## code2:

In [1]:
```
import pandas as pd
import time,datetime
import matplotlib.pyplot as plt
```

In [2]:
```
df = pd.read_csv('D:/MONASH-y2-s2/assignment/3/q10.csv',header=None)
```

In [3]:
```
df.columns=['page_name','reaction_count','posted_at','time']
df
```

Out[3]:

| | page_name | reaction_count | posted_at | time |
|---|---|---|---|---|
| **0** | abc-news | 1149 | 1931/7/14 | 8:08 |
| **1** | abc-news | 1348 | 1931/7/14 | 10:48 |
| **2** | abc-news | 678 | 2006/8/14 | 9:24 |
| **3** | abc-news | 3484 | 2016/12/14 | 3:34 |
| **4** | abc-news | 2094 | 2016/6/15 | 15:55 |
| **...** | ... | ... | ... | ... |
| **289** | abc-news | 147 | 2025/10/16 | 11:59 |
| **290** | abc-news | 4837 | 2002/11/16 | 1:56 |
| **291** | abc-news | 2292 | 2002/11/16 | 11:48 |
| **292** | abc-news | 285 | 2003/11/16 | 11:09 |
| **293** | abc-news | 857 | 2006/11/16 | 12:02 |

294 rows × 4 columns

**explaination:** the code is to import libraries I need; read data generated by unix shell from this path and rename columns'name

## code3:

In [4]:
```
df.posted_at=pd.to_datetime(df.posted_at,format='%Y/%m/%d')
df['day']=df['posted_at'].dt.dayofweek
```

In [5]:
```
df.head()
```

Out[5]:

| | page_name | reaction_count | posted_at | time | day |
|---|---|---|---|---|---|
| **0** | abc-news | 1149 | 1931-07-14 | 8:08 | 1 |
| **1** | abc-news | 1348 | 1931-07-14 | 10:48 | 1 |
| **2** | abc-news | 678 | 2006-08-14 | 9:24 | 0 |
| **3** | abc-news | 3484 | 2016-12-14 | 3:34 | 2 |
| **4** | abc-news | 2094 | 2016-06-15 | 15:55 | 2 |

**explaination:** the code is to change datatype of the date and transform it into weekdays,then add one column to record it.

## code4:

In [6]:
```
df2 = df.groupby('day').agg({'reaction_count':'sum'})
df2=df2.reset_index()
df2
```

Out[6]:

| | day | reaction_count |
|---|---|---|
| **0** | 0 | 295463 |
| **1** | 1 | 94512 |
| **2** | 2 | 112856 |
| **3** | 3 | 151504 |
| **4** | 4 | 161929 |
| **5** | 5 | 179277 |
| **6** | 6 | 185965 |

**explaination:** group the data by "day" (group by weekdays) and sum the reaction_count

## code5:

In [7]:
```
def get_week_day(day):
    List=[]

    week_day_dict = {
            0 : 'Monday',
            1 : 'Tuesday',
            2 : 'Wednesday',
            3 : 'Thursday',
            4 : 'Friday',
            5 : 'Saturday',
            6 : 'Sunday',
      }
    for value in day:
        List.append(week_day_dict[value])

    daySer = pd.Series(List)
    return daySer
```

In [8]:
```
df2['week_day']=get_week_day(df2.day)
```

In [9]:
```
df2
```

Out[9]:

| | day | reaction_count | week_day |
|---|---|---|---|
| **0** | 0 | 295463 | Monday |
| **1** | 1 | 94512 | Tuesday |
| **2** | 2 | 112856 | Wednesday |
| **3** | 3 | 151504 | Thursday |
| **4** | 4 | 161929 | Friday |
| **5** | 5 | 179277 | Saturday |
| **6** | 6 | 185965 | Sunday |

**explaination:** the codes are to define a function that can transform the number in the "day" column into string.

Save weekdays as a column.

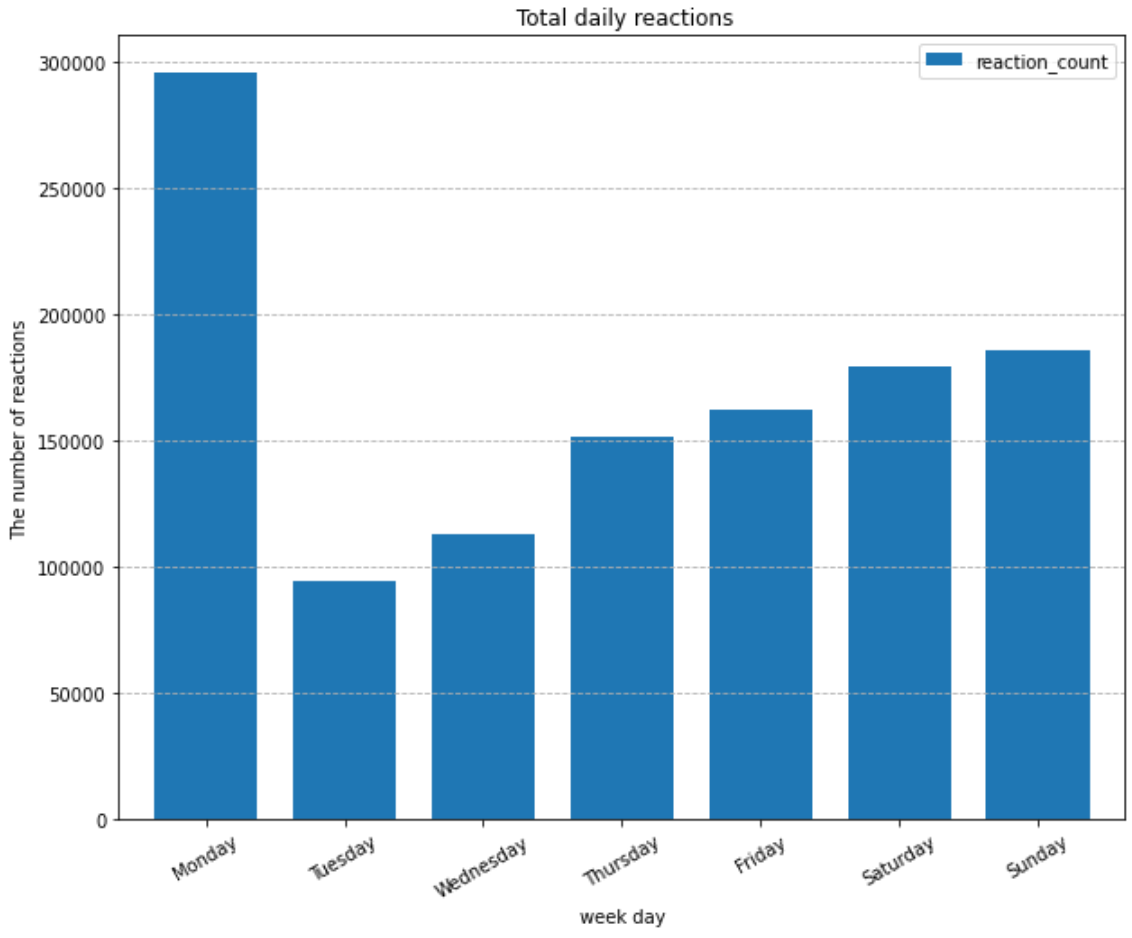
## code6:

In [10]:
```
df2=df2[['reaction_count','week_day']]
df2
```

Out[10]:

| | reaction_count | week_day |
|---|---|---|
| **0** | 295463 | Monday |
| **1** | 94512 | Tuesday |
| **2** | 112856 | Wednesday |
| **3** | 151504 | Thursday |
| **4** | 161929 | Friday |
| **5** | 179277 | Saturday |
| **6** | 185965 | Sunday |

In [11]:
```
chart = df2.plot.bar(figsize=(10,8),width = 0.75)
chart.set_xticklabels(df2['week_day'],rotation=30)
plt.xlabel('week day')
plt.ylabel('The number of reactions')
plt.title(' Total daily reactions')

plt.grid(axis='y',linestyle='--')
```




**explaination:** the code is to select a sub-dataset to draw a bar chart

# Q11

Monday and Sunday, users have shown the most reactions to the posts.

the trend is increasing, and the number of reactions during at weekends is generally more than that at the weekdays(expect Monday). Maybe that reason is people can get more time to play mobile phone and facebook when weekends coming.

# Q12

## code1:

In [12]:
```
dfm = df[(df.day==0)]
dfs = df[(df.day==6)]
dfm=dfm.reset_index()
dfs=dfs.reset_index()
```

**explaination:** the code is to filter these two day in which users have shown the most reactions to the posts, dataframes of them is represented by "dfs" and "dfm"

## code2:

In [13]:
```
def get_day_hour(time):
    List=[]

    for value in time:

        hour=value[:-3]

        List.append(hour)


    hourSer = pd.Series(List)
    return hourSer
```

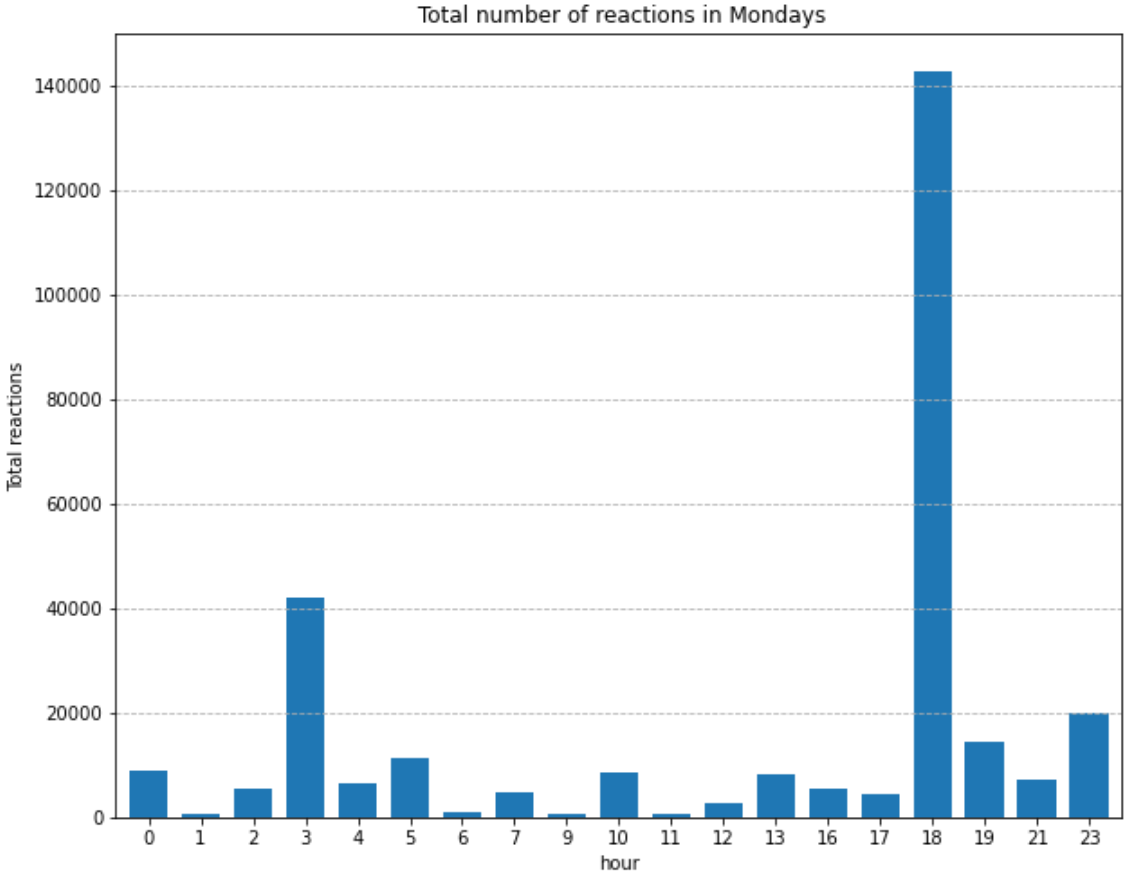
In [14]:
```
dfm['hour']=get_day_hour(dfm.time)
dfs['hour']=get_day_hour(dfs.time)
```

**explaination:** the code is a function that can extract the hour of post time for each post from the "time" column.

## code3:

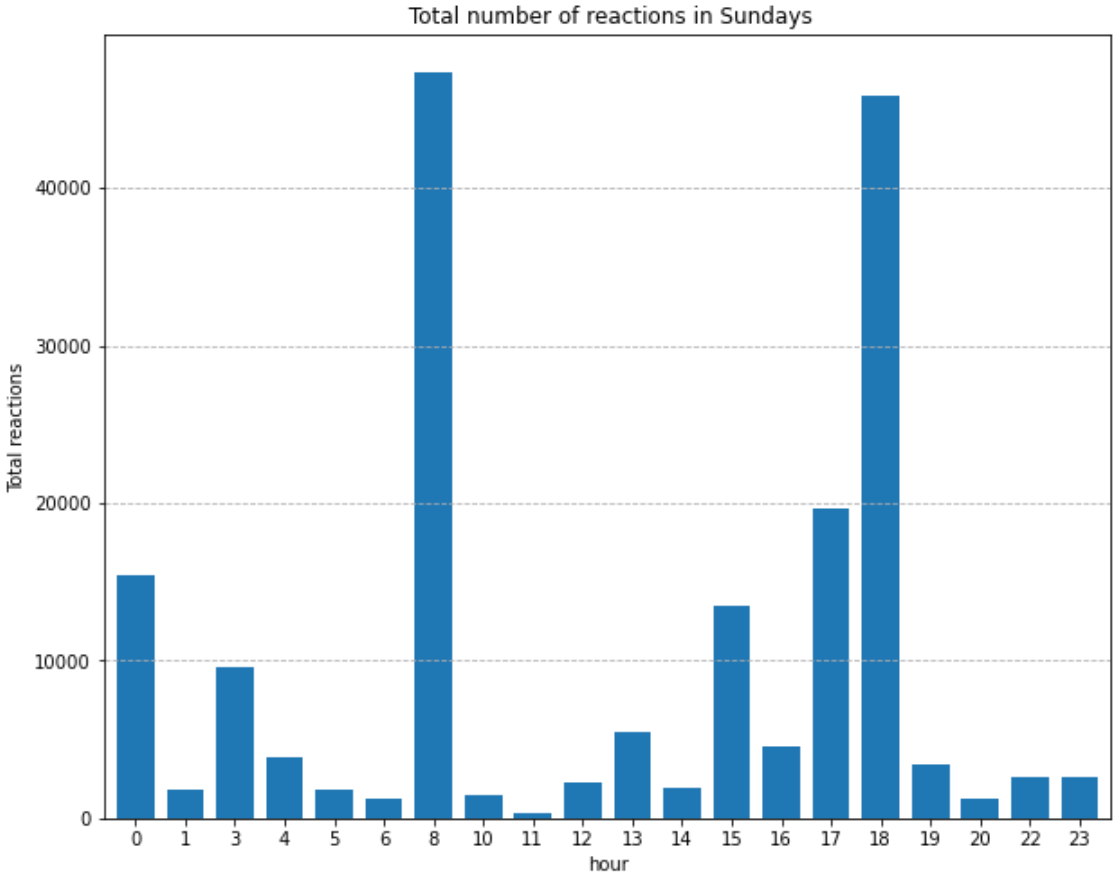
In [15]:
```
dfm.hour = dfm.hour.astype('int')
dfm=dfm.groupby('hour').agg({'reaction_count':'sum'}).reset_index().sort_values(by='hour')

dfs.hour = dfs.hour.astype('int')
dfs=dfs.groupby('hour').agg({'reaction_count':'sum'}).reset_index().sort_values(by='hour')
```

In [16]:
```
Mon = dfm.reaction_count.plot.bar(figsize=(10,8),width = 0.75)
Mon.set_xticklabels(dfm.hour,rotation=0)
plt.xlabel('hour')
plt.ylabel('Total reactions')
plt.title(' Total number of reactions in Mondays')

plt.grid(axis='y',linestyle='--')
```




In [ ]:
```
```

In [17]:
```
Sun = dfs.reaction_count.plot.bar(figsize=(10,8),width = 0.75)
Sun.set_xticklabels(dfs.hour,rotation=0)
plt.xlabel('hour')
plt.ylabel('Total reactions')
plt.title(' Total number of reactions in Sundays')

plt.grid(axis='y',linestyle='--')
```




**explaination:** the code firstly change "hour" column's data type into integer,and sum all reactions after grouping by hours. Then sort the dataframe against the value of hours.Finally, draw the bar chart ( hourly total reactions for each of two days)

Sundays: the most reactions happen around 8:00 am and 18 pm(8:00 am is the highest)
Mondays: the most reactions happen around 18 pm

There are some similarity between the number of hourly reactions in these two days. Firstly, these two days both have the highest bar at around 18:00pm. More, 8:00am and 18:00pm are the two highest in the respectively two days. Secondly, the general shapes(wave shape) of each days also are similar. For example 11 is the point at which both chart are lower than the rest hour, let's look at 3-4-5-6 hour, both have a generally decreasing trend.

# Q13

(a) 18:00pm Monday in which had the maximum number of reactions

(b) I think it's a good idea. I suggest publishing a general post about Trump in Mondays and Sundays and at the peak hours which is 8:00am and 18:00pm. because these two days and peak hours get the most people to look at facebook and to react.

The posts in these time period have some benefits: Being seen by more people; increases its exposure(It had a bigger impact); For the publisher, half the effort is twice the result