In [1]:
```python
import os
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

In [2]:
```python
data=pd.read_csv("Uberdata.csv")
data
```

Out[2]:

	dispatching_base_number	date	active_vehicles	trips
0	B02512	01-01-2015	190	1132
1	B02765	01-01-2015	225	1765
2	B02764	01-01-2015	3427	29421
3	B02682	01-01-2015	945	7679
4	B02617	01-01-2015	1228	9537
...	...	...	...	...
349	B02764	2/28/2015	3952	39812
350	B02617	2/28/2015	1372	14022
351	B02682	2/28/2015	1386	14472
352	B02512	2/28/2015	230	1803
353	B02765	2/28/2015	747	7753

354 rows × 4 columns

In [3]:
```python
from datetime import datetime

data["date"] = pd.to_datetime(data["date"])
data=data.set_index("date")
data.head()
```

Out[3]:

	dispatching_base_number	active_vehicles	trips
date			
2015-01-01	B02512	190	1132
2015-01-01	B02765	225	1765
2015-01-01	B02764	3427	29421
2015-01-01	B02682	945	7679
2015-01-01	B02617	1228	9537

In [4]:
```python
trips=data["trips"]
data
```

Out[4]:

	dispatching_base_number	active_vehicles	trips
date			
2015-01-01	B02512	190	1132
2015-01-01	B02765	225	1765
2015-01-01	B02764	3427	29421
2015-01-01	B02682	945	7679
2015-01-01	B02617	1228	9537
...	...	...	...
2015-02-28	B02764	3952	39812
2015-02-28	B02617	1372	14022
2015-02-28	B02682	1386	14472
2015-02-28	B02512	230	1803
2015-02-28	B02765	747	7753

354 rows × 3 columns

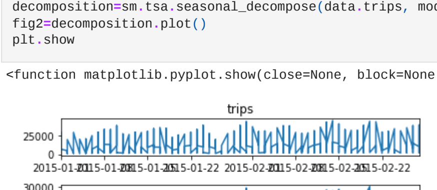
trips.plot(figsize=(10,6))

In [5]:
```python
from statsmodels.tsa.stattools import adfuller
test=adfuller(trips)
test[1]
```

Out[5]:
```
2.8427128542769045e-28
```

In [6]:
```python
import statsmodels.api as sm
decomposition=sm.tsa.seasonal_decompose(data.trips, model='additive',period=2)
fig2=decomposition.plot()
plt.show
```

Out[6]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```



In [7]:
```python
train=data[:283]
test=data[283:]
```

ARIMA

In [8]:
```python
from pmdarima import auto_arima
import warnings
warnings.filterwarnings("ignore")

from statsmodels.tsa.arima.model import ARIMA
```

In [9]:
```python
stepwise_fit=auto_arima(data.trips, trace=True, suppress_warnings=True)
stepwise_fit.summary()
```

```
Performing stepwise search to minimize aic
 ARIMA(2,1,2)(0,0,0)[0] intercept   : AIC=inf, Time=0.59 sec
 ARIMA(0,1,0)(0,0,0)[0] intercept   : AIC=7847.035, Time=0.03 sec
 ARIMA(1,1,0)(0,0,0)[0] intercept   : AIC=7725.542, Time=0.06 sec
 ARIMA(0,1,1)(0,0,0)[0] intercept   : AIC=inf, Time=0.36 sec
 ARIMA(0,1,0)(0,0,0)[0]             : AIC=7845.037, Time=0.03 sec
 ARIMA(2,1,0)(0,0,0)[0] intercept   : AIC=7669.827, Time=0.12 sec
 ARIMA(3,1,0)(0,0,0)[0] intercept   : AIC=7639.726, Time=0.17 sec
 ARIMA(4,1,0)(0,0,0)[0] intercept   : AIC=7625.915, Time=0.20 sec
 ARIMA(5,1,0)(0,0,0)[0] intercept   : AIC=7615.509, Time=0.23 sec
 ARIMA(5,1,1)(0,0,0)[0] intercept   : AIC=inf, Time=0.97 sec
 ARIMA(4,1,1)(0,0,0)[0] intercept   : AIC=inf, Time=1.06 sec
 ARIMA(5,1,0)(0,0,0)[0]             : AIC=7613.514, Time=0.19 sec
 ARIMA(4,1,0)(0,0,0)[0]             : AIC=7623.911, Time=0.16 sec
 ARIMA(5,1,1)(0,0,0)[0]             : AIC=7554.064, Time=0.48 sec
 ARIMA(4,1,1)(0,0,0)[0]             : AIC=7555.463, Time=0.38 sec
 ARIMA(5,1,2)(0,0,0)[0]             : AIC=7554.691, Time=0.76 sec
 ARIMA(4,1,2)(0,0,0)[0]             : AIC=7553.816, Time=0.97 sec
 ARIMA(3,1,2)(0,0,0)[0]             : AIC=7552.271, Time=0.64 sec
 ARIMA(2,1,2)(0,0,0)[0]             : AIC=7548.943, Time=0.69 sec
 ARIMA(1,1,2)(0,0,0)[0]             : AIC=7547.454, Time=0.44 sec
 ARIMA(0,1,2)(0,0,0)[0]             : AIC=7550.389, Time=0.20 sec
 ARIMA(1,1,1)(0,0,0)[0]             : AIC=7546.636, Time=0.56 sec
 ARIMA(0,1,1)(0,0,0)[0]             : AIC=7555.317, Time=0.28 sec
 ARIMA(1,1,0)(0,0,0)[0]             : AIC=7723.543, Time=0.05 sec
 ARIMA(2,1,1)(0,0,0)[0]             : AIC=7556.121, Time=0.25 sec
 ARIMA(2,1,0)(0,0,0)[0]             : AIC=7667.814, Time=0.08 sec
 ARIMA(1,1,1)(0,0,0)[0] intercept   : AIC=inf, Time=0.54 sec

Best model:  ARIMA(1,1,1)(0,0,0)[0]
Total fit time: 10.549 seconds
```

Out[9]:

SARIMAX Results

Dep. Variable:	y	No. Observations:	354
Model:	SARIMAX(1, 1, 1)	Log Likelihood	-3770.318
Date:	Tue, 15 Mar 2022	AIC	7546.636
Time:	12:31:58	BIC	7558.235
Sample:	0	HQIC	7551.251
	- 354		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	-0.1729	0.076	-2.269	0.023	-0.322	-0.024
ma.L1	-0.9823	0.013	-77.016	0.000	-1.007	-0.957
sigma2	1.1e+08	9.87e-11	1.11e+18	0.000	1.1e+08	1.1e+08

Ljung-Box (L1) (Q):	0.27	Jarque-Bera (JB):	116.89
Prob(Q):	0.60	Prob(JB):	0.00
Heteroskedasticity (H):	1.42	Skew:	1.31
Prob(H) (two-sided):	0.06	Kurtosis:	4.02

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).
[2] Covariance matrix is singular or near-singular, with condition number 1.03e+34. Standard errors may be unstable.

In [10]:
```python
modell =sm.tsa.arima.ARIMA(train.trips, order=(1,1,1)).fit()
print(modell.summary())
```

```
                               SARIMAX Results
==============================================================================
Dep. Variable:                  trips   No. Observations:                  283
Model:                 ARIMA(1, 1, 1)   Log Likelihood               -3009.855
Date:                Tue, 15 Mar 2022   AIC                           6025.710
Time:                        12:31:59   BIC                           6036.636
Sample:                             0   HQIC                          6030.092
                                - 283
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1         -0.1604      0.088     -1.814      0.070      -0.334       0.013
ma.L1         -0.9798      0.017    -57.287      0.000      -1.013      -0.946
sigma2      1.082e+08   8.11e-11   1.33e+18      0.000    1.08e+08    1.08e+08
===================================================================================
Ljung-Box (L1) (Q):                   0.20   Jarque-Bera (JB):                93.09
Prob(Q):                              0.65   Prob(JB):                         0.00
Heteroskedasticity (H):               1.45   Skew:                             1.31
Prob(H) (two-sided):                  0.07   Kurtosis:                         4.02
===================================================================================

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).
[2] Covariance matrix is singular or near-singular, with condition number 2.92e+34. Standard errors may be unstable.
```

In [11]:
```python
start=len(train.trips)
end=len(data.trips)-1
pred=modell.predict(start=len(train.trips),end=len(data.trips)-1)
pred.index=data.index[start:end+1]
pred
```

Out[11]:
```
date
2015-02-17    10218.647616
2015-02-17    13787.538412
2015-02-17    13215.201761
2015-02-17    13306.986369
2015-02-17    13292.267035
                  ...
2015-02-28    13294.301314
2015-02-28    13294.301314
2015-02-28    13294.301314
2015-02-28    13294.301314
2015-02-28    13294.301314
Name: predicted_mean, Length: 71, dtype: float64
```

In [12]:
```python
from sklearn.metrics import mean_squared_error
```

In [13]:
```python
error=np.sqrt(mean_squared_error(pred,test.trips))
error
```

Out[13]:
```
10967.75511066838
```

In [14]:
```python
data.trips.mean(),np.sqrt(test.var())
```

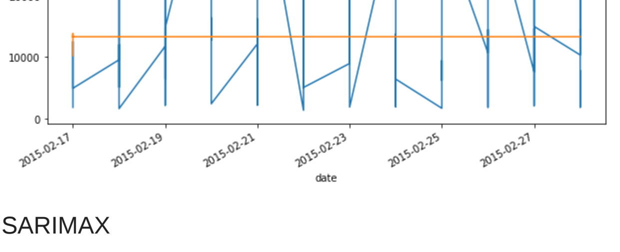
Out[14]:
```
(11667.316384180791,
 active_vehicles     1180.492327
 trips              11083.610744
 dtype: float64)
```

In [15]:
```python
forec=modell.forecast(150)
forec
```

Out[15]:
```
283    10218.647616
284    13787.538412
285    13215.201761
286    13306.986369
287    13292.267035
           ...
428    13294.301314
429    13294.301314
430    13294.301314
431    13294.301314
432    13294.301314
Name: predicted_mean, Length: 150, dtype: float64
```

In [16]:
```python
#train.plot(legend=True, label='Train')
test['trips'].plot(legend=True, label='Test',figsize=(10,6))
pred.plot(legend=True, label='ARIMA')
#forec.plot(legend=True, label='forecast')
```

Out[16]:
```
<AxesSubplot:xlabel='date'>
```



SARIMAX

In [17]:
```python
model2 = sm.tsa.statespace.SARIMAX(train.trips, order = (1,1,1),seasonal_order = (1,1,1,14))
results = model2.fit()
print(results.summary())
```

```
                                     SARIMAX Results
==========================================================================================
Dep. Variable:                              trips   No. Observations:                  283
Model:             SARIMAX(1, 1, 1)x(1, 1, 1, 14)   Log Likelihood               -2911.629
Date:                            Tue, 15 Mar 2022   AIC                           5833.258
Time:                                    12:32:00   BIC                           5851.213
Sample:                                         0   HQIC                          5840.470
                                            - 283
Covariance Type:                              opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1         -0.1566      0.167     -0.938      0.348      -0.484       0.171
ma.L1         -0.9617      0.063    -15.335      0.000      -1.085      -0.839
ar.S.L14      -0.1106      0.174     -0.635      0.525      -0.452       0.231
ma.S.L14      -0.7940      0.142     -5.610      0.000      -1.071      -0.517
sigma2      2.719e+08    2.2e-10   1.23e+18      0.000    2.72e+08    2.72e+08
===================================================================================
Ljung-Box (L1) (Q):                   0.07   Jarque-Bera (JB):                50.94
Prob(Q):                              0.78   Prob(JB):                         0.00
Heteroskedasticity (H):               1.39   Skew:                             1.00
Prob(H) (two-sided):                  0.12   Kurtosis:                         3.77
===================================================================================

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).
[2] Covariance matrix is singular or near-singular, with condition number 5.56e+33. Standard errors may be unstable.
```

In [18]:
```python
pred=results.predict(start=len(train),end=len(data.trips)-1)
pred.index=data.index[start:end+1]
pred
```

Out[18]:
```
date
2015-02-17     6972.942605
2015-02-17    13870.923353
2015-02-17    22899.874997
2015-02-17    13142.173989
2015-02-17    12497.570047
                  ...
2015-02-28    18992.315488
2015-02-28    16085.091120
2015-02-28    11778.838133
2015-02-28    25577.014662
2015-02-28    11541.236084
Name: predicted_mean, Length: 71, dtype: float64
```

In [19]:
```python
rmse=np.sqrt(mean_squared_error(test.trips,pred))
rmse
```

Out[19]:
```
11910.806477635628
```

In [20]:
```python
test['trips'].mean(),np.sqrt(test.var())
```

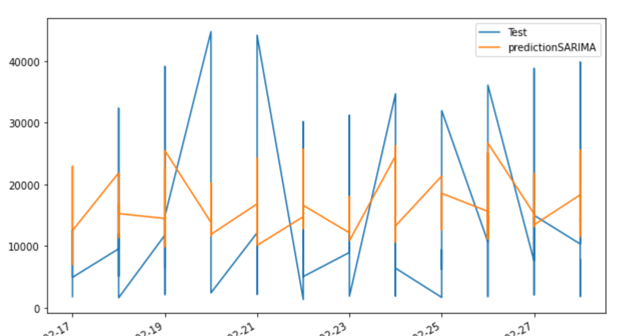
Out[20]:
```
(13421.0,
 active_vehicles     1180.492327
 trips              11083.610744
 dtype: float64)
```

In [21]:
```python
index_future_dates=pd.date_range(start='2019-01-01',end='2019-12-31')
fore=results.predict(start=1462,end=1826,type='levels')
fore.index=index_future_dates
fore.sum()
```

Out[21]:
```
22220529.513826527
```

In [22]:
```python
#train.plot(legend=True, label='Train')
test['trips'].plot(legend=True, label='Test',figsize=(10,6))
pred.plot(legend=True, label='predictionSARIMA')
#fore.plot(legend=True, label='forecastSARIMA',figsize=(10,6))
```

Out[22]:
```
<AxesSubplot:xlabel='date'>
```



HOLTWINTERS

In [23]:
```python
from statsmodels.tsa.holtwinters import ExponentialSmoothing
```

In [24]:
```python
hmodel=ExponentialSmoothing(train['trips'],trend='mul',seasonal='mul',seasonal_periods=7).fit()
hmodel
```

Out[24]:
```
<statsmodels.tsa.holtwinters.results.HoltWintersResultsWrapper at 0x2970258b220>
```

In [25]:
```python
predic=hmodel.predict(start=len(train),end=len(trips)-1)
predic.index=data.index[start:end+1]
predic
```

Out[25]:
```
date
2015-02-17    10999.526758
2015-02-17    12649.590071
2015-02-17    18893.507648
2015-02-17    15102.329697
2015-02-17    13362.264651
                  ...
2015-02-28    18452.770035
2015-02-28    16326.672885
2015-02-28    20019.813053
2015-02-28    25256.693016
2015-02-28    13742.326026
Length: 71, dtype: float64
```

In [26]:
```python
from sklearn.metrics import mean_squared_error
error=np.sqrt(mean_squared_error(test.trips,predic))
error
```

Out[26]:
```
12031.311661861006
```

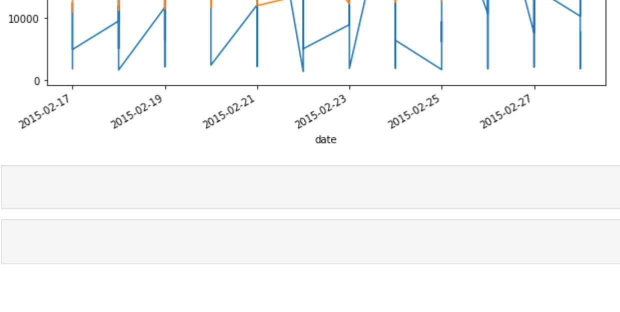
In [27]:
```python
test['trips'].mean(),np.sqrt(test.var())
```

Out[27]:
```
(13421.0,
 active_vehicles     1180.492327
 trips              11083.610744
 dtype: float64)
```

In [ ]:

In [28]:
```python
#train.plot(legend=True, label='Train')
test['trips'].plot(legend=True, label='Test',figsize=(10,6))
predic.plot(legend=True, label='prediction_hw')
```

Out[28]:
```
<AxesSubplot:xlabel='date'>
```



In [ ]:

In [ ]: