Time Series Analysis of Daily births using FBprophet

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

```
In [3]: import seaborn as sns
        import matplotlib.pyplot as plt
        %matplotlib inline
```

```
In [4]: import fbprophet
```

```
ERROR:fbprophet:Importing plotly failed. Interactive plots will not work.
```

```
In [5]: df = pd.read_csv('birthrate california.csv')
```

```
In [6]: df.head()
```

Out[6]:

	date	births
0	1959-01-01	35
1	1959-01-02	32
2	1959-01-03	30
3	1959-01-04	31
4	1959-01-05	44

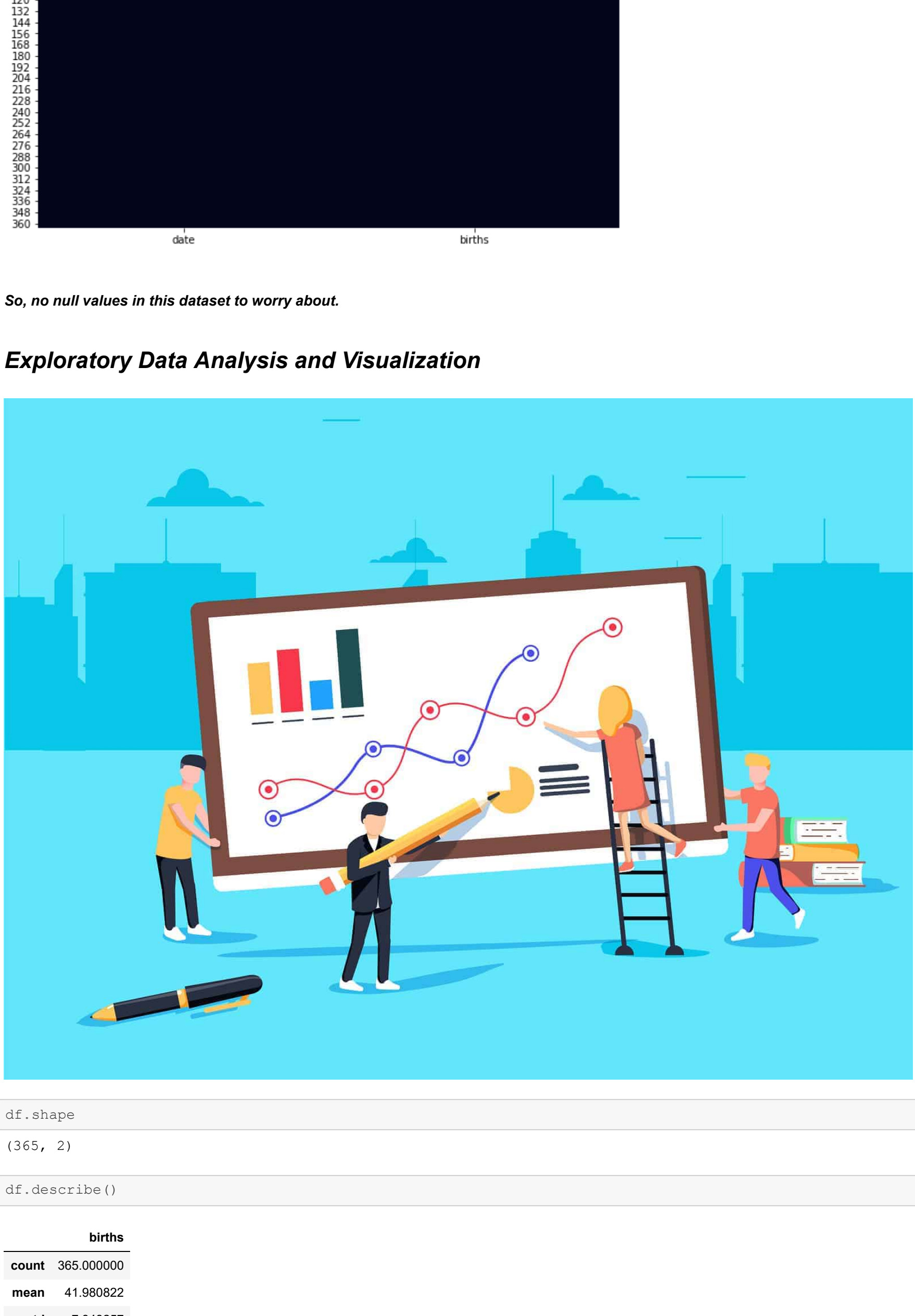
Checking for null values

```
In [7]: plt.figure(figsize=(10,6))
        sns.heatmap(df.isnull(), cbar=False)
        sns.set_context('poster')
```

```
INFO:numexpr.utils:NumExpr defaulting to 8 threads.
```

So, no null values in this dataset to worry about.

Exploratory Data Analysis and Visualization



```
In [19]: df.shape
```

Out[19]: (365, 2)

```
In [21]: df.describe()
```

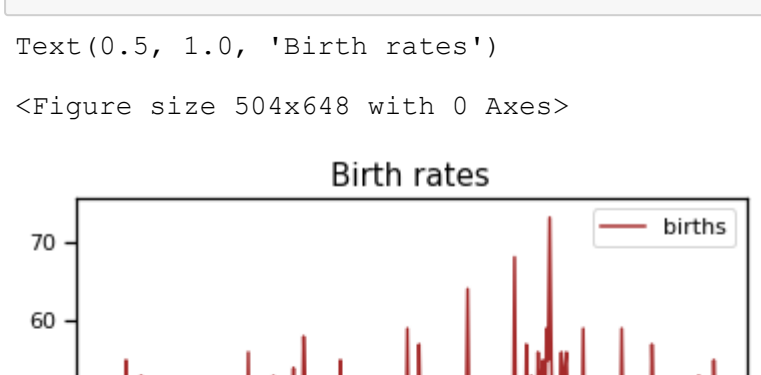
Out[21]:

	births
count	365.000000
mean	41.980822
std	7.348257
min	23.000000
25%	37.000000
50%	42.000000
75%	46.000000
max	73.000000

```
In [23]: plt.figure(figsize=(7,9))
         df.plot(color='brown')
         sns.set_context('notebook')
         plt.title('Birth rates', size = 15)
```

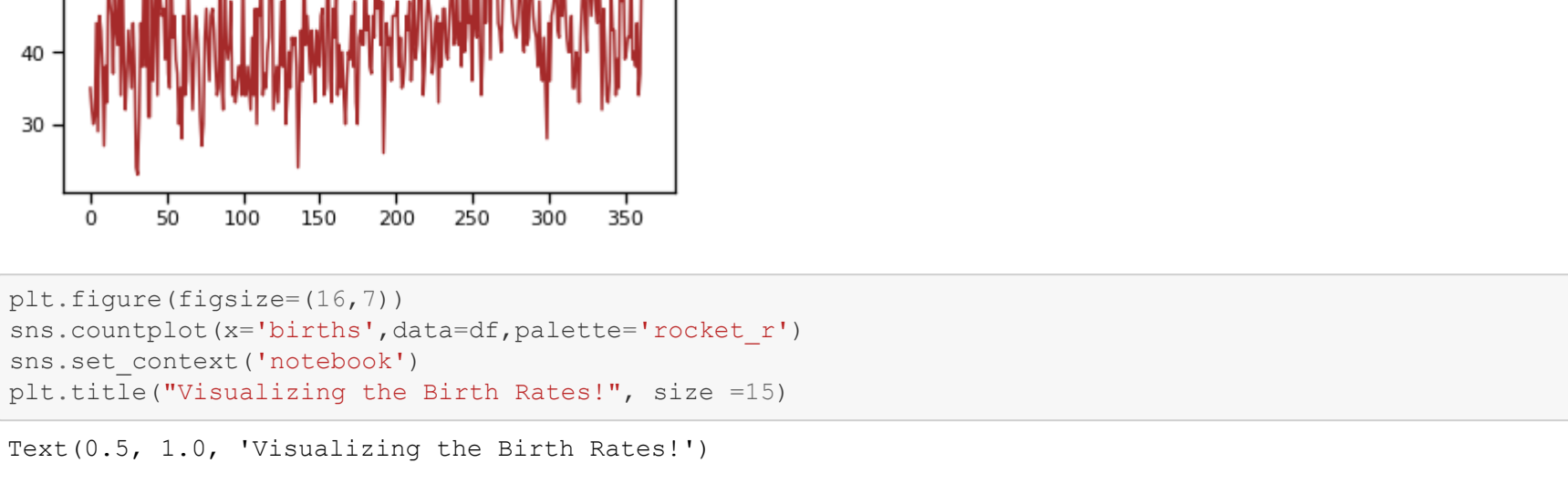
Out[23]: Text(0.5, 1.0, 'Birth rates')

<Figure size 504x648 with 0 Axes>

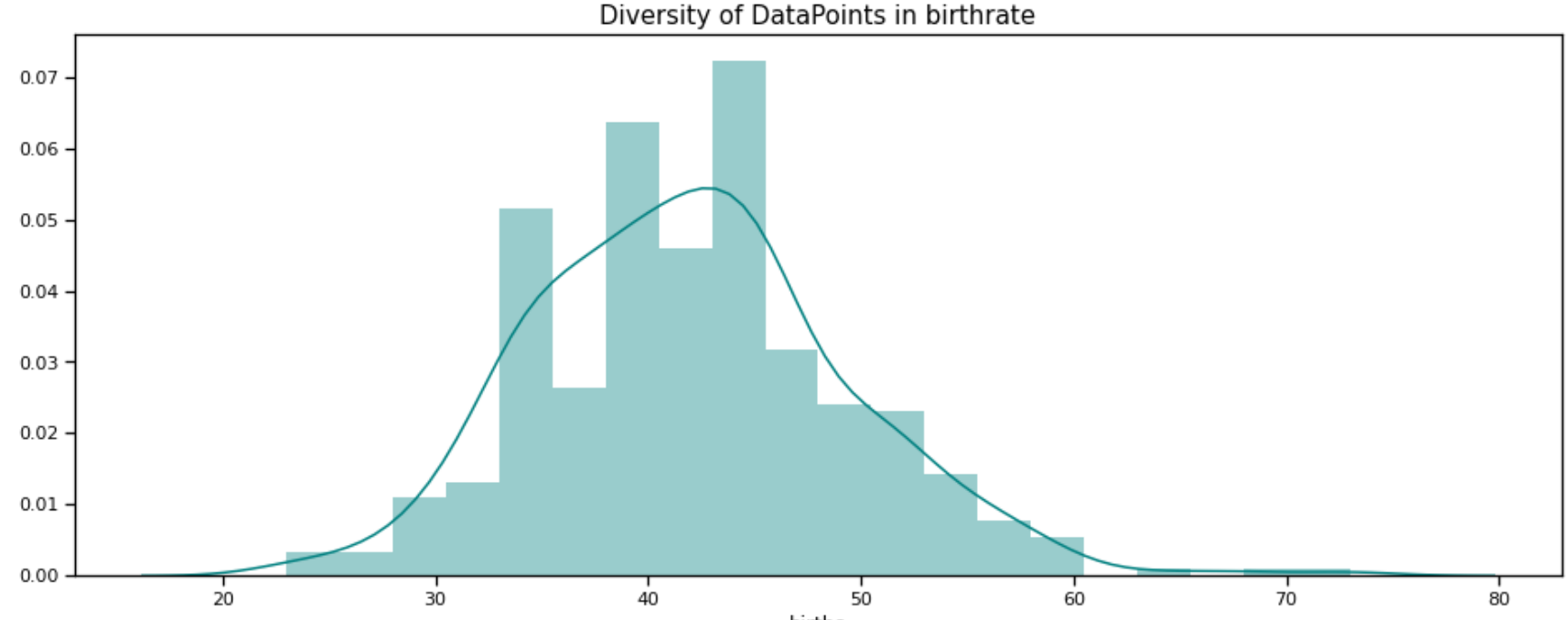



```
In [27]: plt.figure(figsize=(16,7))
         sns.countplot(x='births',data=df,palette='rocket_r')
         sns.set_context('notebook')
         plt.title("Visualizing the Birth Rates!", size =15)
```

Out[27]: Text(0.5, 1.0, 'Visualizing the Birth Rates!')




```
In [32]: plt.figure(figsize=(16,6))
         sns.distplot(df['births'], color = 'teal')
         plt.title('Diversity of DataPoints in birthrate', size = 15)
         plt.show()
```




```
In [33]: from fbprophet.plot import add_changepoints_to_plot
```

```
In [37]: plt.figure(figsize=(20,8))
         plt.plot(df['date'], df['births'], color = 'purple');
         plt.grid()
         plt.title('Daily Births - 1959')
         plt.show()
```

```
INFO:matplotlib.category:Using categorical units to plot a list of strings that are all parsable as f
loats or dates. If these strings should be plotted as numbers, cast to the appropriate data type befo
re plotting.
INFO:matplotlib.category:Using categorical units to plot a list of strings that are all parsable as f
loats or dates. If these strings should be plotted as numbers, cast to the appropriate data type befo
re plotting.
```


```
In [38]: # Renaming column
         df.columns = ['ds','y']
```

```
In [39]: df.head()
```

Out[39]:

	ds	y
0	1959-01-01	35
1	1959-01-02	32
2	1959-01-03	30
3	1959-01-04	31
4	1959-01-05	44

Importing Prophet :

```
In [40]: from fbprophet import Prophet
```

Creating Model

```
In [41]: model=Prophet()
```

```
In [42]: df.columns
```

Out[42]: Index(['ds', 'y'], dtype='object')

Fitting the Model :

```
In [43]: import warnings
         warnings.filterwarnings("ignore")
```

```
In [44]: from fbprophet.plot import add_changepoints_to_plot
```

```
In [45]: with warnings.catch_warnings():
             warnings.simplefilter("ignore")
             model = fbprophet.Prophet(yearly_seasonality=True, daily_seasonality=False,
                                       changepoint_range=0.9,
                                       changepoint_prior_scale=0.5,
                                       seasonality_mode='multiplicative')
             model.fit(df)
```

Expanding the date columns:

```
In [46]: future = model.make_future_dataframe(periods=90, freq='d')
         future.tail()
```

Out[46]:

	ds
450	1960-03-26
451	1960-03-27
452	1960-03-28
453	1960-03-29
454	1960-03-30

And Hence, new dates are added !

Predicting ...

```
In [48]: prediction = model.predict(future)
```

```
In [50]: prediction.tail()
```

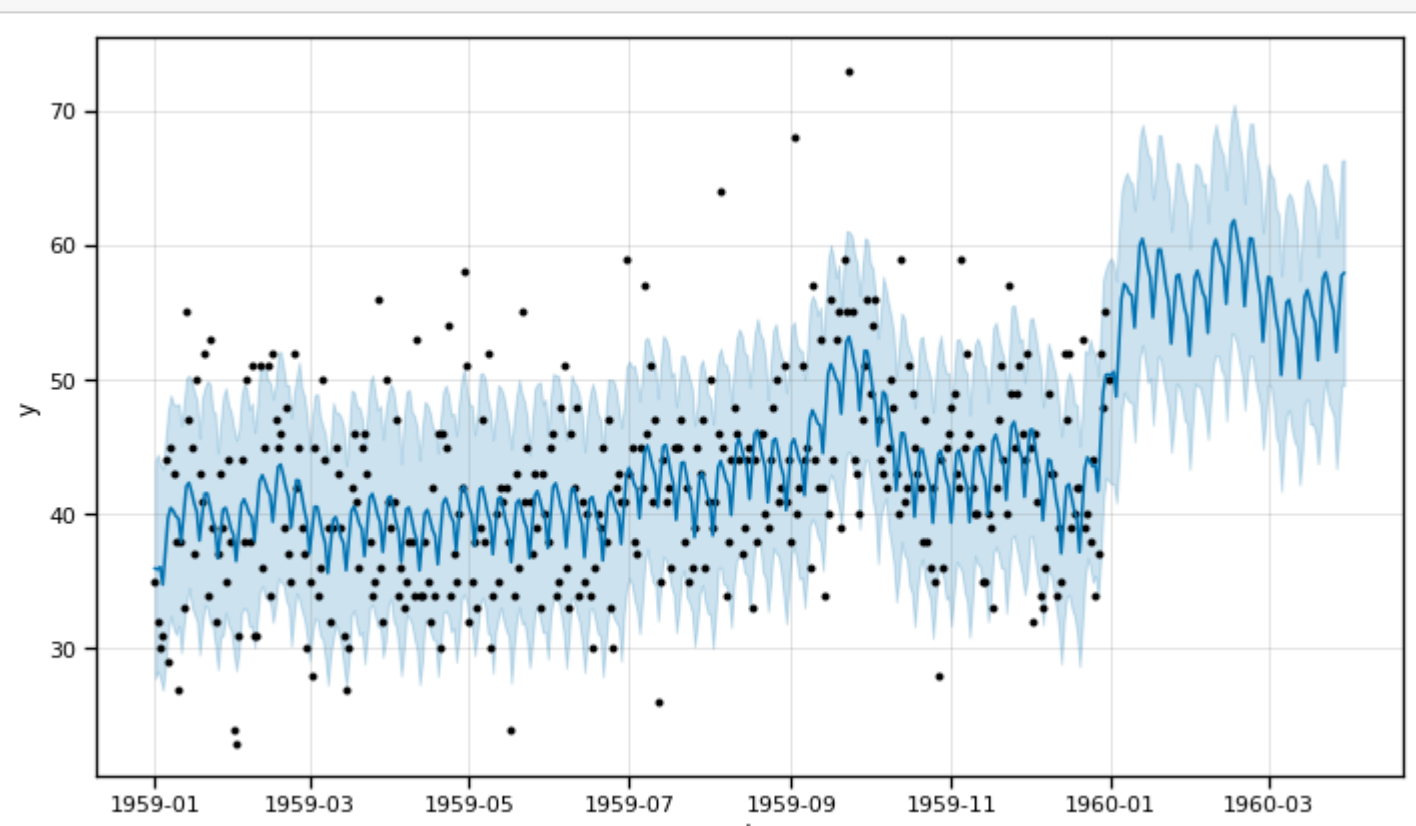
Out[50]:

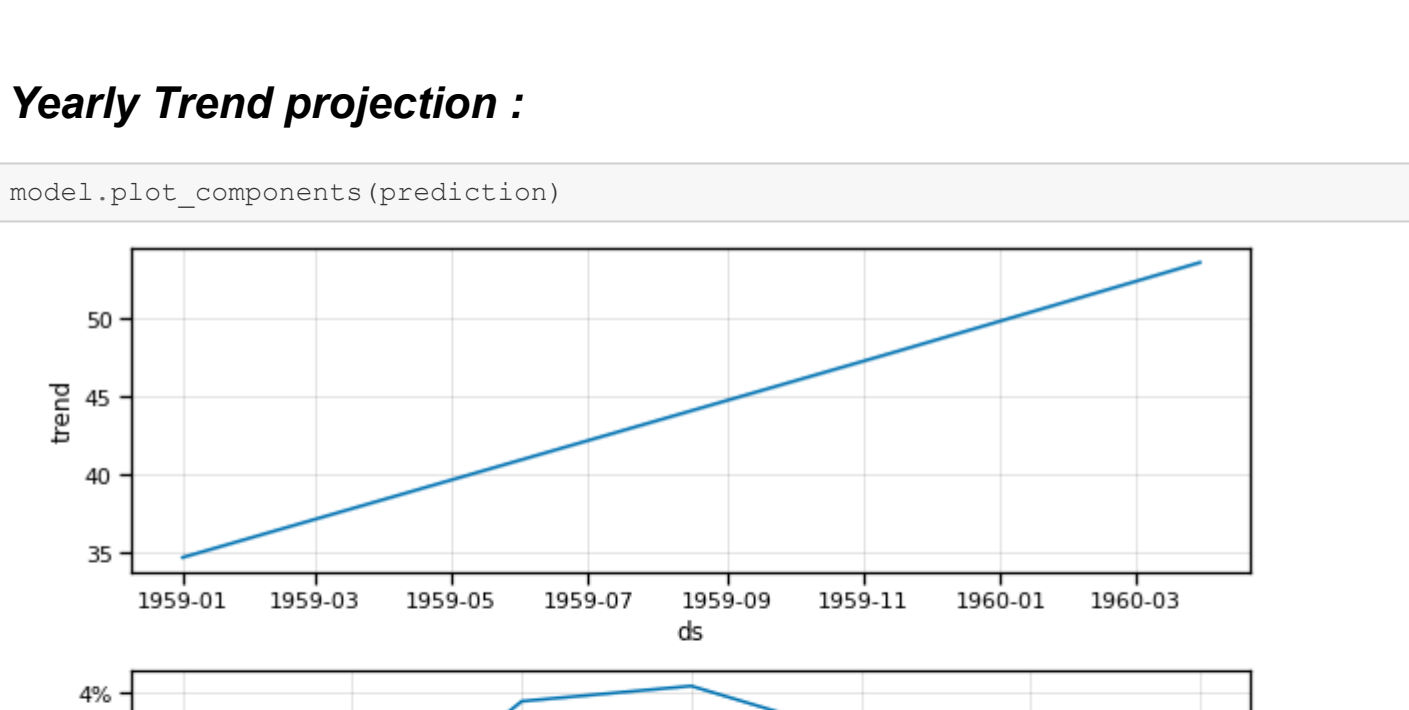
	ds	trend	yhat_lower	yhat_upper	trend_lower	trend_upper	multiplicative_terms	multiplicative_terms_lower	multiplicative_ter
450	1960-03-26	53.441448	47.257098	63.210890	53.435661	53.447833	0.031310	0.031310	
451	1960-03-27	53.483633	43.387404	60.568555	53.477754	53.490063	-0.026680	-0.026680	
452	1960-03-28	53.525818	46.110337	62.652386	53.519826	53.532300	0.026589	0.026589	
453	1960-03-29	53.568003	49.612341	66.297938	53.561892	53.574530	0.077434	0.077434	
454	1960-03-30	53.610187	49.569960	66.321982	53.603954	53.616806	0.080339	0.080339	

Plotting the Prediction!

```
In [51]: model.plot(prediction)
```

Out[51]:

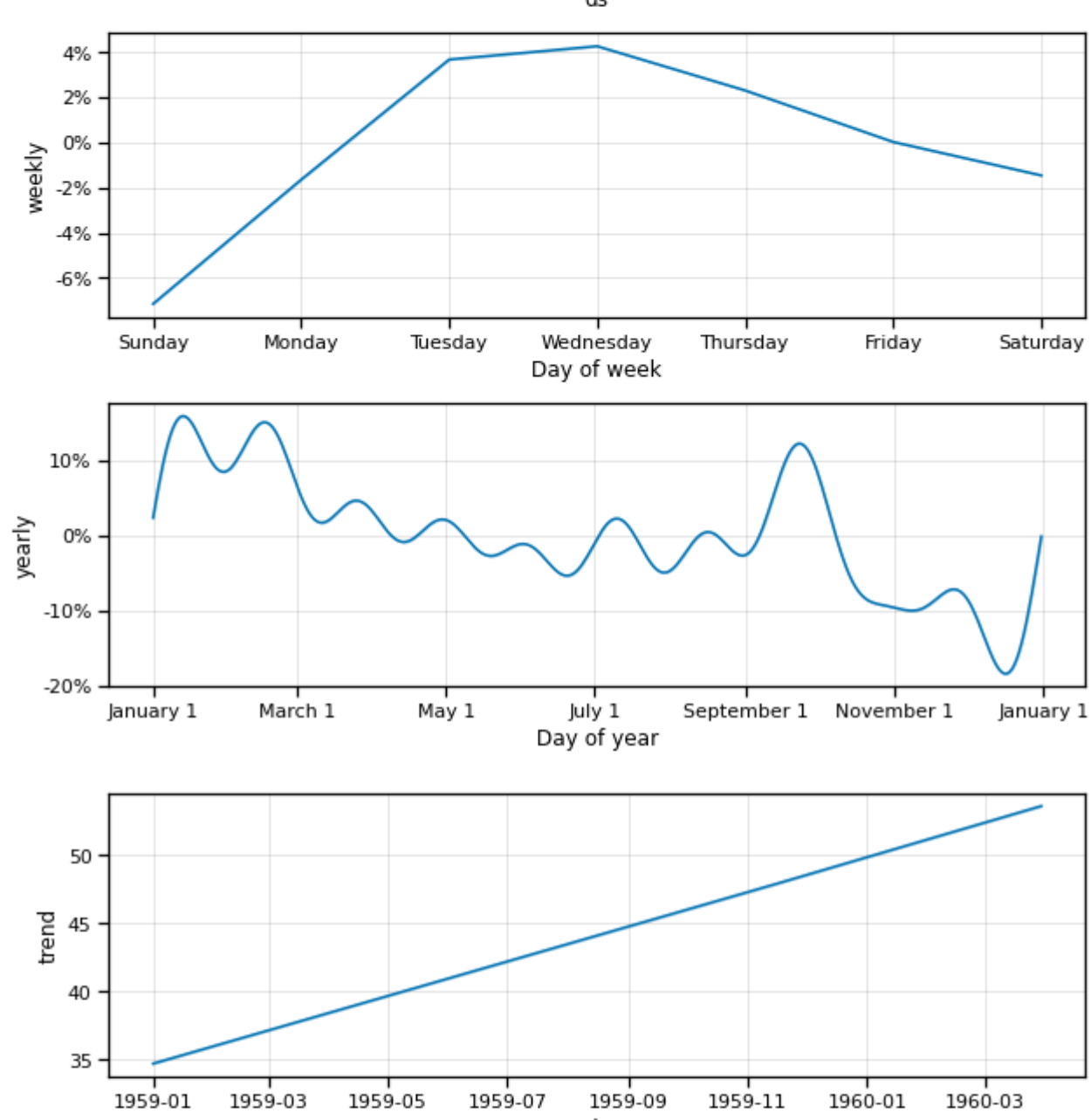


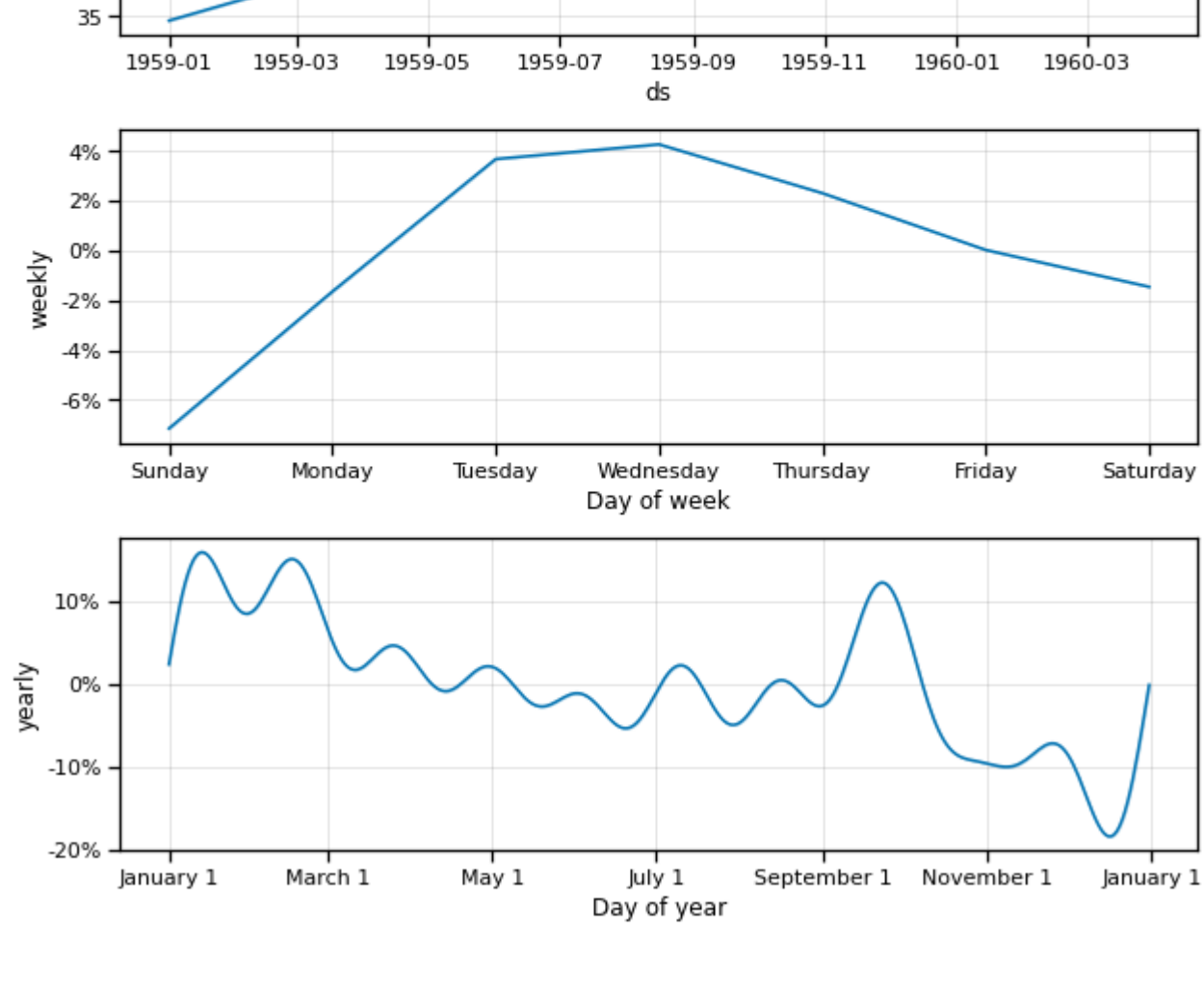


Yearly Trend projection :

```
In [52]: model.plot_components(prediction)
```

Out[52]:





Conclusion :

Here , I have done the Time series Analysis
And Prediction has been done using FB prophet.

And, with interactive visualization, shown the Analysis as well as the Prediction.