In [4]:
```
import numpy as np
import matplotlib.pyplot as plt
```

In [2]:
```
time = np.arange(1,51)
time
```

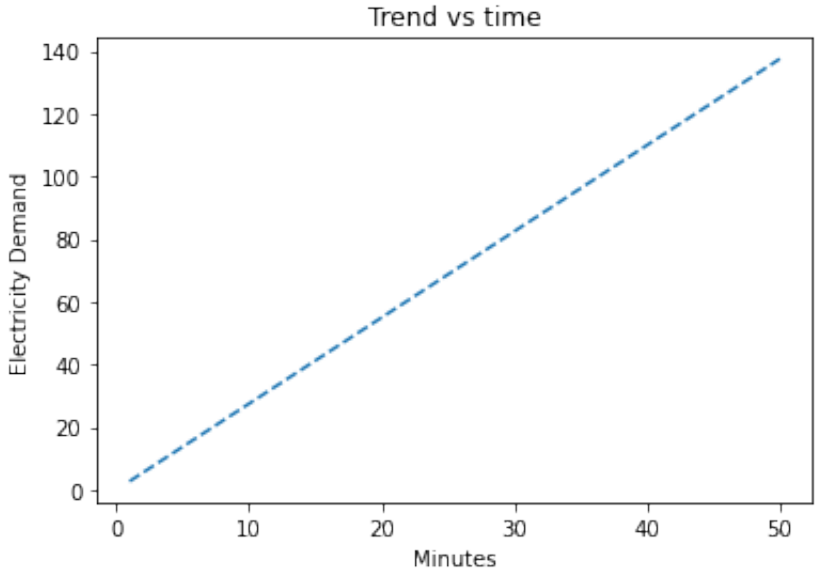
Out[2]:
```
array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16, 17,
       18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,
       35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50])
```

In [3]:
```
trend = time*2.75
trend
```

Out[3]:
```
array([  2.75,   5.5 ,   8.25,  11.  ,  13.75,  16.5 ,  19.25,  22.  ,
        24.75,  27.5 ,  30.25,  33.  ,  35.75,  38.5 ,  41.25,  44.  ,
        46.75,  49.5 ,  52.25,  55.  ,  57.75,  60.5 ,  63.25,  66.  ,
        68.75,  71.5 ,  74.25,  77.  ,  79.75,  82.5 ,  85.25,  88.  ,
        90.75,  93.5 ,  96.25,  99.  , 101.75, 104.5 , 107.25, 110.  ,
       112.75, 115.5 , 118.25, 121.  , 123.75, 126.5 , 129.25, 132.  ,
       134.75, 137.5 ])
```

In [7]:

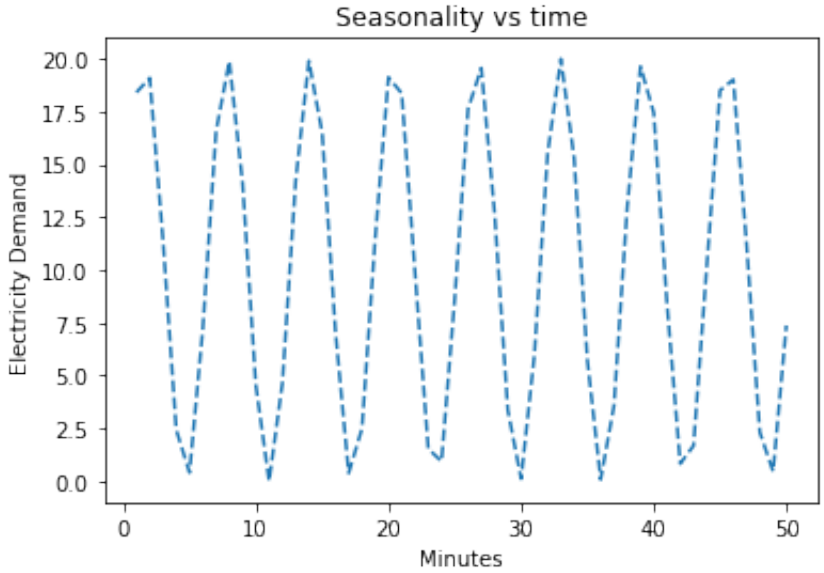
```
plt.plot(time,trend,linestyle='dashed')
plt.title("Trend vs time")
plt.xlabel("Minutes")
plt.ylabel("Electricity Demand")
plt.show()
```



In [8]:
```
seasonal = 10 + np.sin(time)*10
seasonal
```

Out[8]:
```
array([1.84147098e+01, 1.90929743e+01, 1.14112001e+01, 2.43197505e+00,
       4.10757253e-01, 7.20584502e+00, 1.65698660e+01, 1.98935825e+01,
       1.41211849e+01, 4.55978889e+00, 9.79344930e-05, 4.63427082e+00,
       1.42016704e+01, 1.99060736e+01, 1.65028784e+01, 7.12096683e+00,
       3.86025081e-01, 2.49012753e+00, 1.14987721e+01, 1.91294525e+01,
       1.83665564e+01, 9.91148691e+00, 1.53779596e+00, 9.44216380e-01,
       8.67648250e+00, 1.76255845e+01, 1.95637593e+01, 1.27090579e+01,
       3.36366116e+00, 1.19683759e-01, 5.95962355e+00, 1.55142668e+01,
       1.99991186e+01, 1.52908269e+01, 5.71817331e+00, 8.22114656e-02,
       3.56461867e+00, 1.29636858e+01, 1.96379539e+01, 1.74511316e+01,
       8.41377331e+00, 8.34784521e-01, 1.68225257e+00, 1.01770193e+01,
       1.85090352e+01, 1.90178835e+01, 1.12357312e+01, 2.31745339e+00,
       4.62473472e-01, 7.37625146e+00])
```

In [9]:
```
plt.plot(time,seasonal,linestyle='dashed')
plt.title("Seasonality vs time")
plt.xlabel("Minutes")
plt.ylabel("Electricity Demand")
plt.show()
```

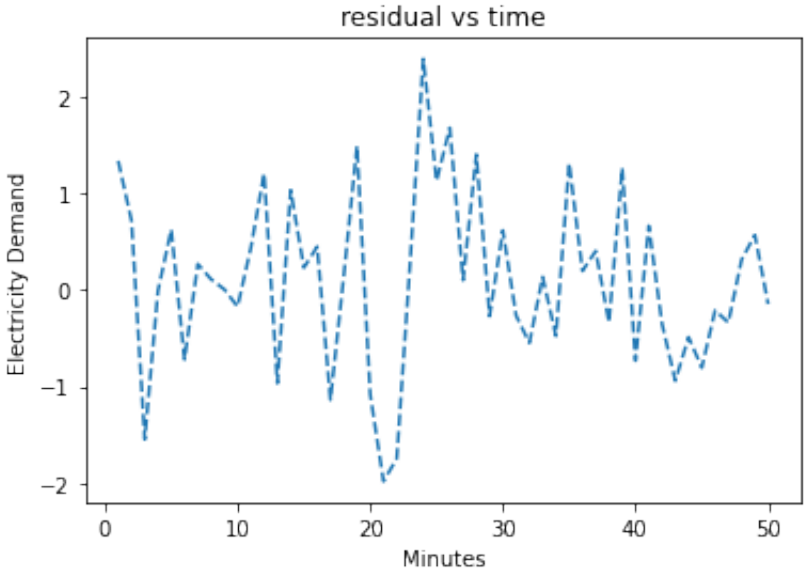


In [10]:
```
np.random.seed(10)
residual = np.random.normal(loc=0.0,scale = 1,size = len(time))
residual
```

Out[10]:
```
array([ 1.3315865 ,  0.71527897, -1.54540029, -0.00838385,  0.62133597,
       -0.72008556,  0.26551159,  0.10854853,  0.00429143, -0.17460021,
        0.43302619,  1.20303737, -0.96506567,  1.02827408,  0.22863013,
        0.44513761, -1.13660221,  0.13513688,  1.484537  , -1.07980489,
       -1.97772828, -1.7433723 ,  0.26607016,  2.38496733,  1.12369125,
        1.67262221,  0.09914922,  1.39799638, -0.27124799,  0.61320418,
       -0.26731719, -0.54930901,  0.1327083 , -0.47614201,  1.30847308,
        0.19501328,  0.40020999, -0.33763234,  1.25647226, -0.7319695 ,
        0.66023155, -0.35087189, -0.93943336, -0.48933722, -0.80459114,
       -0.21269764, -0.33914025,  0.31216994,  0.56515267, -0.14742026])
```

In [11]:

```
plt.plot(time,residual,linestyle='dashed')
plt.title("residual vs time")
plt.xlabel("Minutes")
plt.ylabel("Electricity Demand")
plt.show()
```



In [12]:

```
# aggregate

additive = trend+seasonal+residual
additive
```

Out[12]:

```
array([ 22.49629635,  25.30825324,  18.11579979,  13.4235912 ,
        14.78209323,  22.98575946,  36.08537757,  42.00213099,
        38.87547628,  31.88518868,  30.68312412,  38.83730819,
        48.9866047 ,  59.43434763,  57.98150853,  51.56610445,
        45.99942287,  52.12526441,  65.2333091 ,  73.04964762,
        74.1388281 ,  68.66811461,  65.05386612,  69.32918371,
        78.55017375,  90.79820672,  93.9129085 ,  91.10705426,
        82.84241317,  83.23288794,  90.94230636, 102.9649578 ,
       110.8818269 , 108.31468485, 103.27664639,  99.27722474,
       105.71482865, 117.12605345, 128.14442613, 126.7191621 ,
       121.82400486, 115.98391263, 118.99281921, 130.68768203,
       141.4544441 , 145.30518584, 140.14659098, 134.62962332,
       135.77762614, 144.72883121])
```

In [13]:

```
plt.plot(time,additive,linestyle='dashed')
plt.title("Additive time series")
plt.xlabel("Minutes")
plt.ylabel("Electricity Demand")
plt.show()
```