In [1]:
```
1 import numpy as np
2 import pandas as pd
```

In [2]:
```
1 df = pd.read_csv('out.csv')
```

In [3]:
```
1 len(df)
```

Out[3]: 3258

In [4]:
```
1 df['Date']
```

Out[4]:
```
0       2019-06-09
1       2019-04-28
2       2019-05-18
3       2019-04-18
4       2019-04-08
5       2019-06-30
6       2019-04-23
7       2019-05-13
8       2019-05-15
9       2019-04-14
10      2019-06-01
11      2019-04-18
12      2019-06-11
13      2019-05-29
14      2019-04-21
15      2019-06-06
16      2019-04-18
17      2019-05-16
18      2019-06-11
19      2019-05-07
20      2019-05-01
21      2019-04-27
22      2019-05-11
23      2019-04-25
24      2019-05-26
25      2019-06-21
26      2019-04-10
27      2019-05-09
28      2019-04-22
29      2019-05-30
           ...
3228    2019-02-26
3229    2019-01-12
3230    2019-02-25
3231    2019-01-12
3232    2019-03-30
3233    2019-01-28
3234    2019-03-15
3235    2019-03-08
3236    2019-02-12
3237    2019-03-31
3238    2019-01-11
3239    2019-03-21
3240    2019-03-13
3241    2019-01-25
3242    2019-01-22
3243    2019-01-18
3244    2019-03-25
3245    2019-02-22
3246    2019-02-26
3247    2019-02-13
3248    2019-02-21
3249    2019-02-18
3250    2019-03-18
3251    2019-03-20
```

```
3252    2019-03-13
3253    2019-01-06
3254    2019-01-24
3255    2019-01-09
3256    2019-02-04
3257    2019-03-04
Name: Date, Length: 3258, dtype: object
```

In [5]:
```
df['Month'] = df['Date'].apply(lambda x: ((int(x[5:7])-1) * 2) + 1 if in
```

In [6]:
```
s = df.groupby('Date')[df.columns[6]].mean()
```

In [7]:
```
1 s
```

Out[7]:
```
Date
2019-01-01     986.000000
2019-01-02     878.833333
2019-01-03     918.111111
2019-01-04     894.055556
2019-01-05     857.055556
2019-01-06     870.000000
2019-01-07     932.666667
2019-01-08     962.388889
2019-01-09     961.277778
2019-01-10     989.833333
2019-01-11    1052.805556
2019-01-12     905.000000
2019-01-13     848.416667
2019-01-14    1002.111111
2019-01-15    1107.722222
2019-01-16    1000.055556
2019-01-17     966.833333
2019-01-18     941.500000
2019-01-19    1056.722222
2019-01-20     859.277778
2019-01-21     870.055556
2019-01-22     907.055556
2019-01-23     944.388889
2019-01-24     915.277778
2019-01-25     976.277778
2019-01-26    1039.055556
2019-01-27     876.722222
2019-01-28     893.111111
2019-01-29     951.111111
2019-01-30     932.638889
                  ...
2019-06-01     889.166667
2019-06-02     834.333333
2019-06-03     901.055556
2019-06-04     900.444444
2019-06-05     930.277778
2019-06-06     963.833333
2019-06-07     950.611111
2019-06-08     918.722222
2019-06-09     876.500000
2019-06-10     947.000000
2019-06-11    1014.777778
2019-06-12     930.555556
2019-06-13    1039.611111
2019-06-14     969.333333
2019-06-15     920.500000
2019-06-16     855.333333
2019-06-17     915.388889
2019-06-18     997.833333
2019-06-19     918.555556
2019-06-20     953.722222
2019-06-21     974.722222
2019-06-22     933.500000
2019-06-23     839.388889
```

```
2019-06-24     892.277778
2019-06-25     917.388889
2019-06-26     942.388889
2019-06-27     949.888889
2019-06-28     984.333333
2019-06-29     905.888889
2019-06-30    1032.638889
Name: Daily Mean Travel Time, Length: 181, dtype: float64
```

In [8]:
```
from matplotlib import pyplot
from statsmodels.tsa.seasonal import seasonal_decompose
series = np.array(s)
result = seasonal_decompose(series, model='additive', freq=30)
result.plot()
pyplot.show()
```

```
<Figure size 640x480 with 4 Axes>
```

In [9]:
```
from statsmodels.tsa.statespace.sarimax import SARIMAX
```

In [10]:
```
model = SARIMAX(series)
```

In [ ]:
```

```

In [12]:
```
my_order = (1, 1, 1)
my_seasonal_order = (1, 1, 1, 30)
# define model
model = SARIMAX(series, order=my_order, seasonal_order=my_seasonal_orde
```

In [13]:
```
model_fit = model.fit()
```

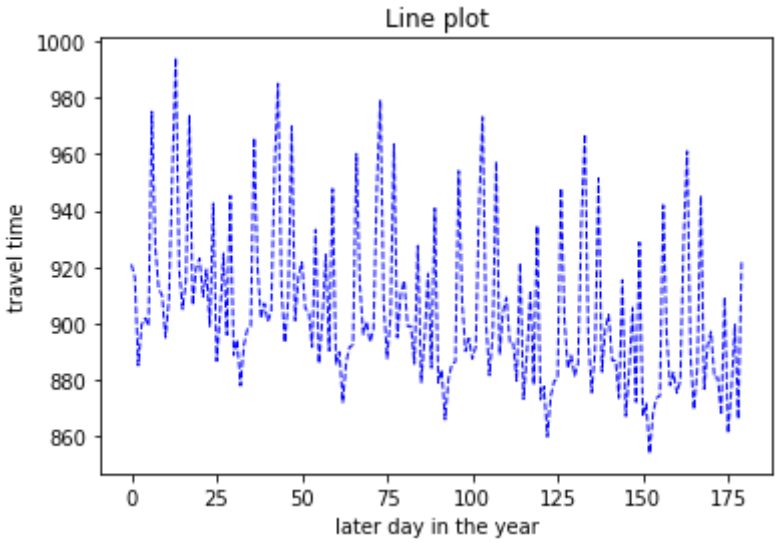
In [15]:
```
model_fit.forecast(30)
```

Out[15]:
```
array([920.91705845, 916.28836179, 885.14939783, 899.57061085,
       901.90686598, 899.12631338, 974.93419443, 927.31134257,
       912.21397452, 909.93269923, 894.97772401, 908.41120691,
       953.38037211, 994.03566588, 919.18029899, 905.00534153,
       915.06348209, 973.81905849, 906.50139119, 919.63613392,
       923.0519376 , 909.4036186 , 919.13181311, 899.00734835,
       942.48351242, 886.76403829, 903.22140407, 924.77919459,
       896.02470049, 945.22874323])
```

In [ ]:
```

```

In [24]:
```
1
2  import matplotlib.pyplot as plt
3
4  x = [i for i in range(180)]
5  y = model_fit.forecast(180)
6
7  plt.plot(x,y,"b--",linewidth=1)
8  plt.xlabel("later day in the year")
9  plt.ylabel("travel time")
10 plt.title("Line plot")
11 plt.show()
12 plt.savefig("line.jpg")
13
```




```
<Figure size 432x288 with 0 Axes>
```

In [ ]:
```
1
```