

In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

# 下面这三行代码是为了画图可以显示中文
from pylab import *
mpl.rcParams['font.sans-serif'] = ['SimHei']
mpl.rcParams['axes.unicode_minus'] = False
```

In [2]:

```
def type_is_same(puck_type, airport_type):
    # 判断飞机的到达 (或起飞) 类型是否与登机口的到达 (或起飞) 类型相同
    airport_type = airport_type.replace(' ', '')
    airport_type = airport_type.split(',')
    #airport_type = [s.split() for s in airport_type]
    #print('puck type is {}'.format(puck_type))
    #print('gate type is {}'.format(airport_type))
    if puck_type in airport_type:
        return True
    else:
        return False

def classify_airport(all_airports):
    # airport 是所有的登机口
    # classes: 字典用于存储每种类别的登机口
    classes = {0:[], 1:[], 2:[], 3:[], 4:[], 5:[], 6:[], 7:[], 8:[],
               9:[], 10:[], 11:[], 12:[], 13:[], 14:[], 15:[], 16:[], 17:[]}

    for airport in all_airports:
        if airport['body_type'] == 'N':
            if airport['a_type'] == 'D':
                if airport['d_type'] == 'D':    classes[0].append(airport)
                elif airport['d_type'] == 'I':    classes[1].append(airport)
                else:    classes[2].append(airport)
            elif airport['a_type'] == 'I':
                if airport['d_type'] == 'D':    classes[3].append(airport)
                elif airport['d_type'] == 'I':    classes[4].append(airport)
                else:    classes[5].append(airport)
            else:
                if airport['d_type'] == 'D':    classes[6].append(airport)
                elif airport['d_type'] == 'I':    classes[7].append(airport)
                else:    classes[8].append(airport)
        else:
            if airport['a_type'] == 'D':
                if airport['d_type'] == 'D':    classes[9].append(airport)
                elif airport['d_type'] == 'I':    classes[10].append(airport)
                else:    classes[11].append(airport)
            elif airport['a_type'] == 'I':
                if airport['d_type'] == 'D':    classes[12].append(airport)
                elif airport['d_type'] == 'I':    classes[13].append(airport)
                else:    classes[14].append(airport)
            else:
                if airport['d_type'] == 'D':    classes[15].append(airport)
                elif airport['d_type'] == 'I':    classes[16].append(airport)
                else:    classes[17].append(airport)

    return classes
```

In [3]:

```
def classify_puck(all_pucks):
    # all_pucks: 列表, 所有转场记录的飞机航班
    # puck_classes: 字典, 每种类别的转场记录飞机航班

    puck_classes = {0:[], 1:[], 2:[], 3:[],
                    4:[], 5:[], 6:[], 7:[]}
    for puck in all_pucks:
        if puck['plane_type'] == 'N':
            if puck['a_type'] == 'D':
                if puck['d_type'] == 'D':puck_classes[0].append(puck)
                else:puck_classes[1].append(puck)
            else:
                if puck['d_type'] == 'D': puck_classes[2].append(puck)
                else:puck_classes[3].append(puck)
        else:
            if puck['a_type'] == 'D':
                if puck['d_type'] == 'D':puck_classes[4].append(puck)
                else:puck_classes[5].append(puck)
            else:
                if puck['d_type'] == 'D': puck_classes[6].append(puck)
                else:puck_classes[7].append(puck)

    return puck_classes
```

In [4]:

```
def create_gates(gates):
    # puck_data : puck_dataframe类型, 包含全部登机口的信息
    # puck_data的形状是[num_gates, 6]
    # 返回: airports: 包含全部登机口的列表, 每一个元素是一个登机口
    airports = []
    for i in range(gates.shape[0]):
        gate_data = gates.loc[i, :]
        gate = {'gate':gate_data['登机口'], 'terminal':gate_data['终端厅'], 'region':gate_data['区域'], 'a_type':gate_data['到达类型'],
                'd_type':gate_data['出发类型'], 'body_type':gate_data['机体类别'],
                'puck_records':[], 'assign_flag':False}
        airports.append(gate)

    return airports

def create_pucks(pucks):
    # puck_data : puck_dataframe类型, 包含全部转场记录的信息
    # puck_data的形状是[num_pucks, 8]
    # 返回: allpucks: 包含全部转场记录的列表, 每一个元素是一个转场记录
    allpucks = []
    for i in range(pucks.shape[0]):
        puck_data = pucks.loc[i, :]
        puck = {'record':puck_data['飞机转场记录号'], 'arrive_time':puck_data['到达相对时间min'], 'a_flight':puck_data['到达航班'], 'a_type': puck_data['到达类型'],
                'plane_type':puck_data['飞机型号'], 'depart_time':puck_data['出发相对时间min'], 'de_flight':puck_data['出发航班'], 'd_type':puck_data['出发类型'],
                'airport':'', 'temporary': 0}

        allpucks.append(puck)

    return allpucks
```

In [5]:

```
def plane_type_map(plane_type):  
    Wide_body = ['332', '333', '33E', '33H', '33L', '773']  
    Narrow_body = ['319', '320', '321', '323', '325', '738', '73A', '73E', '73H', '73L']  
    plane_type = str(plane_type)  
    if (plane_type in Wide_body):  
        return 'W'  
    else:  
        return 'N'
```

In [6]:

```
def sort_pucks(puck_class):  
    # 此函数将同一类别的转场记录按照起飞时间的先后排序  
    # puck_class: 列表  
    # sort_puckclass: 排序好的转场记录, 按照起飞时间非递减排序  
    de_times = [puck['depart_time'] for puck in puck_class]  
    sort_index = np.argsort(de_times)  
    sort_puckclass = [puck_class[ind] for ind in sort_index]  
  
    return sort_puckclass
```

In [7]:

```
def greedysselector(sort_puck_class, airport):
    # sort_puck_class: 排序好的转场记录, 列表形式
    # airport: 一个登机口类实例
    start_times = [puck['arrive_time'] for puck in sort_puck_class]
    depart_times = [puck['depart_time'] for puck in sort_puck_class]

    j = 0
    while(sort_puck_class[j]['airport'] != ''):
        j = j + 1

    if airport['assign_flag'] == False: # 登机口没有被分配
        airport['busy_time'] = np.zeros(288)
        sp_ind = max(int(start_times[j]/5)-1, 0)
        ep_ind = int(depart_times[j]/5)
        sort_puck_class[j]['airport'] = airport['gate']
        airport['puck_records'].append(sort_puck_class[j]['record'])
        airport['busy_time'][sp_ind:ep_ind] = 1

    k = j
    for i in range(j+1, len(sort_puck_class)):
        if start_times[i]>=depart_times[k]:
            if sort_puck_class[i]['airport']=='': # 如果该转场记录没有被分配
                sort_puck_class[i]['airport'] = airport['gate']
                k = i
                airport['puck_records'].append(sort_puck_class[k]['record'])
                if start_times[i] == 0:
                    s_ind = 0
                else:
                    s_ind = int(start_times[i]/5)
                    e_ind = int(depart_times[i]/5)
                    airport['busy_time'][s_ind:e_ind] = 1

        airport['assign_flag'] = True
    else:
        for i in range(j+1, len(sort_puck_class)):
            if sort_puck_class[i]['airport']=='': # 如果该转场记录没有被分配
                puck_time = np.zeros(288)
                if start_times[i] == 0:
                    s_ind = 0
                else:
                    s_ind = int(start_times[i]/5)
                    e_ind = int(depart_times[i]/5)
                print('该记录的起止时间下标分别是{}和{}'.format(s_ind, e_ind))
                puck_time[s_ind:e_ind] = 1
                temp_time = puck_time + airport['busy_time']
                if np.max(temp_time)<=1:
                    print('可以安排插入航班.....')
                    airport['busy_time'] = temp_time
                    sort_puck_class[i]['airport'] = airport['gate']
                    airport['puck_records'].append(sort_puck_class[i]['record'])

        print('gates{} has assigned {}'.format(airport['gate'], airport['puck_record
s']))
    return sort_puck_class, airport
```

In [8]:

```
def assign_puck(puck_class, gate_class):
    if len(puck_class)==0 or len(gate_class)==0:
        return puck_class, gate_class
    sort_puck_class = sort_pucks(puck_class)
    puck = sort_puck_class[0]
    gate = gate_class[0]
    if puck['plane_type'] == gate['body_type']:
        if (type_is_same(puck['a_type'], gate['a_type']) & (type_is_same(puck['d
_type'], gate['d_type']))):
            #final_gates = []    #; final_pucks = []
            print('类型相匹配...')
            for i in range(len(gate_class)):
                puck_not_assign = [puck for puck in sort_puck_class if puck['air
port']==']
                if len(puck_not_assign)==0:
                    break
                sort_puck_class, airport = greedysselector(sort_puck_class, gate_
class[i])
                gate_class[i] = airport
                #final_pucks.extend(assign_pucks)
            return sort_puck_class, gate_class
        else:
            return sort_puck_class, gate_class
    else:
        return sort_puck_class, gate_class
```

In [9]:

```

gates = pd.read_csv('./gates (1).csv')
new_gates = gates[['登机口', '终端厅', '区域', '到达类型', '出发类型', '机体类别']]

puck_data = pd.read_csv('./puck_data.csv', encoding='gbk')
cols = ['飞机转场记录号', '到达相对时间min', '到达航班', '到达类型',
        '飞机型号', '出发相对时间min', '出发航班', '出发类型']
puck_data = puck_data[cols]

airports = create_gates(new_gates)
allpucks = create_pucks(puck_data)

puck_classes = classify_puck(allpucks)
gate_classes = classify_airport(airports)

single_type_gate = [0, 1, 3, 4, 9, 10, 12, 13]
multi_type_gate = [2, 5, 6, 7, 8, 11, 14, 15, 16, 17]
single_gate_classes = [gate_classes[code] for code in single_type_gate]
multi_gate_classes = [gate_classes[code] for code in multi_type_gate]

assign_pucks = []; assign_gates = []
for i in range(len(puck_classes)):
    as_puck, as_gate = assign_puck(puck_classes[i], single_gate_classes[i])
    assign_pucks.append(as_puck)
    assign_gates.append(as_gate)

for j in range(len(multi_type_gate)):
    for k in range(len(puck_classes)):

        am_puck, am_gate = assign_puck(assign_pucks[k], multi_gate_classes[j])
        ass_puck = [puck for puck in am_puck if puck['airport']!='']
        print('has assigned '+str(len(ass_puck)))
        print(len(am_puck))
        print('-----')
        assign_pucks[k] = am_puck
        multi_gate_classes[j] = am_gate
        #print(multi_gate_classes[j])
    print('=====')
    assign_gates.append(multi_gate_classes[j])

```

类型相匹配...

```
gatesT10 has assigned ['PK179', 'PK476', 'PK259', 'PK276', 'PK293',  
'PK325', 'PK355', 'PK396', 'PK422', 'PK440']  
gatesT11 has assigned ['PK173', 'PK473', 'PK267', 'PK284', 'PK309',  
'PK339', 'PK362', 'PK398', 'PK429']  
gatesT12 has assigned ['PK150', 'PK478', 'PK273', 'PK283', 'PK312',  
'PK348', 'PK388', 'PK415', 'PK438']  
gatesT13 has assigned ['PK188', 'PK477', 'PK266', 'PK288', 'PK323',  
'PK360', 'PK406', 'PK431']  
gatesT14 has assigned ['PK447', 'PK484', 'PK277', 'PK294', 'PK335',  
'PK380', 'PK413', 'PK437']  
gatesT15 has assigned ['PK180', 'PK482', 'PK279', 'PK346', 'PK414',  
'PK439']  
gatesT16 has assigned ['PK175', 'PK479', 'PK305', 'PK345', 'PK416']  
gatesT17 has assigned ['PK192', 'PK486', 'PK300', 'PK385', 'PK417']  
gatesT18 has assigned ['PK117', 'PK488', 'PK306', 'PK376', 'PK418']  
gatesT19 has assigned ['PK443', 'PK490', 'PK299', 'PK364', 'PK419']  
gatesS1 has assigned ['PK182', 'PK494', 'PK310', 'PK367', 'PK423']  
gatesS2 has assigned ['PK170', 'PK489', 'PK282', 'PK369', 'PK424']  
gatesS3 has assigned ['PK446', 'PK501', 'PK298', 'PK356', 'PK425']  
gatesS4 has assigned ['PK136', 'PK498', 'PK307', 'PK381', 'PK430']  
gatesS5 has assigned ['PK147', 'PK496', 'PK315', 'PK383', 'PK433']  
gatesS6 has assigned ['PK187', 'PK493', 'PK297', 'PK395', 'PK435']  
gatesS7 has assigned ['PK171', 'PK497', 'PK291', 'PK401']  
gatesS8 has assigned ['PK442', 'PK491', 'PK320', 'PK399']  
gatesS9 has assigned ['PK181', 'PK500', 'PK326', 'PK407']  
gatesS10 has assigned ['PK155', 'PK254', 'PK403']  
gatesS14 has assigned ['PK191', 'PK256', 'PK394']  
gatesS15 has assigned ['PK195', 'PK264', 'PK392']  
gatesS16 has assigned ['PK184', 'PK272', 'PK390']  
gatesS17 has assigned ['PK448', 'PK389']  
gatesS18 has assigned ['PK193', 'PK342']  
gatesS19 has assigned ['PK168', 'PK387']  
gatesS20 has assigned ['PK197', 'PK412']  
gatesS21 has assigned ['PK174', 'PK410']  
gatesS22 has assigned ['PK194', 'PK357']  
gatesS23 has assigned ['PK102', 'PK408']  
gatesS24 has assigned ['PK145', 'PK428']  
gatesS25 has assigned ['PK196']  
gatesS26 has assigned ['PK441']  
gatesS27 has assigned ['PK062']  
gatesS28 has assigned ['PK072']
```

类型相匹配...

```
gatesT1 has assigned ['PK149', 'PK470', 'PK260', 'PK287', 'PK324',  
'PK358', 'PK409']  
gatesS11 has assigned ['PK144', 'PK450', 'PK261', 'PK292', 'PK334',  
'PK361', 'PK421']  
gatesS12 has assigned ['PK465', 'PK274', 'PK303', 'PK344', 'PK386']  
gatesS13 has assigned ['PK455', 'PK314', 'PK371']
```

类型相匹配...

```
gatesT2 has assigned ['PK107', 'PK449', 'PK304', 'PK347']  
gatesT3 has assigned ['PK129', 'PK454', 'PK316', 'PK411']  
gatesT4 has assigned ['PK131', 'PK461', 'PK333', 'PK400']  
gatesT26 has assigned ['PK104', 'PK464', 'PK308']  
gatesT27 has assigned ['PK108', 'PK456', 'PK382']  
gatesT28 has assigned ['PK460', 'PK373']  
gatesS31 has assigned ['PK468', 'PK328']  
gatesS32 has assigned ['PK159', 'PK336']  
gatesS33 has assigned ['PK094', 'PK338']  
gatesS34 has assigned ['PK466', 'PK370']  
gatesS35 has assigned ['PK453', 'PK340']
```



```
gatesS36 has assigned ['PK089', 'PK349']
gatesS37 has assigned ['PK208', 'PK377']
gatesS38 has assigned ['PK463', 'PK363']
gatesS39 has assigned ['PK471']
gatesS40 has assigned ['PK458']
gatesS41 has assigned ['PK457']
类型相匹配...
gatesT20 has assigned ['PK480']
has assigned 145
145
```

类型相匹配...

该记录的起止时间下标分别是0和112
该记录的起止时间下标分别是0和116
该记录的起止时间下标分别是0和119
该记录的起止时间下标分别是0和122
该记录的起止时间下标分别是0和123
该记录的起止时间下标分别是101和123
该记录的起止时间下标分别是0和127
该记录的起止时间下标分别是107和129
该记录的起止时间下标分别是0和131
该记录的起止时间下标分别是0和132
该记录的起止时间下标分别是111和136
该记录的起止时间下标分别是0和137
该记录的起止时间下标分别是123和144
该记录的起止时间下标分别是125和146
该记录的起止时间下标分别是125和148
该记录的起止时间下标分别是123和150
该记录的起止时间下标分别是122和153
该记录的起止时间下标分别是126和156
该记录的起止时间下标分别是120和161
该记录的起止时间下标分别是132和167
该记录的起止时间下标分别是156和178
该记录的起止时间下标分别是116和180
该记录的起止时间下标分别是161和183
该记录的起止时间下标分别是166和188
该记录的起止时间下标分别是184和212
该记录的起止时间下标分别是177和215
该记录的起止时间下标分别是189和216
该记录的起止时间下标分别是199和221
该记录的起止时间下标分别是198和225
该记录的起止时间下标分别是203和227
该记录的起止时间下标分别是208和229
该记录的起止时间下标分别是196和231
该记录的起止时间下标分别是218和243
该记录的起止时间下标分别是227和249
该记录的起止时间下标分别是238和263
该记录的起止时间下标分别是254和273
该记录的起止时间下标分别是281和287
可以安排插入航班.....
该记录的起止时间下标分别是280和287
该记录的起止时间下标分别是265和287
该记录的起止时间下标分别是273和287
该记录的起止时间下标分别是255和287
该记录的起止时间下标分别是242和287
该记录的起止时间下标分别是278和287
该记录的起止时间下标分别是252和287
gatesT20 has assigned ['PK480', 'PK436']
gatesT21 has assigned ['PK156', 'PK481', 'PK271', 'PK313', 'PK350',
'PK402', 'PK434']
has assigned 8

45

has assigned 0

36

has assigned 22

28

has assigned 0

0

has assigned 0

5

has assigned 0

5

has assigned 39

39

=====

has assigned 145

145

has assigned 8

45

has assigned 0

36

has assigned 22

28

has assigned 0

0

has assigned 0

5

has assigned 0

5

has assigned 39

39

=====

类型相匹配...

has assigned 145

145

has assigned 8

45

类型相匹配...

gatesT8 has assigned ['PK459', 'PK495', 'PK278', 'PK317', 'PK353',
'PK404']

gatesT9 has assigned ['PK467', 'PK269', 'PK285', 'PK311', 'PK351',
'PK427']

has assigned 12

36

```
has assigned 22
28
```

```
-----
has assigned 0
0
```

```
-----
has assigned 0
5
```

```
-----
has assigned 0
5
```

```
-----
has assigned 39
39
```

```
=====
has assigned 145
145
```

```
-----
has assigned 8
45
```

```
-----
has assigned 12
36
```

```
-----
has assigned 22
28
```

```
-----
has assigned 0
0
```

```
-----
has assigned 0
5
```

```
-----
has assigned 0
5
```

```
-----
has assigned 39
39
```

```
=====
类型相匹配...
has assigned 145
145
```

```
-----
类型相匹配...
gatesT7 has assigned ['PK177', 'PK257', 'PK281', 'PK319', 'PK365',
'PK420']
gatesT22 has assigned ['PK165', 'PK262', 'PK286', 'PK332', 'PK378',
'PK426']
has assigned 20
45
```

```
-----
类型相匹配...
该记录的起止时间下标分别是0和116
该记录的起止时间下标分别是85和123
该记录的起止时间下标分别是0和125
该记录的起止时间下标分别是56和132
该记录的起止时间下标分别是72和134
该记录的起止时间下标分别是87和153
该记录的起止时间下标分别是132和156
```

该记录的起止时间下标分别是128和159
该记录的起止时间下标分别是118和165
该记录的起止时间下标分别是166和195
该记录的起止时间下标分别是173和197
该记录的起止时间下标分别是173和204
该记录的起止时间下标分别是191和213
该记录的起止时间下标分别是198和220
该记录的起止时间下标分别是191和240
该记录的起止时间下标分别是225和247
该记录的起止时间下标分别是221和261
该记录的起止时间下标分别是227和261
该记录的起止时间下标分别是240和262
该记录的起止时间下标分别是235和262
该记录的起止时间下标分别是231和264
该记录的起止时间下标分别是234和267
该记录的起止时间下标分别是246和287

gatesT7 has assigned ['PK177', 'PK257', 'PK281', 'PK319', 'PK365',
'PK420']

该记录的起止时间下标分别是0和116
该记录的起止时间下标分别是85和123
该记录的起止时间下标分别是0和125
该记录的起止时间下标分别是56和132
该记录的起止时间下标分别是72和134
该记录的起止时间下标分别是87和153
该记录的起止时间下标分别是132和156
该记录的起止时间下标分别是128和159
该记录的起止时间下标分别是118和165
该记录的起止时间下标分别是166和195
该记录的起止时间下标分别是173和197
该记录的起止时间下标分别是173和204
该记录的起止时间下标分别是191和213
该记录的起止时间下标分别是198和220
该记录的起止时间下标分别是191和240
该记录的起止时间下标分别是225和247
该记录的起止时间下标分别是221和261
该记录的起止时间下标分别是227和261
该记录的起止时间下标分别是240和262
该记录的起止时间下标分别是235和262
该记录的起止时间下标分别是231和264
该记录的起止时间下标分别是234和267
该记录的起止时间下标分别是246和287

gatesT22 has assigned ['PK165', 'PK262', 'PK286', 'PK332', 'PK378',
'PK426']

has assigned 12

36

类型相匹配...

该记录的起止时间下标分别是144和227
该记录的起止时间下标分别是188和231
该记录的起止时间下标分别是178和243
该记录的起止时间下标分别是210和287
该记录的起止时间下标分别是245和287

gatesT7 has assigned ['PK177', 'PK257', 'PK281', 'PK319', 'PK365',
'PK420']

该记录的起止时间下标分别是144和227
该记录的起止时间下标分别是188和231
该记录的起止时间下标分别是178和243
该记录的起止时间下标分别是210和287
该记录的起止时间下标分别是245和287

gatesT22 has assigned ['PK165', 'PK262', 'PK286', 'PK332', 'PK378',
'PK426']

```
has assigned 22
28
-----
has assigned 0
0
-----
has assigned 0
5
-----
has assigned 0
5
-----
has assigned 39
39
-----
=====
has assigned 145
145
-----
has assigned 20
45
-----
has assigned 12
36
-----
has assigned 22
28
-----
has assigned 0
0
-----
has assigned 0
5
-----
has assigned 0
5
-----
has assigned 39
39
-----
=====
has assigned 145
145
-----
has assigned 20
45
-----
has assigned 12
36
-----
has assigned 22
28
-----
has assigned 0
0
-----
has assigned 0
5
-----
类型相匹配...
gatesT5 has assigned ['PK452', 'PK483', 'PK331']
```

```
has assigned 3
5
-----
类型相匹配...
has assigned 39
39
-----
=====
has assigned 145
145
-----
has assigned 20
45
-----
has assigned 12
36
-----
has assigned 22
28
-----
has assigned 0
0
-----
has assigned 0
5
-----
has assigned 3
5
-----
has assigned 39
39
-----
=====
has assigned 145
145
-----
has assigned 20
45
-----
has assigned 12
36
-----
has assigned 22
28
-----
has assigned 0
0
-----
类型相匹配...
gatesT25 has assigned ['PK106', 'PK487', 'PK280', 'PK352']
has assigned 4
5
-----
has assigned 3
5
-----
类型相匹配...
has assigned 39
39
-----
=====
```

```

has assigned 145
145
-----
has assigned 20
45
-----
has assigned 12
36
-----
has assigned 22
28
-----
has assigned 0
0
-----
类型相匹配...
gatesT6 has assigned ['PK445']
has assigned 5
5
-----
类型相匹配...
该记录的起止时间下标分别是234和269
可以安排插入航班.....
gatesT6 has assigned ['PK445', 'PK374']
gatesT23 has assigned ['PK151']
has assigned 5
5
-----
类型相匹配...
has assigned 39
39
-----
=====

```

In [10]:

```

assign_gates = [assign_gates[i] for i in range(len(assign_gates)) if len(assign_gates[i])>0]

```

In [11]:

```
gate_sum = 0
puck_sum = 0

final_assign_pucks = []
final_assign_gates = []

for i in range(len(assign_gates)):
    for j in range(len(assign_gates[i])):
        num_pucks = len(assign_gates[i][j]['puck_records'])
        puck_sum += num_pucks
        if num_pucks > 0:
            gate_sum += 1
            print(assign_gates[i][j]['puck_records'])
            final_assign_gates.append(assign_gates[i][j]['gate'])
            final_assign_pucks.append(assign_gates[i][j]['puck_records'])
```



```
['PK179', 'PK476', 'PK259', 'PK276', 'PK293', 'PK325', 'PK355', 'PK3
96', 'PK422', 'PK440']
['PK173', 'PK473', 'PK267', 'PK284', 'PK309', 'PK339', 'PK362', 'PK3
98', 'PK429']
['PK150', 'PK478', 'PK273', 'PK283', 'PK312', 'PK348', 'PK388', 'PK4
15', 'PK438']
['PK188', 'PK477', 'PK266', 'PK288', 'PK323', 'PK360', 'PK406', 'PK4
31']
['PK447', 'PK484', 'PK277', 'PK294', 'PK335', 'PK380', 'PK413', 'PK4
37']
['PK180', 'PK482', 'PK279', 'PK346', 'PK414', 'PK439']
['PK175', 'PK479', 'PK305', 'PK345', 'PK416']
['PK192', 'PK486', 'PK300', 'PK385', 'PK417']
['PK117', 'PK488', 'PK306', 'PK376', 'PK418']
['PK443', 'PK490', 'PK299', 'PK364', 'PK419']
['PK182', 'PK494', 'PK310', 'PK367', 'PK423']
['PK170', 'PK489', 'PK282', 'PK369', 'PK424']
['PK446', 'PK501', 'PK298', 'PK356', 'PK425']
['PK136', 'PK498', 'PK307', 'PK381', 'PK430']
['PK147', 'PK496', 'PK315', 'PK383', 'PK433']
['PK187', 'PK493', 'PK297', 'PK395', 'PK435']
['PK171', 'PK497', 'PK291', 'PK401']
['PK442', 'PK491', 'PK320', 'PK399']
['PK181', 'PK500', 'PK326', 'PK407']
['PK155', 'PK254', 'PK403']
['PK191', 'PK256', 'PK394']
['PK195', 'PK264', 'PK392']
['PK184', 'PK272', 'PK390']
['PK448', 'PK389']
['PK193', 'PK342']
['PK168', 'PK387']
['PK197', 'PK412']
['PK174', 'PK410']
['PK194', 'PK357']
['PK102', 'PK408']
['PK145', 'PK428']
['PK196']
['PK441']
['PK062']
['PK072']
['PK149', 'PK470', 'PK260', 'PK287', 'PK324', 'PK358', 'PK409']
['PK144', 'PK450', 'PK261', 'PK292', 'PK334', 'PK361', 'PK421']
['PK465', 'PK274', 'PK303', 'PK344', 'PK386']
['PK455', 'PK314', 'PK371']
['PK107', 'PK449', 'PK304', 'PK347']
['PK129', 'PK454', 'PK316', 'PK411']
['PK131', 'PK461', 'PK333', 'PK400']
['PK104', 'PK464', 'PK308']
['PK108', 'PK456', 'PK382']
['PK460', 'PK373']
['PK468', 'PK328']
['PK159', 'PK336']
['PK094', 'PK338']
['PK466', 'PK370']
['PK453', 'PK340']
['PK089', 'PK349']
['PK208', 'PK377']
['PK463', 'PK363']
['PK471']
['PK458']
['PK457']
```

```
['PK480', 'PK436']  
['PK156', 'PK481', 'PK271', 'PK313', 'PK350', 'PK402', 'PK434']  
['PK459', 'PK495', 'PK278', 'PK317', 'PK353', 'PK404']  
['PK467', 'PK269', 'PK285', 'PK311', 'PK351', 'PK427']  
['PK177', 'PK257', 'PK281', 'PK319', 'PK365', 'PK420']  
['PK165', 'PK262', 'PK286', 'PK332', 'PK378', 'PK426']  
['PK452', 'PK483', 'PK331']  
['PK106', 'PK487', 'PK280', 'PK352']  
['PK445', 'PK374']  
['PK151']
```

In [13]:

```
assign_dict = dict(zip(final_assign_gates, final_assign_pucks))
```

In [14]:

```
import csv
# 从字典写入csv文件

csvFile3 = open('问题一答案.csv','w', newline='')
writer2 = csv.writer(csvFile3)
for key in assign_dict:
    writer2.writerow([key, assign_dict[key]])
csvFile3.close()
```

Out[14]:

98

Out[14]:

89

Out[14]:

89

Out[14]:

80

Out[14]:

80

Out[14]:

62

Out[14]:

53

Out[14]:

53

Out[14]:

53

Out[14]:

53

Out[14]:

52

Out[14]:

52

Out[14]:

52

Out[14]:

52

Out[14]:

52

Out[14]:

52

Out[14]:

43

Out[14]:

43

Out[14]:

43

Out[14]:

35

Out[14]:

35

Out[14]:

35

Out[14]:

35

Out[14]:

26

Out[14]:

26

Out[14]:

26

Out[14]:

26

Out[14]:

26

Out[14]:

26

Out[14]:

26

Out[14]:

26

Out[14]:

15

Out[14]:

15

Out[14]:

15

Out[14]:

15

Out[14]:

70

Out[14]:

71

Out[14]:

53

Out[14]:

35

Out[14]:

43

Out[14]:

43

Out[14]:

43

Out[14]:

35

Out[14]:

35

Out[14]:

26

Out[14]:

26

Out[14]:

26

Out[14]:

26

Out[14]:

26

Out[14]:

26

Out[14]:

26

Out[14]:

26

Out[14]:

26

Out[14]:

15

Out[14]:

15

Out[14]:

15

Out[14]:

26

Out[14]:

71

Out[14]:

61

Out[14]:

61

Out[14]:

61

Out[14]:

62

Out[14]:

34

Out[14]:

44

Out[14]:

25

Out[14]:

15

In [15]:

```
gate_sum  
puck_sum
```

Out[15]:

66

Out[15]:

248

画图

In [16]:

```
num_assign_pucks = [len(pucks) for pucks in final_assign_pucks]
assign_dict = dict(zip(final_assign_gates, final_assign_pucks))
assign_dict1 = dict(zip(final_assign_gates, num_assign_pucks))
assigns = pd.DataFrame(assign_dict1, index=[0])
assigns = assigns.T
```

画出被使用的登机口安排的航班数量图

In [17]:

```
plt.figure(figsize=(20, 10))
x = list(assigns.index)
plt.bar(x, assigns[0]*2, facecolor='b')
plt.xlabel('登机口', fontsize=18)
plt.ylabel('登机口分配的总航班数量', fontsize=18)
plt.xticks(rotation = 90, fontsize=16)
plt.title('登机口航班分配情况', fontsize=18)
plt.yticks(fontsize=16)
plt.show()
```

Out[17]:

<Figure size 1440x720 with 0 Axes>

Out[17]:

<BarContainer object of 66 artists>

Out[17]:

Text(0.5,0,'登机口')

Out[17]:

Text(0,0.5,'登机口分配的总航班数量')

Out[17]:

```
([0,  
 1,  
 2,  
 3,  
 4,  
 5,  
 6,  
 7,  
 8,  
 9,  
10,  
11,  
12,  
13,  
14,  
15,  
16,  
17,  
18,  
19,  
20,  
21,  
22,  
23,  
24,  
25,  
26,  
27,  
28,  
29,  
30,  
31,  
32,  
33,  
34,  
35,  
36,  
37,  
38,  
39,  
40,  
41,  
42,  
43,  
44,  
45,  
46,  
47,  
48,  
49,  
50,  
51,  
52,  
53,  
54,  
55,  
56,  
57,  
58,
```

```

59,
60,
61,
62,
63,
64,
65],
<a list of 66 Text xticklabel objects>)

```

Out[17]:

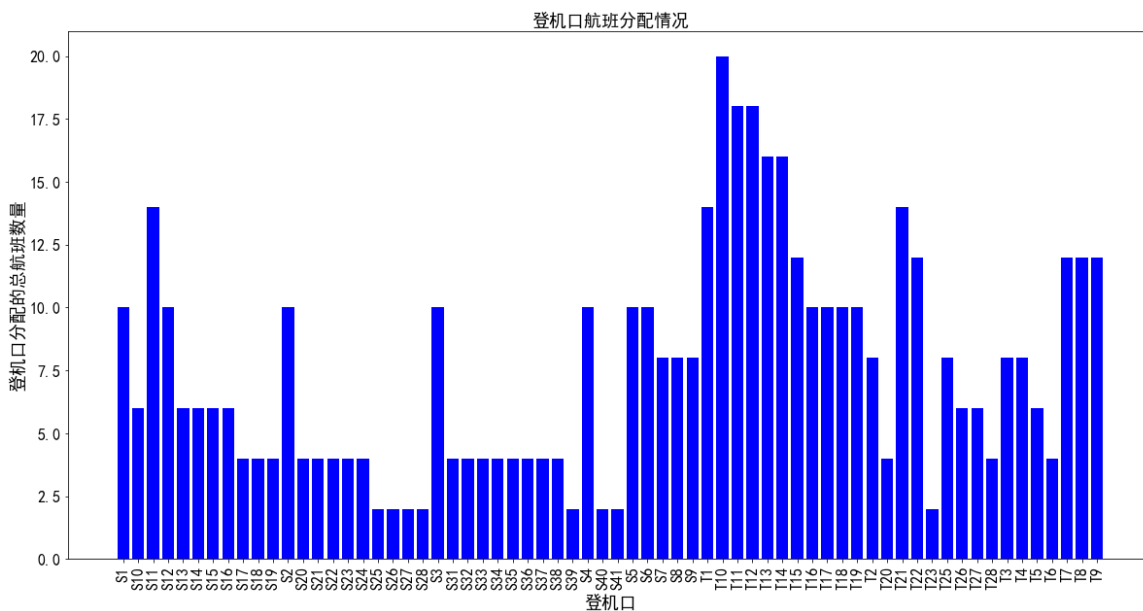
```
Text(0.5,1,'登机口航班分配情况')
```

Out[17]:

```

(array([ 0. ,  2.5,  5. ,  7.5, 10. , 12.5, 15. , 17.5, 20. , 22.
 5]),
<a list of 10 Text yticklabel objects>)

```



按照宽体机和窄体机画出登机口安排的航班数量

In [18]:

```

wide_gates = [airport['gate'] for airport in airports if airport['body_type']=='W']
narrow_gates = [airport['gate'] for airport in airports if airport['body_type']=='N']

```

In [19]:

```

narrow_assign_num = {}; wide_assign_num = {}
for gate in assign_dict.keys():
    if gate in wide_gates:
        #print(len(assign_dict[gate]))
        wide_assign_num[gate] = len(assign_dict[gate])
    else:
        #print('narrow'+str(len(assign_dict[gate])))
        narrow_assign_num[gate] = len(assign_dict[gate])

```

In [20]:

```
narrow_assign_num = pd.DataFrame(narrow_assign_num, index=[0]).T  
wide_assign_num = pd.DataFrame(wide_assign_num, index=[0]).T
```

In [21]:

```
plt.figure(figsize=(20, 10))
x = list(narrow_assign_num.index)
plt.bar(x, narrow_assign_num[0]*2, facecolor='b')
plt.xlabel('窄体机登机口', fontsize=18)
plt.ylabel('每个登机口分配航班数量', fontsize=18)
plt.xticks(rotation = 90, fontsize=16)
plt.yticks(fontsize=16)
plt.title('窄体登机口航班分配情况', fontsize=18)
plt.show()
```

Out[21]:

<Figure size 1440x720 with 0 Axes>

Out[21]:

<BarContainer object of 45 artists>

Out[21]:

Text(0.5,0,'窄体机登机口')

Out[21]:

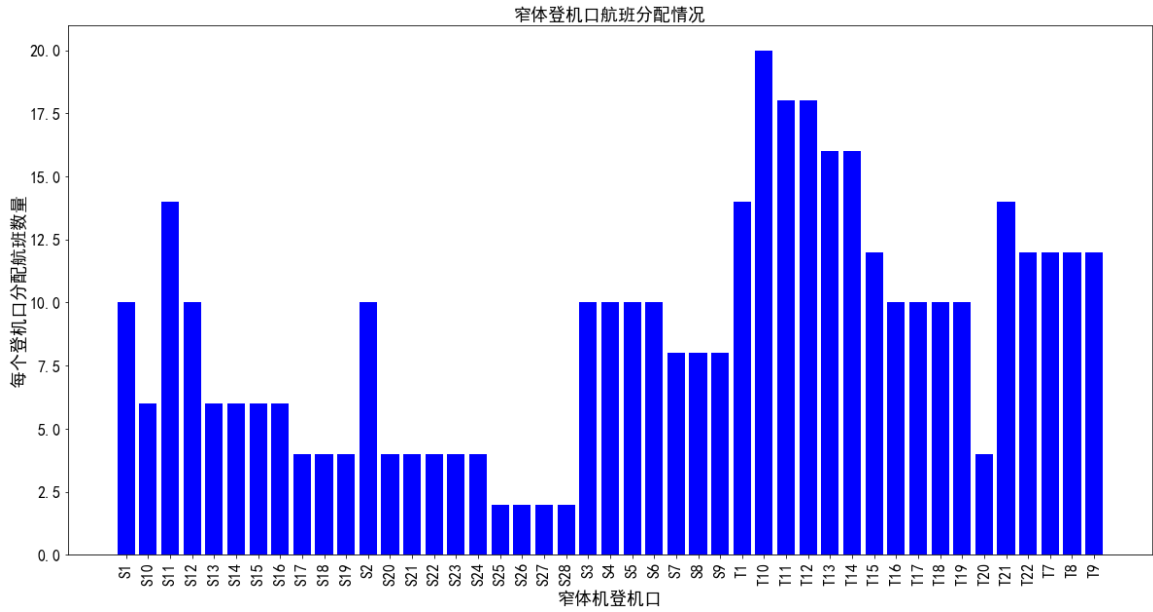
Text(0,0.5,'每个登机口分配航班数量')

Out[21]:

```
([0,
 1,
 2,
 3,
 4,
 5,
 6,
 7,
 8,
 9,
10,
11,
12,
13,
14,
15,
16,
17,
18,
19,
20,
21,
22,
23,
24,
25,
26,
27,
28,
29,
30,
31,
32,
33,
34,
35,
36,
37,
38,
39,
40,
41,
42,
43,
44],
<a list of 45 Text xticklabel objects>)
```

```
Out[21]:  
  
(array([ 0. ,  2.5,  5. ,  7.5, 10. , 12.5, 15. , 17.5, 20. , 22.  
5]),  
 <a list of 10 Text yticklabel objects>)
```

```
Out[21]:  
  
Text(0.5,1,'窄体登机口航班分配情况')
```



In [22]:

```
plt.figure(figsize=(20, 10))
x = list(wide_assign_num.index)
plt.bar(x, wide_assign_num[0]*2, facecolor='b')
plt.xlabel('宽体机登机口', fontsize=18)
plt.ylabel('每个登机口分配航班数量', fontsize=18)
plt.xticks(rotation = 90, fontsize=14)
plt.yticks(fontsize=14)
plt.title('宽体机登机口航班分配情况', fontsize=18)
plt.show()
```

Out[22]:

<Figure size 1440x720 with 0 Axes>

Out[22]:

<BarContainer object of 21 artists>

Out[22]:

Text(0.5,0,'宽体机登机口')

Out[22]:

Text(0,0.5,'每个登机口分配航班数量')

Out[22]:

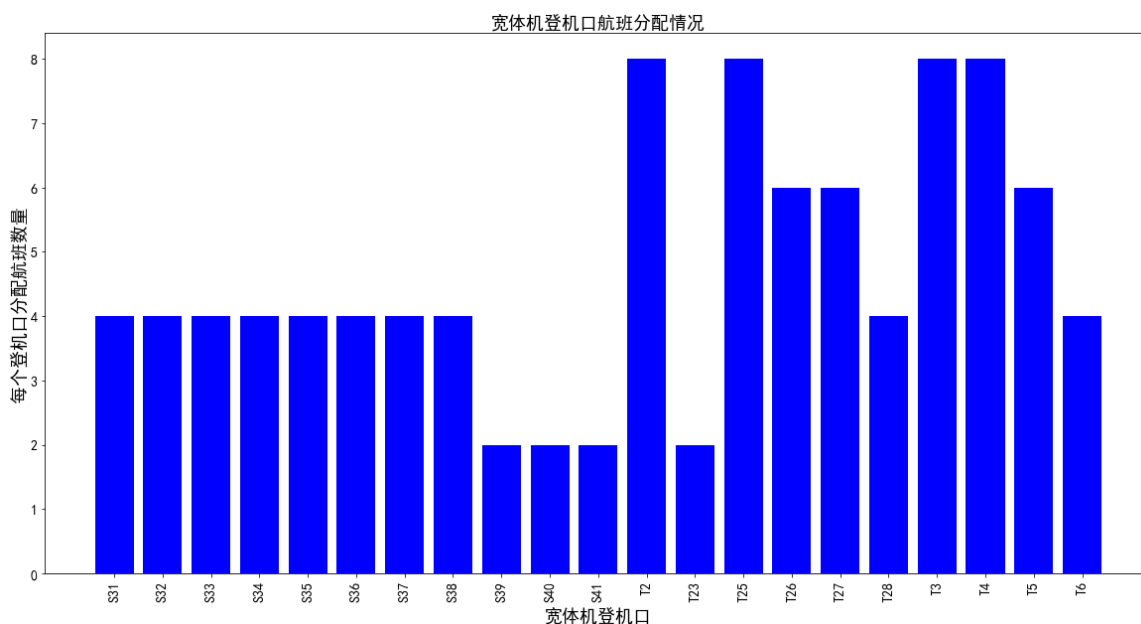
([0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20],
<a list of 21 Text xticklabel objects>)

Out[22]:

(array([0., 1., 2., 3., 4., 5., 6., 7., 8., 9.]),
<a list of 10 Text yticklabel objects>)

Out[22]:

Text(0.5,1,'宽体机登机口航班分配情况')



按照卫星厅和航站楼登机口画出登机口的使用数目和登机口的平均使用率

In [23]:

```
s_gates = [airport['gate'] for airport in airports if 'S' in airport['gate']]  
t_gates = [airport['gate'] for airport in airports if 'T' in airport['gate']]
```

In [24]:

```
s_gates_assign = {}; t_gates_assign = {}  
for gate in assign_dict.keys():  
    if gate in s_gates:  
        #print(len(assign_dict[gate]))  
        s_gates_assign[gate] = len(assign_dict[gate])  
    else:  
        #print('narrow'+str(len(assign_dict[gate])))  
        t_gates_assign[gate] = len(assign_dict[gate])
```

In [25]:

```
s_gates_assign = pd.DataFrame(s_gates_assign, index=[0]).T  
t_gates_assign = pd.DataFrame(t_gates_assign, index=[0]).T
```

In [26]:

```
plt.figure(figsize=(20, 10))
x = list(t_gates_assign.index)
plt.bar(x, t_gates_assign[0]*2, facecolor='b')
plt.xlabel('航站楼登机口', fontsize=18)
plt.ylabel('每个登机口分配航班数量', fontsize=18)
plt.xticks(rotation = 90, fontsize=16)
plt.yticks(fontsize=16)
plt.title('航站楼登机口航班分配情况', fontsize=18)
plt.show()
```

Out[26]:

<Figure size 1440x720 with 0 Axes>

Out[26]:

<BarContainer object of 27 artists>

Out[26]:

Text(0.5,0,'航站楼登机口')

Out[26]:

Text(0,0.5,'每个登机口分配航班数量')

Out[26]:

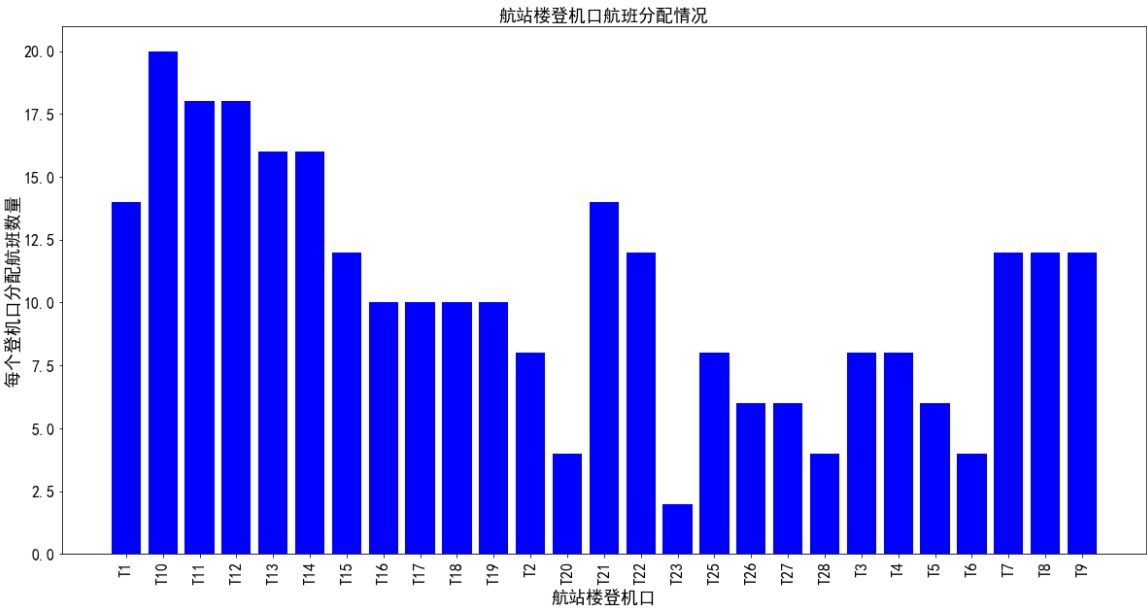
```
([0,
 1,
 2,
 3,
 4,
 5,
 6,
 7,
 8,
 9,
10,
11,
12,
13,
14,
15,
16,
17,
18,
19,
20,
21,
22,
23,
24,
25,
26],
<a list of 27 Text xticklabel objects>)
```

Out[26]:

```
(array([ 0. ,  2.5,  5. ,  7.5, 10. , 12.5, 15. , 17.5, 20. , 22.
 5]),
<a list of 10 Text yticklabel objects>)
```

Out[26]:

Text(0.5,1,'航站楼登机口航班分配情况')



In [27]:

```
plt.figure(figsize=(20, 10))
x = list(s_gates_assign.index)
plt.bar(x, s_gates_assign[0]*2, facecolor='b')
plt.xlabel('卫星厅登机口', fontsize=18)
plt.ylabel('每个登机口分配航班数量', fontsize=18)
plt.xticks(rotation = 90, fontsize=16)
plt.yticks(fontsize=16)
plt.title('卫星厅登机口航班分配情况', fontsize=18)
plt.show()
```

Out[27]:

<Figure size 1440x720 with 0 Axes>

Out[27]:

<BarContainer object of 39 artists>

Out[27]:

Text(0.5,0,'卫星厅登机口')

Out[27]:

Text(0,0.5,'每个登机口分配航班数量')

Out[27]:

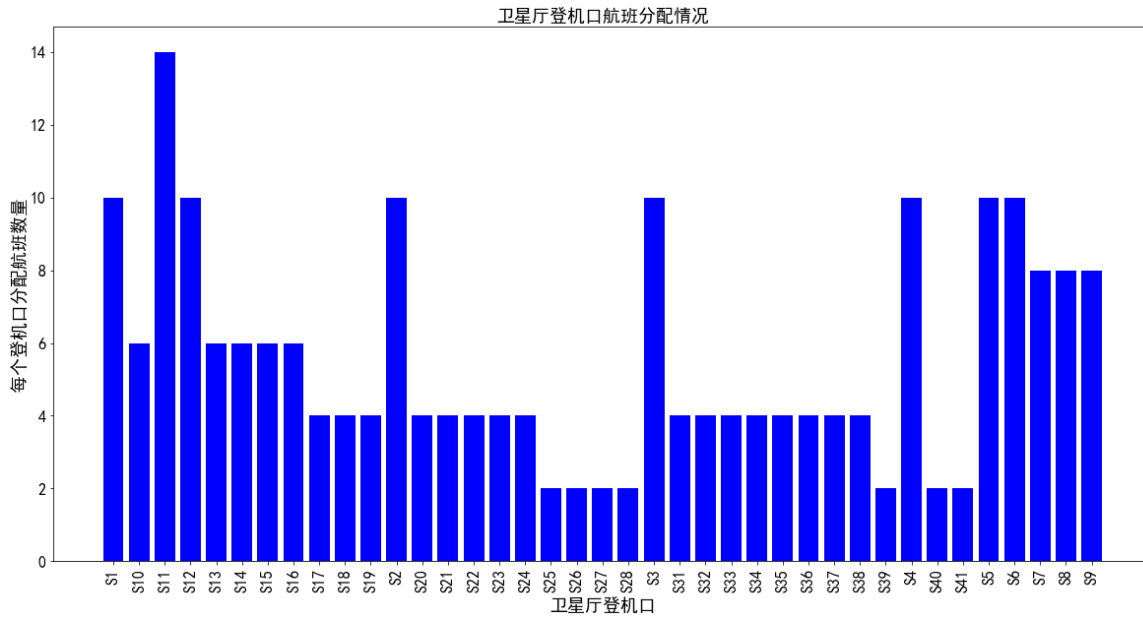
```
([0,
 1,
 2,
 3,
 4,
 5,
 6,
 7,
 8,
 9,
10,
11,
12,
13,
14,
15,
16,
17,
18,
19,
20,
21,
22,
23,
24,
25,
26,
27,
28,
29,
30,
31,
32,
33,
34,
35,
36,
37,
38],
<a list of 39 Text xticklabel objects>)
```

Out[27]:

```
(array([ 0.,  2.,  4.,  6.,  8., 10., 12., 14., 16.]),
<a list of 9 Text yticklabel objects>)
```

Out[27]:

Text(0.5,1,'卫星厅登机口航班分配情况')



In [28]:

```
s_airports = [airport for airport in airports if 'S' in airport['gate']]
t_airports = [airport for airport in airports if 'T' in airport['gate']]
```

In [29]:

```
s_busy_ratio = {}; t_busy_ratio = {}
assign_airport = list(assign_dict.keys())
for s_airport in s_airports:
    if s_airport['gate'] in assign_airport:
        all_time = len(s_airport['busy_time'])
        num_pucks = len(s_airport['puck_records'])
        busy_ratio = np.round((np.sum(s_airport['busy_time']) - 9*num_pucks) / all_time, 4)*100
        s_busy_ratio[s_airport['gate']] = busy_ratio

for t_airport in t_airports:
    if t_airport['gate'] in assign_airport:
        all_timet = len(t_airport['busy_time'])
        num_pucks = len(t_airport['puck_records'])
        busy_ratio = np.round((np.sum(t_airport['busy_time']) - 9*num_pucks) / all_timet, 4)*100
        t_busy_ratio[t_airport['gate']] = busy_ratio
```


In [30]:

```
s_busy_ratio = pd.DataFrame(s_busy_ratio, index=[0]).T  
t_busy_ratio = pd.DataFrame(t_busy_ratio, index=[0]).T
```

In [31]:

```
np.mean(s_busy_ratio)  
np.mean(t_busy_ratio)
```

Out[31]:

```
0    59.472564  
dtype: float64
```

Out[31]:

```
0    48.55963  
dtype: float64
```

In [32]:

```
plt.figure(figsize=(20, 10))
x = list(s_busy_ratio.index)
plt.bar(x, s_busy_ratio[0], facecolor='b')
plt.xlabel('卫星厅登机口', fontsize=18)
plt.ylabel('每个登机口使用率(%)', fontsize=18)
plt.xticks(rotation = 90, fontsize=16)
plt.yticks(fontsize=16)
plt.title('卫星厅登机口使用情况', fontsize=18)
plt.show()
```

Out[32]:

<Figure size 1440x720 with 0 Axes>

Out[32]:

<BarContainer object of 39 artists>

Out[32]:

Text(0.5,0,'卫星厅登机口')

Out[32]:

Text(0,0.5,'每个登机口使用率(%)')

Out[32]:

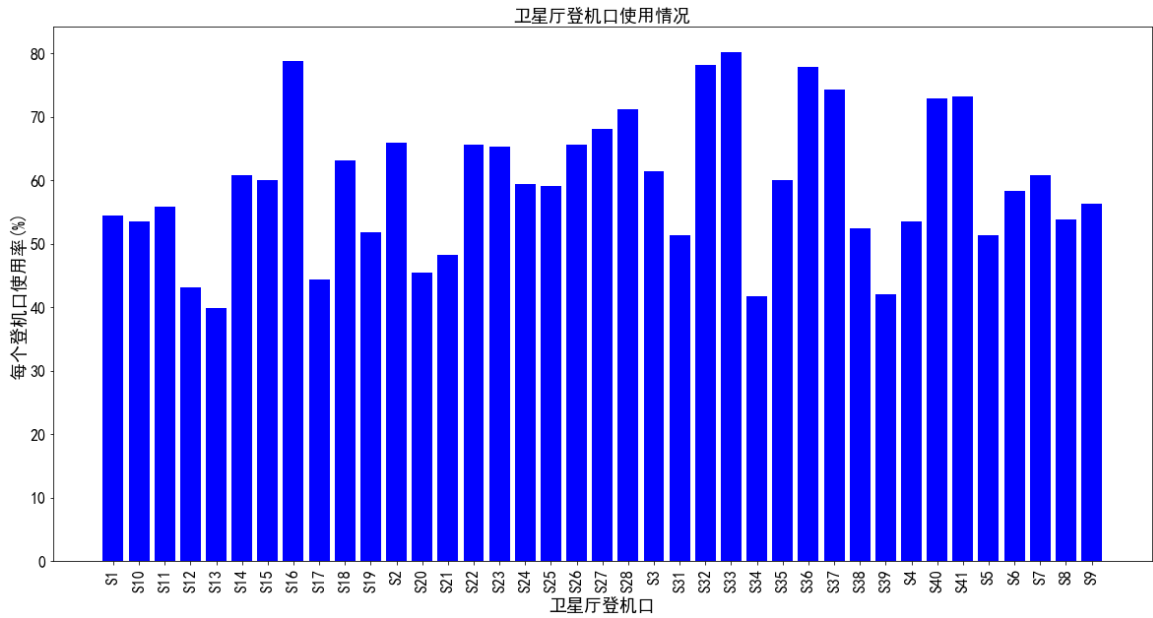
```
([0,
 1,
 2,
 3,
 4,
 5,
 6,
 7,
 8,
 9,
10,
11,
12,
13,
14,
15,
16,
17,
18,
19,
20,
21,
22,
23,
24,
25,
26,
27,
28,
29,
30,
31,
32,
33,
34,
35,
36,
37,
38],
<a list of 39 Text xticklabel objects>)
```

Out[32]:

```
(array([ 0., 10., 20., 30., 40., 50., 60., 70., 80., 90.]),
<a list of 10 Text yticklabel objects>)
```

Out[32]:

Text(0.5,1,'卫星厅登机口使用情况')



In [33]:

```
plt.figure(figsize=(20, 10))
x = list(t_busy_ratio.index)
plt.bar(x, t_busy_ratio[0], facecolor='b')
plt.xlabel('航站楼登机口', fontsize=18)
plt.ylabel('每个登机口使用率(%)', fontsize=18)
plt.xticks(rotation = 90, fontsize=16)
plt.yticks(fontsize=16)
plt.title('航站楼登机口使用情况', fontsize=18)
plt.show()
```

Out[33]:

<Figure size 1440x720 with 0 Axes>

Out[33]:

<BarContainer object of 27 artists>

Out[33]:

Text(0.5,0,'航站楼登机口')

Out[33]:

Text(0,0.5,'每个登机口使用率(%)')

Out[33]:

```
([0,
 1,
 2,
 3,
 4,
 5,
 6,
 7,
 8,
 9,
10,
11,
12,
13,
14,
15,
16,
17,
18,
19,
20,
21,
22,
23,
24,
25,
26],
<a list of 27 Text xticklabel objects>)
```

Out[33]:

```
(array([ 0., 10., 20., 30., 40., 50., 60., 70.]),
<a list of 8 Text yticklabel objects>)
```

Out[33]:

Text(0.5,1,'航站楼登机口使用情况')

