**Topic: A comparative analysis of Taxi data**

**(2018~2020, Mar~Sep)**

**General Notice**

The purpose of comparing and analyzing taxi data from 2018-2020, March to September is to see changes in taxi occupancy percentage and empty percentage after COVID-19. The results of the analysis show that the taxi occupancy percentage dropped after COVI-19 outbreak, but at some point it recovered.

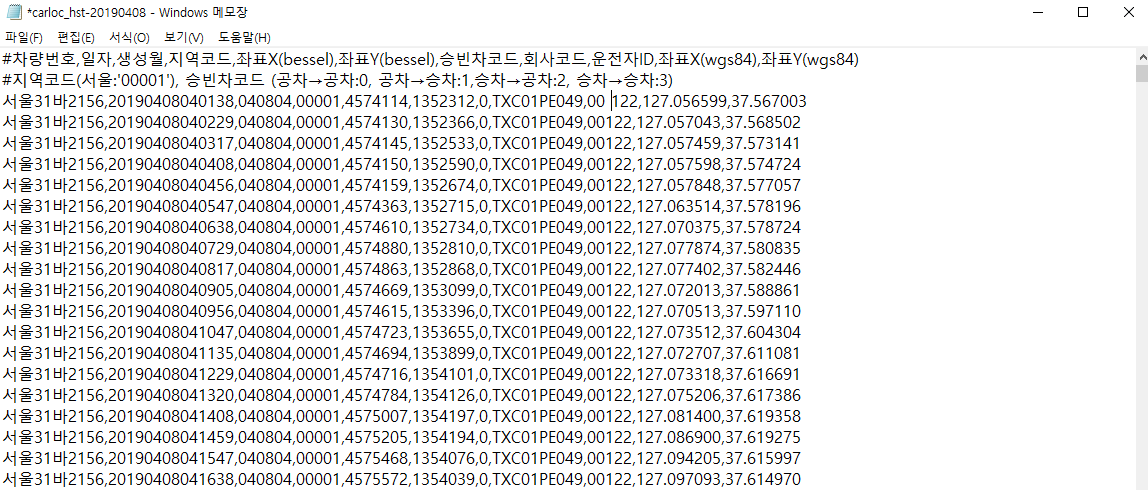
**Exercise 1: Data preprocessing**

Data preprocessing means that data is processed appropriatly prior to analysis. Here, R, a statistical analysis tool is used to preprocess data and Excel is used to make a graph by using data generated by R.

**Step 1. Use R**

* Required library: 1)dplyr, 2)sqldf, 3)stringr
* The index of R starts at 1.
* 2018,2019 can be analyzed as it is because there is only Seoul taxi data, but 2020 contains Seoul and Daegu taxi data, so only Seoul taxi data should be extracted and analyzed. (Seoul area code = 1)

First of all, the taxi data received from ‘NaviCall’ adds to a unique folder. Since there is 7 days of data in one zip file, unzip the zip files using R's unzip() function. The file is in \*.dat format, so use read.tabel() function to read. If you read the raw data, you can see the fields, [Car number, Date, Generated month, Area code, Coordinate, Occupancy/Empty code, Company code, Driver ID]. (Figure 1) The description of field is in the file.



Figure

Subsequently, calculate the number of each hourly occupancy/empty code by date, and save it as a csv file. (Figure 2) How to do this, first, extract the time using String function (substr()). And after creating a dataframe consisting of time (1~24h), count the number of occupancy/empty code by the time and fill the dataframe. Figure2 is the result, V1 means code 0(Empty), V2 means code 1(In), V3 means code 2(Out), and V4 means code 3(Occ).

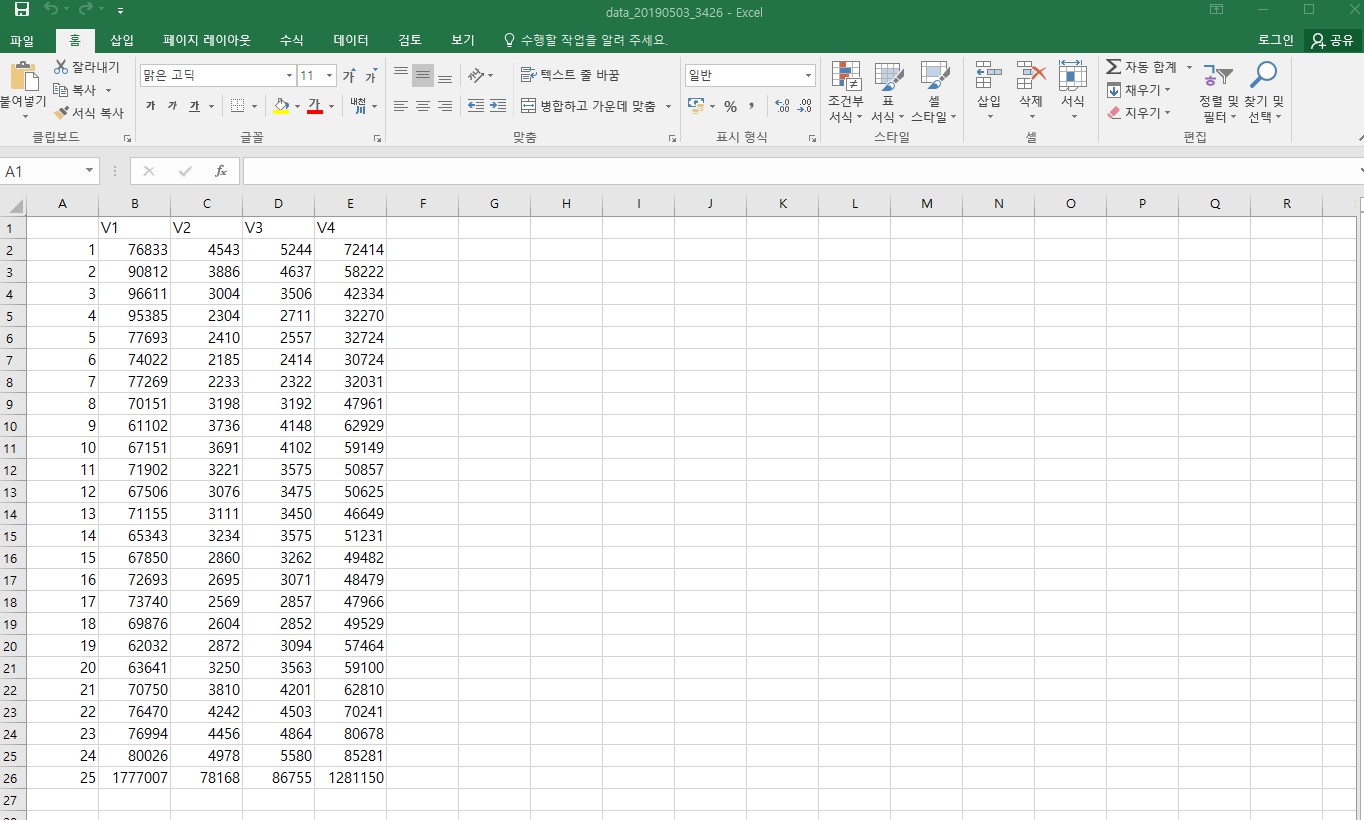


Figure 2

Occupancy/Empty code's hourly data is obtained by date, then calculate 1)the hourly occupancy/empty percentage by month and 2)the weekly occupancy/empty percentage by month. See Table 1 Expression, you can calculate the percentage using this expression. To calculate the hourly occupancy/empty percentage by month, add all the values of the files in the same month. (Figure 3) And to calculate the weekly (mon~sun) occupancy/empty percentage by month, you have to make a weekly dataframe, and extract the date. After you get the date, divided by 7 and obtain the remainder to find the day of the week and add all the values for each day of the week based on the month. (Figure 4)

|  |  |
| --- | --- |
| All\_occupancy (All\_Occ) | In + Out + Occupancy |
| Occupancy percentage (Occ%) | All\_occupancy/(Empty + All\_occupancy) |
| Empty percentage (Empty%) | Empty/(Empty +All\_occupancy) |

Table 1

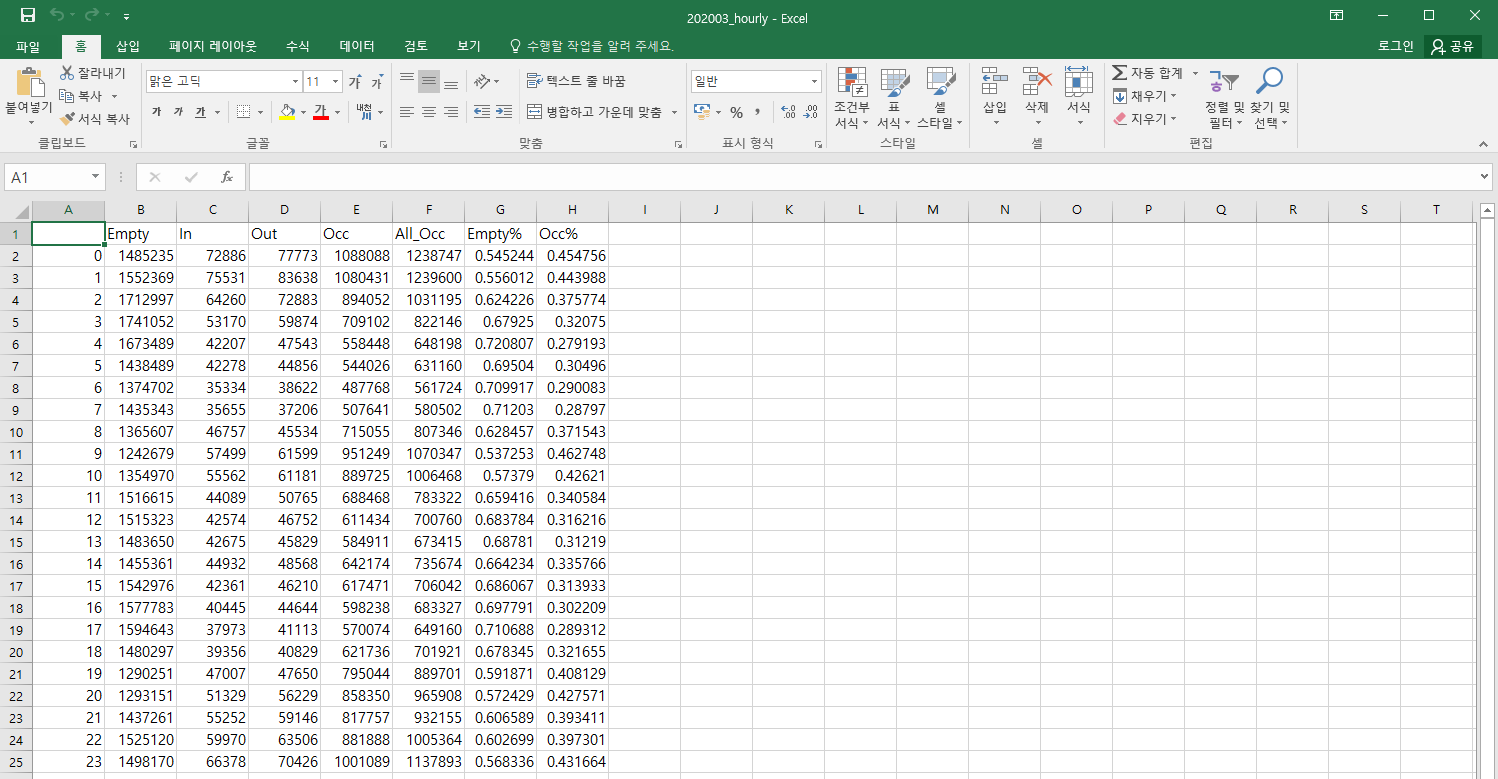


Figure 3

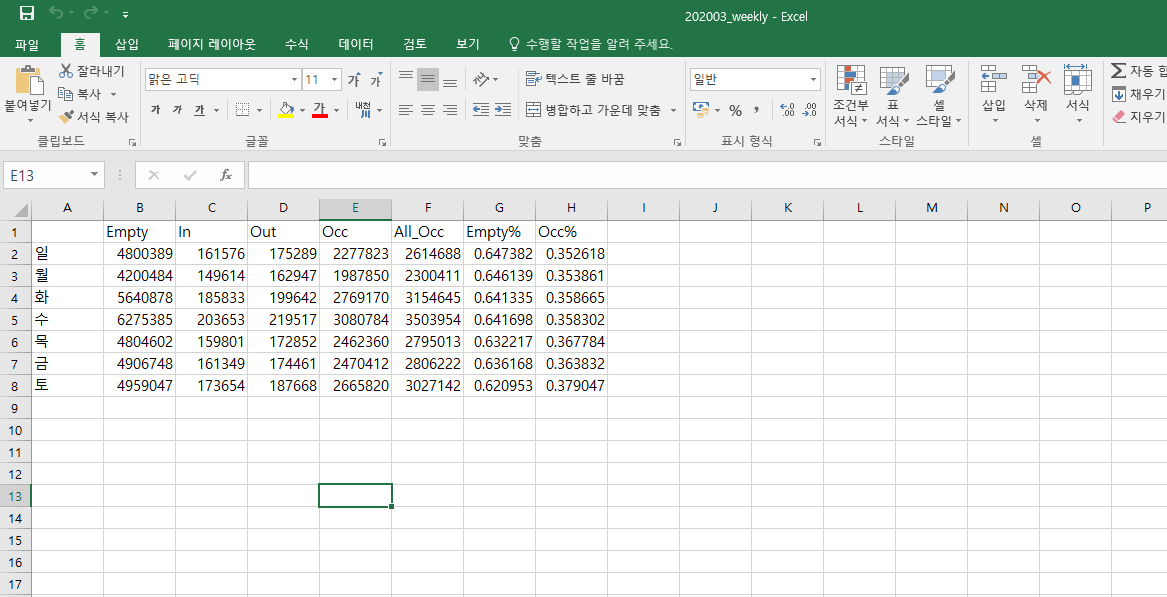


Figure 4

**Step 2. Use Excel**

And then make a occupancy/empty percentage graph by using excel and compare year by year based on the same month. Select the column that you want to make a graph (Hour/the day of week, Empty%, Occupancy%) and click [insert] → [chart]. There are so many kinds of charts, I chose the graph of broken line for the hourly graph and the vertical bar graph for the weekly graph. After inserting the graph, an option button appears next to it, which allows you to change the color of the graph or the order of the data.

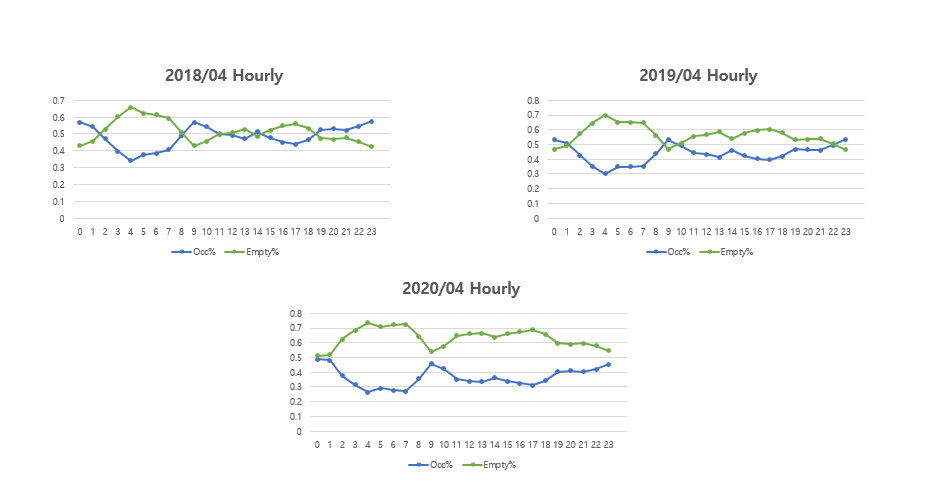


Figure 3

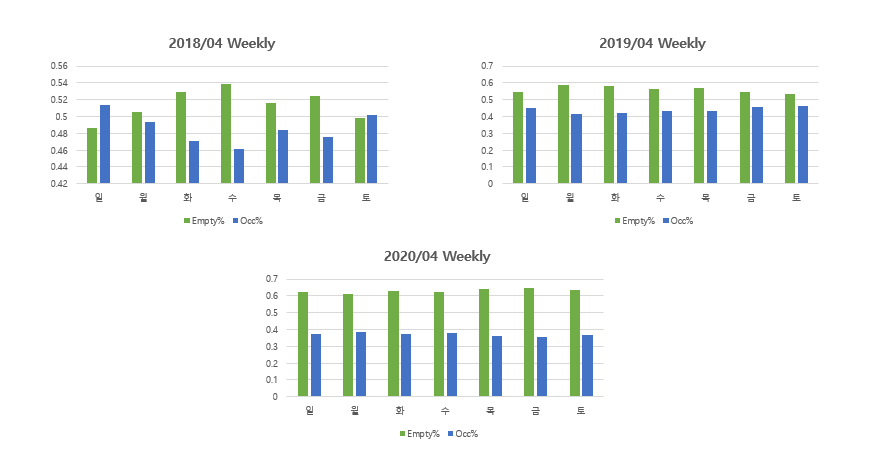


Figure 4

Finally, make the new Excel file to combine the data of 2018 and 2019’s average and the data of 2020 into a single file and create a graph in the same way as above. It is also made to two versions, hourly and weekly. (Figure 5, 6)

Figure 5

Figure 6

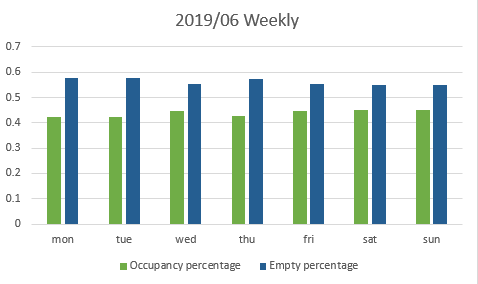
**Python**

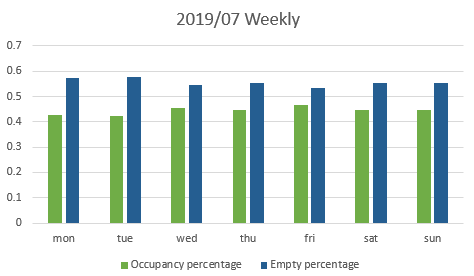
**1.Weekly divide**

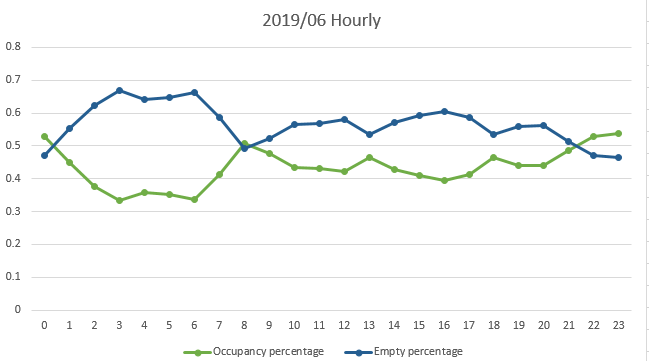
|  |
| --- |
| import pandas as pd  import os  import datetime  import zipfile  import zlib  import re  from tqdm import tqdm\_notebook  import glob  def func(order):  #파일 경로  zipped\_files = glob.glob("C:\\Users\\scsi\_public\\Desktop\intern\\A comparative analysis of Taxi data\\\*.zip")  #zip파일 안에 있는 csv를 하나로  def make\_total\_df(zipfile\_, file\_list):  total\_df=pd.DataFrame([], [])  for i in tqdm\_notebook(range(len(file\_list))):  # read csv file in zip file  df = pd.read\_csv(zipfile\_.open(file\_list[i]), encoding = 'cp949')  #concat dataframe  total\_df=pd.concat([total\_df, df])  return total\_df  allData = []  zf = zipped\_files[order]  KT\_2\_zip = zipfile.ZipFile(zf)  file\_list = zipfile.ZipFile.namelist(KT\_2\_zip)  total\_df=make\_total\_df(zipfile\_=KT\_2\_zip, file\_list=file\_list)  total\_df.drop(['#차량번호', '생성월', '지역코드', '좌표X(bessel)', '좌표Y(bessel)', '회사코드', '운전자ID', '좌표X(wgs84)', '좌표Y(wgs84)'], axis = 1, inplace = True)  total\_df = total\_df.drop(total\_df.index[0])  total\_df  def takeget(total\_df):  data\_0123 = []  for i in range(0,4):  data = total\_df[(total\_df['승빈차코드']==i)].copy()  data.drop(['승빈차코드'], axis = 1, inplace = True)  data.columns = [i]  data\_0123.append(data)  return data\_0123  data\_0123 = takeget(total\_df)  def getDay\_c(a,b,c):  daylist = [0, 1, 2, 3, 4, 5, 6]  return daylist[datetime.date(a,b,c).weekday()]    date\_0123 = [[0]\*7, [0]\*7, [0]\*7, [0]\*7]  import math  for i in range(0,4):  day\_count = [0]\*7  for j in range(1, len(data\_0123[i].index)):  x = str(math.trunc(int(data\_0123[i].iloc[j])))  y = int(x[0:4])  if y == 0 : break  m = int(x[4:6])  d = int(x[6:8])  day = getDay\_c(y,m,d)  day\_count[day] += 1  if j == len(data\_0123[i].index) - 1:  for k in range(len(day\_count)):  date\_0123[i][k] = day\_count[k]  return date\_0123  date\_0123 = [[0]\*7,[0]\*7, [0]\*7, [0]\*7]  while True:  order = int(input('몇번째 파일? : '))  new = func(order)  for i in range(0,4):  for j in range(0,7):  date\_0123[i][j] = date\_0123[i][j] + new[i][j]  YorN = input('끝인가요? : Y or N: ')  if YorN == 'Y' : break  Occ = [[0]\*7, [0]\*7, [0]\*7]  for i in range(0,7):  All\_Occ = date\_0123[1][i]+date\_0123[2][i]+date\_0123[3][i]  Occ[0][i] = All\_Occ  Occ[1][i] = All\_Occ/(date\_0123[0][i] + All\_Occ)  Occ[2][i] = date\_0123[0][i]/(date\_0123[0][i] + All\_Occ)  import pandas as pd  data = {'Empty' : date\_0123[0], 'In' : date\_0123[1], 'Out' : date\_0123[2], 'Occupancy' : date\_0123[3], 'All occupancy' : Occ[0], 'Occupancy percentage' : Occ[1] , 'Empty percentage' : Occ[2]}  df = pd.DataFrame(data, index = ['mon', 'tue', 'wed', 'thu', 'fri', 'sat', 'sun']) |

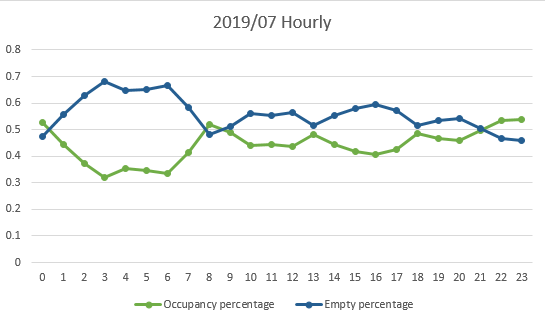
**2. Time divide**

|  |
| --- |
| import pandas as pd  import os  import zipfile  import zlib  import re  from tqdm import tqdm\_notebook  import glob  import math  def func(order):  zipped\_files = glob.glob("C:\\Users\\scsi\_public\\Desktop\intern\\A comparative analysis of Taxi data\\\*.zip")  def make\_total\_df(zipfile\_, file\_list):  total\_df=pd.DataFrame([], [])  for i in tqdm\_notebook(range(len(file\_list))):    df = pd.read\_csv(zipfile\_.open(file\_list[i]), encoding = 'cp949')  #concat dataframe  total\_df=pd.concat([total\_df, df])  return total\_df  allData = []  zf = zipped\_files[order]  KT\_2\_zip = zipfile.ZipFile(zf)  file\_list = zipfile.ZipFile.namelist(KT\_2\_zip)  total\_df=make\_total\_df(zipfile\_=KT\_2\_zip, file\_list=file\_list)  total\_df.drop(['#차량번호', '생성월', '지역코드', '좌표X(bessel)', '좌표Y(bessel)', '회사코드', '운전자ID', '좌표X(wgs84)', '좌표Y(wgs84)'], axis = 1, inplace = True)  total\_df = total\_df.drop(total\_df.index[0])  total\_df  def takeget(total\_df):  data\_0123 = []  for i in range(0,4):  data = total\_df[(total\_df['승빈차코드']==i)].copy()  data.drop(['승빈차코드'], axis = 1, inplace = True)  data.columns = [i]  data\_0123.append(data)  return data\_0123  data\_0123 = takeget(total\_df)    time\_0123 = [[0]\*24, [0]\*24, [0]\*24, [0]\*24]    for i in range(0,4):  time\_count = [0]\*24  for j in range(1, len(data\_0123[i].index)):  x = str(math.trunc(int(data\_0123[i].iloc[j])))  y = int(x[0:4])  if y == 0 : break  time = int(x[8:10])  time\_count[time] += 1  if j == len(data\_0123[i].index) - 1:  for k in range(len(time\_count)):  time\_0123[i][k] = time\_count[k]    return time\_0123  time\_0123 = [[0]\*24, [0]\*24, [0]\*24, [0]\*24]  while True:  order = int(input('몇번째 파일? : '))  new = func(order)  for i in range(0,4):  for j in range(0,24):  time\_0123[i][j] += new[i][j]  YorN = input('끝인가요? : Y or N: ')  if YorN == 'Y' : break  Occ = [[0]\*24, [0]\*24, [0]\*24]  for i in range(0,24):  All\_Occ = time\_0123[1][i]+time\_0123[2][i]+time\_0123[3][i]  Occ[0][i] = All\_Occ  Occ[1][i] = All\_Occ/(time\_0123[0][i] + All\_Occ)  Occ[2][i] = time\_0123[0][i]/(time\_0123[0][i] + All\_Occ)  import pandas as pd  data = {'Empty' : time\_0123[0], 'In' : time\_0123[1], 'Out' : time\_0123[2], 'Occupancy' : time\_0123[3], 'All occupancy' : Occ[0], 'Occupancy percentage' : Occ[1] , 'Empty percentage' : Occ[2]}  df = pd.DataFrame(data, index = ['0-1', '1-2', '2-3', '3-4', '4-5' ,'5-6', '6-7', '7-8', '8-9', '9-10', '10-11', '11-12', '12-13', '13-14', '14-15', '15-16', '16-17', '17-18', '18-19', '19-20', '20-21', '21-22', '22-23', '23-24'])  df.to\_csv("C:\\Users\\scsi\_public\\Desktop\\intern\\taxi\_occupancy\_time.csv", encoding = 'cp949') |

****

****

****

****