daycount']= 1
data['day'] = data['time'].map(lambda x:x.day)
gp_by_day = data.groupby(by='day').count()['daycount']
gp_by_day.plot(kind='bar')
plt.xlabel('号')
plt.ylabel('下单数量')
plt.title('下单量与日期的关系')
```