Autocovariance:

In [24]:
```
import pandas as pd
```

In [25]:
```
df=pd.DataFrame({'a':[13,5,11,12,9]})
```

In [26]:
```
df
```

Out[26]:

	a
0	13
1	5
2	11
3	12
4	9

In [27]:
```
import statsmodels.api as sm
from statsmodels.tsa.stattools import acovf,acf
arr=acovf(df['a'])
```

In [28]:
```
arr
```

Out[28]:
```
array([ 8. , -4. , -1.6,  2.2, -0.6])
```

Autocorrelation:

In [29]:
```
acf=acf(df['a'])
acf
#we get first value as 1 since autocorrelation of a variable to itself is 1.
```

Out[29]:
```
array([ 1.   , -0.5  , -0.2  ,  0.275, -0.075])
```

Calculate auto covariance of temperature given in the dataset
daily-minimum-temperatures.csv

In [10]:
```
df=pd.read_csv('daily-minimum-temperatures.csv',index_col='Date',parse_dates=
```

In [15]:
```
a=df.head(10)
```

In [16]:
```
df.shape
```

Out[16]:
```
(3650, 1)
```

In [17]:
```
arr=acovf(a['Temp'])
arr
```

Out[17]:
```
array([ 5.2624 ,  1.98564,  0.34088, -1.61388, -1.88944, -2.3618 ,
       -0.91416,  0.08628,  1.12752,  0.60776])
```

In [18]:
```
var=a['Temp'].var()
```

In [19]:
```
var
```

Out[19]:
```
5.8471111111111105
```

In [20]:
```
import statistics
var=statistics.variance(a['Temp'])
```

In [21]:
```
var
```

Out[21]:
```
5.847111111111111
```

In [ ]:
```

```

In [32]:
```
df=pd.read_csv("samples.csv",index_col="Unnamed: 0",parse_dates=True)
```

In [33]:
```
df.head()
```

Out[33]:

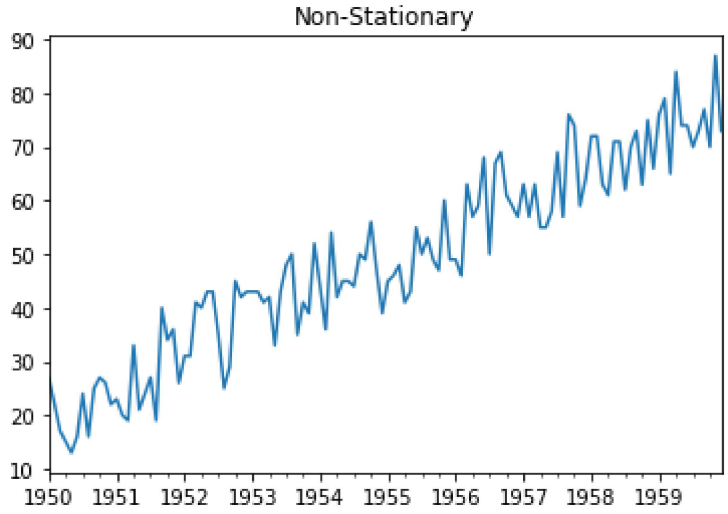
	a	b	c	d
1950-01-01	36	27	0	67
1950-02-01	58	22	3	31
1950-03-01	61	17	5	67
1950-04-01	37	15	8	47
1950-05-01	66	13	8	62

In [35]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 120 entries, 1950-01-01 to 1959-12-01
Data columns (total 4 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   a       120 non-null    int64
 1   b       120 non-null    int64
 2   c       120 non-null    int64
 3   d       120 non-null    int64
dtypes: int64(4)
memory usage: 4.7 KB
```

In [42]:
```
df['b'].plot(title="Non-Stationary")
```

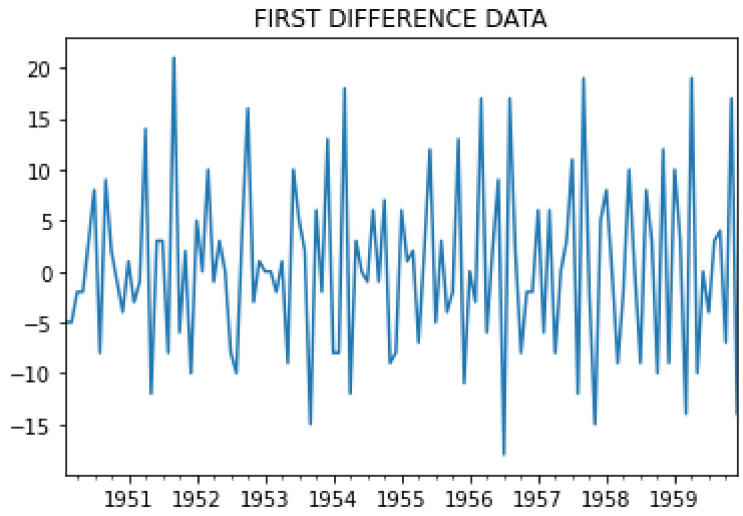
Out[42]: <AxesSubplot:title={'center':'Non-Stationary'}>



In [34]:
```
#Removing non-stationarity using differencing
from statsmodels.tsa.statespace.tools import diff
```

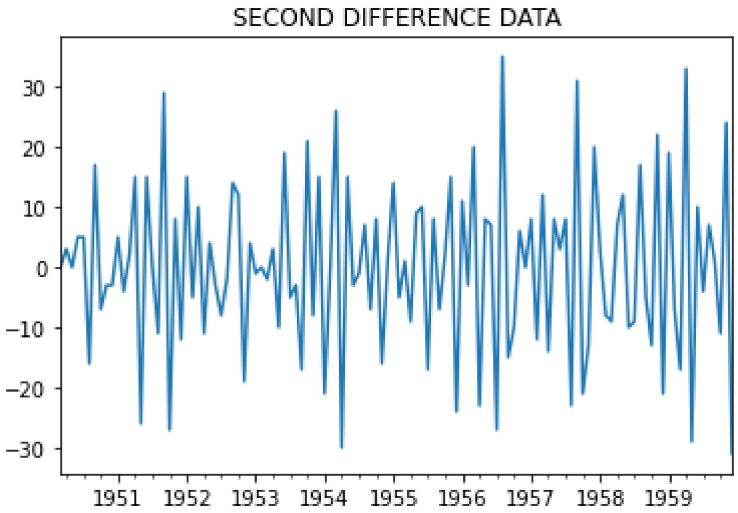
First difference:

In [36]:
```
df['d1']=diff(df['b'],k_diff=1)
df['d1'].plot(title="FIRST DIFFERENCE DATA").autoscale(axis='x',tight=True)
```



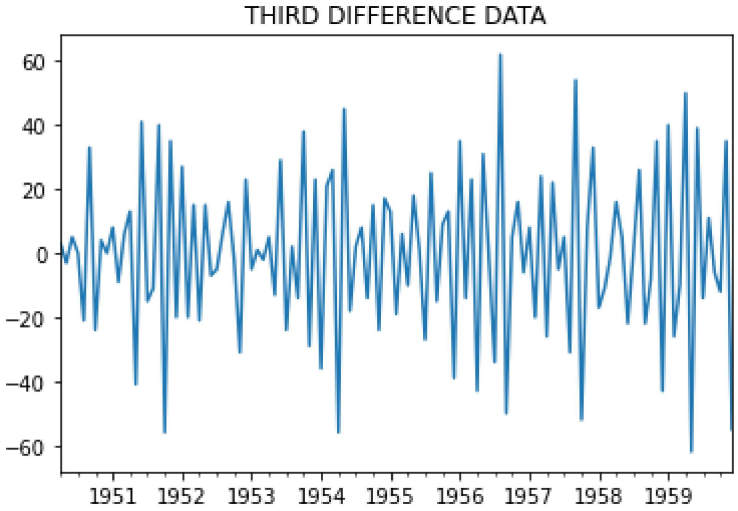
Second difference:

In [38]:
```
df['d1']=diff(df['b'],k_diff=2)
df['d1'].plot(title="SECOND DIFFERENCE DATA").autoscale(axis='x',tight=True)
```



Third difference:

In [40]:
```
df['d1']=diff(df['b'],k_diff=3)
df['d1'].plot(title="THIRD DIFFERENCE DATA").autoscale(axis='x',tight=True)
```



Regression:Predictive analysis

ARIMA:Auto Regressive Integrated Moving Average

In [ ]:
```
Simple moving average:applicable to time series
    where u r finding rolling mean
```

In [ ]:

```
Case study:
    Use Superstore.xls
    The index column is Time series.
    The data mostly contains categorical variables.
    Objective:To understand the general trend in the data
        and also perform EDA to gain insights.
        Analyse,Predict and forecast the sales of the category:Technology
```