

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

```
In [2]: df = pd.read_csv(r'C:\Users\309962\Desktop\VehicleTraffic.csv', parse_dates=[0],
```

```
In [3]: # Measurements taken at different times
df
```

Out[3]:

	Vehicles	Average Speed (mph)	Accidents
TimeStamp			
2018-12-04 13:00:00	95.0	38.0	0.0
2018-12-04 14:00:00	90.0	32.0	1.0
2018-12-04 15:00:00	98.0	30.0	1.0
2018-12-04 16:00:00	98.0	26.0	3.0
2018-12-04 17:00:00	NaN	NaN	NaN
2018-12-04 18:00:00	NaN	NaN	NaN
2018-12-04 19:00:00	84.0	35.0	2.0
2018-12-04 20:00:00	82.0	40.0	0.0
2018-12-04 21:00:00	77.0	45.0	0.0
2018-12-04 22:00:00	93.0	45.0	1.0

```
In [4]: # Remove NaN values
df.dropna()
```

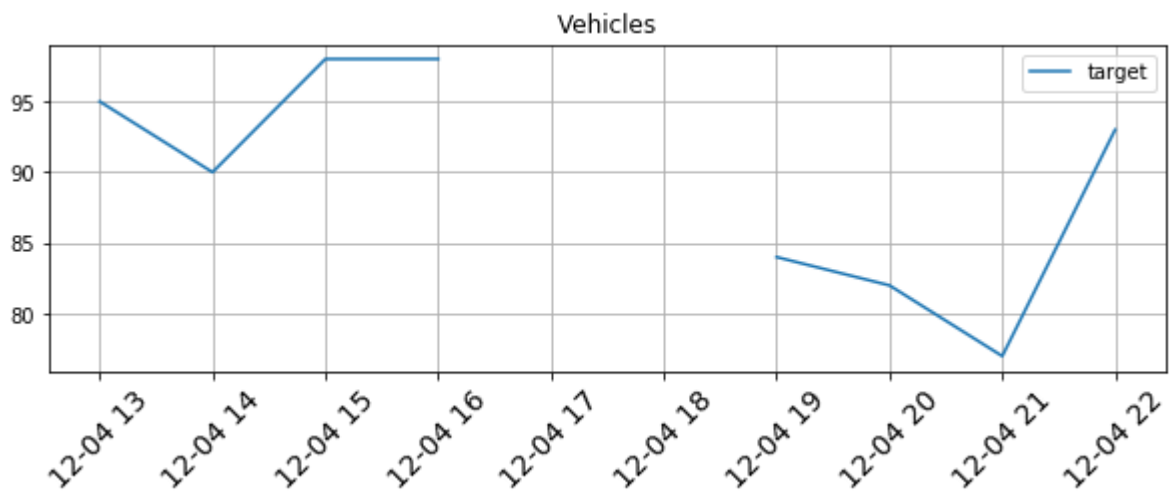
Out[4]:

	Vehicles	Average Speed (mph)	Accidents
TimeStamp			
2018-12-04 13:00:00	95.0	38.0	0.0
2018-12-04 14:00:00	90.0	32.0	1.0
2018-12-04 15:00:00	98.0	30.0	1.0
2018-12-04 16:00:00	98.0	26.0	3.0
2018-12-04 19:00:00	84.0	35.0	2.0
2018-12-04 20:00:00	82.0	40.0	0.0
2018-12-04 21:00:00	77.0	45.0	0.0
2018-12-04 22:00:00	93.0	45.0	1.0

```
In [5]: # Mean values of numeric columns  
df.mean()
```

```
Out[5]: Vehicles            89.625  
Average Speed (mph)      36.375  
Accidents                1.000  
dtype: float64
```

```
In [6]: # Let's visualize vehicles data  
# How does missing data show up?  
plt.figure(figsize=(10,3))  
plt.title('Vehicles')  
plt.plot(df['Vehicles'], label='target')  
plt.xticks(fontsize=14, rotation=45)  
plt.legend()  
plt.grid()
```

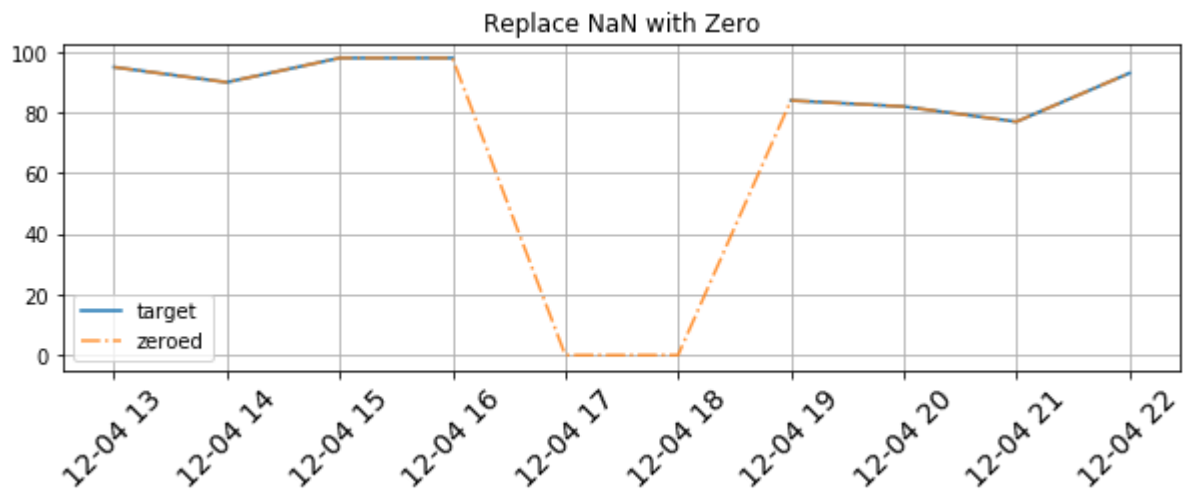


```
In [7]: # Replace missing values with zero

plt.figure(figsize=(10,3))
plt.title('Replace NaN with Zero')
plt.plot(df['Vehicles'], label='target')

# fillna to replace NaNs with provided value
vehicles = df['Vehicles'].fillna(0)

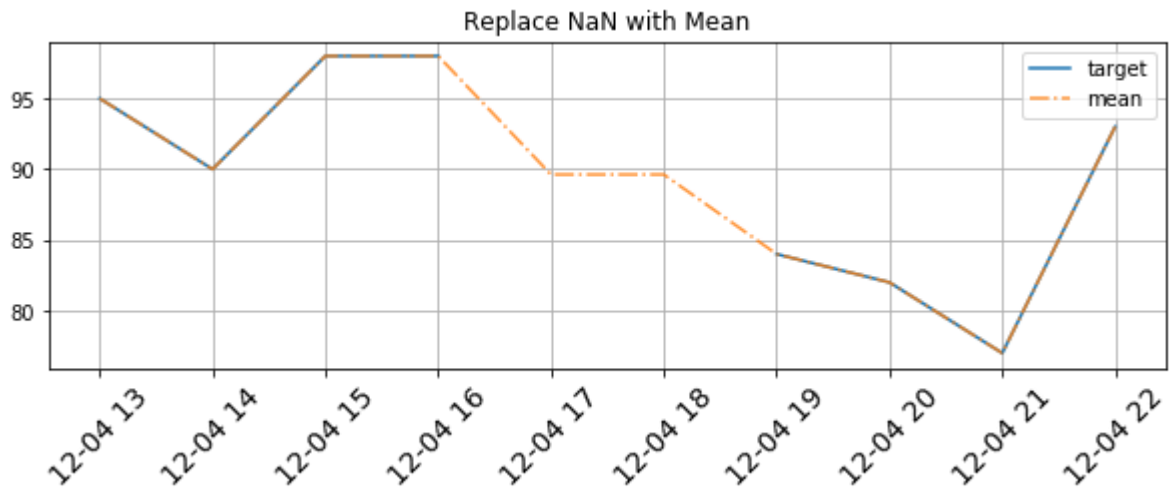
plt.plot(vehicles,ls='-.',alpha=0.8,label='zeroed')
plt.xticks(fontsize=14, rotation=45)
plt.legend()
plt.grid()
```



```
In [8]: # Replace missing values with mean value for that attribute
plt.figure(figsize=(10,3))
plt.title('Replace NaN with Mean')
plt.plot(df['Vehicles'], label='target')

#fillna to replace NaNs with provided value
vehicles = df['Vehicles'].fillna(df['Vehicles'].mean())

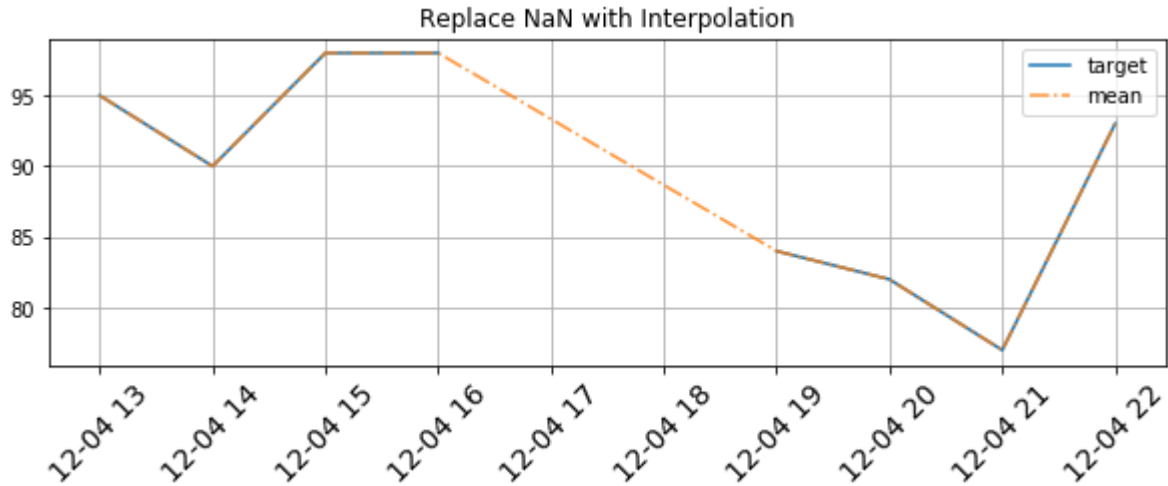
plt.plot(vehicles,ls='-.',alpha=0.8,label='mean')
plt.xticks(fontsize=14, rotation=45)
plt.legend()
plt.grid()
```



```
In [9]: # Replace missing values with interpolated value for that attribute
plt.figure(figsize=(10,3))
plt.title('Replace NaN with Interpolation')
plt.plot(df['Vehicles'], label='target')

vehicles = df['Vehicles'].interpolate()

plt.plot(vehicles,ls='-.',alpha=0.8,label='mean')
plt.xticks(fontsize=14, rotation=45)
plt.legend()
plt.grid()
```



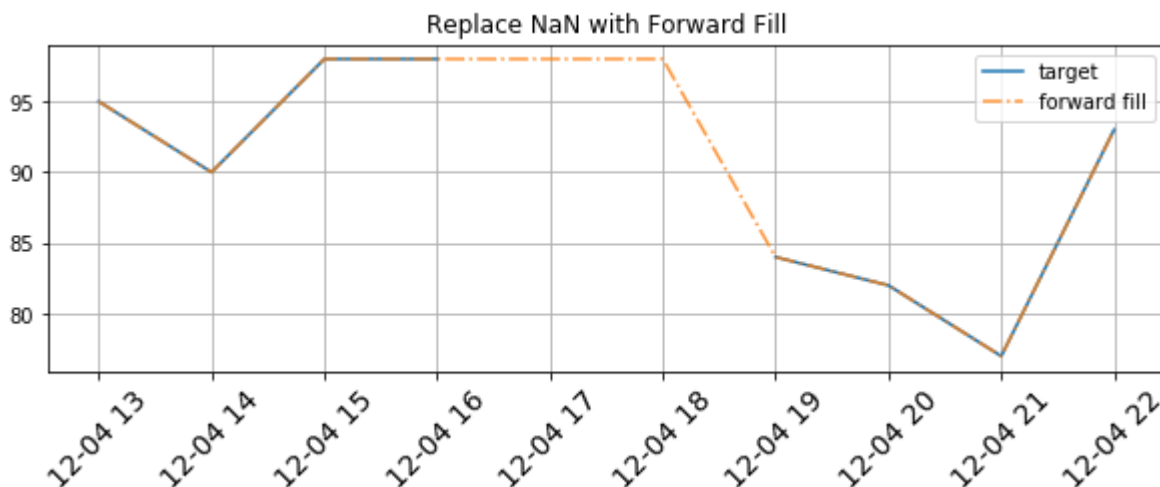
```
In [10]: vehicles
```

```
Out[10]: Timestamp
2018-12-04 13:00:00    95.000000
2018-12-04 14:00:00    90.000000
2018-12-04 15:00:00    98.000000
2018-12-04 16:00:00    98.000000
2018-12-04 17:00:00    93.333333
2018-12-04 18:00:00    88.666667
2018-12-04 19:00:00    84.000000
2018-12-04 20:00:00    82.000000
2018-12-04 21:00:00    77.000000
2018-12-04 22:00:00    93.000000
Name: Vehicles, dtype: float64
```

```
In [11]: # Replace missing values with previous valid value for that attribute
plt.figure(figsize=(10,3))
plt.title('Replace NaN with Forward Fill')
plt.plot(df['Vehicles'], label='target')

vehicles = df['Vehicles'].fillna(method='ffill')

plt.plot(vehicles,ls='-.',alpha=0.8,label='forward fill')
plt.xticks(fontsize=14, rotation=45)
plt.legend()
plt.grid()
```



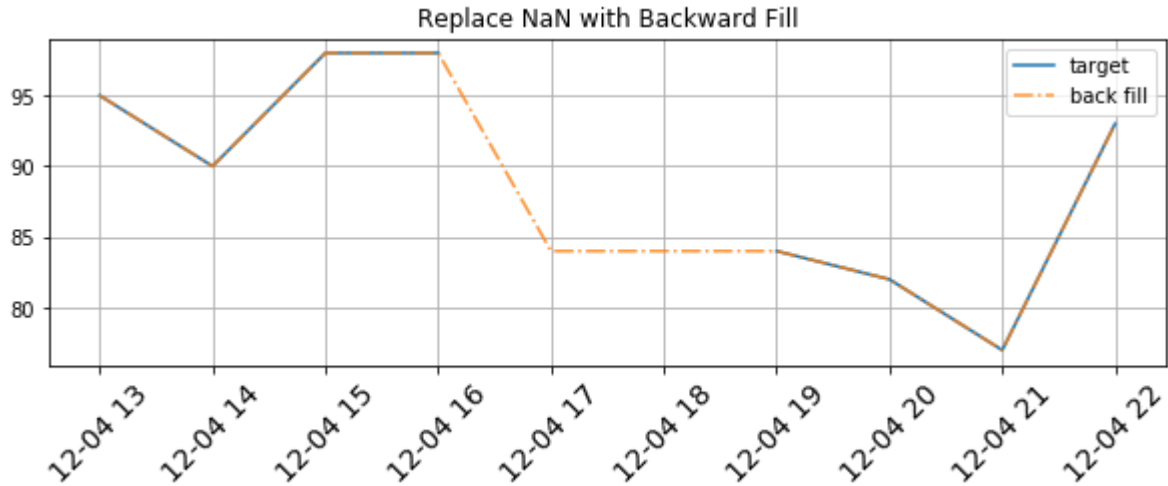
```
In [12]: vehicles
```

```
Out[12]: Timestamp
2018-12-04 13:00:00    95.0
2018-12-04 14:00:00    90.0
2018-12-04 15:00:00    98.0
2018-12-04 16:00:00    98.0
2018-12-04 17:00:00    98.0
2018-12-04 18:00:00    98.0
2018-12-04 19:00:00    84.0
2018-12-04 20:00:00    82.0
2018-12-04 21:00:00    77.0
2018-12-04 22:00:00    93.0
Name: Vehicles, dtype: float64
```

```
In [13]: # Replace missing values with next valid value for that attribute
plt.figure(figsize=(10,3))
plt.title('Replace NaN with Backward Fill')
plt.plot(df['Vehicles'], label='target')

vehicles = df['Vehicles'].fillna(method='bfill')

plt.plot(vehicles,ls='-.',alpha=0.8,label='back fill')
plt.xticks(fontsize=14, rotation=45)
plt.legend()
plt.grid()
```



```
In [14]: vehicles
```

```
Out[14]: Timestamp
2018-12-04 13:00:00    95.0
2018-12-04 14:00:00    90.0
2018-12-04 15:00:00    98.0
2018-12-04 16:00:00    98.0
2018-12-04 17:00:00    84.0
2018-12-04 18:00:00    84.0
2018-12-04 19:00:00    84.0
2018-12-04 20:00:00    82.0
2018-12-04 21:00:00    77.0
2018-12-04 22:00:00    93.0
Name: Vehicles, dtype: float64
```

```
In [15]: df
```

Out[15]:

	Vehicles	Average Speed (mph)	Accidents
TimeStamp			
2018-12-04 13:00:00	95.0	38.0	0.0
2018-12-04 14:00:00	90.0	32.0	1.0
2018-12-04 15:00:00	98.0	30.0	1.0
2018-12-04 16:00:00	98.0	26.0	3.0
2018-12-04 17:00:00	NaN	NaN	NaN
2018-12-04 18:00:00	NaN	NaN	NaN
2018-12-04 19:00:00	84.0	35.0	2.0
2018-12-04 20:00:00	82.0	40.0	0.0
2018-12-04 21:00:00	77.0	45.0	0.0
2018-12-04 22:00:00	93.0	45.0	1.0

```
In [ ]:
```