## Imports

```
from statsmodels.tsa.holtwinters
import ExponentialSmoothing
```

## Steps to fit the model and check it

| | |
|---|---|
| train = df.iloc[:n] | use iloc to split the original dataset |
| fitted_mod = ExponentialSmoothing(train, trend='mul or add', seasonal='mul or add', seasonal_periods=n_unit).fit() | create and fit the model |
| predictions = fitted_mod.forecast(n of units) | forecast |
| train.plot()<br>test.plot()<br>predictions.plot() | plot forecasted values together with train and test data |

## Evaluation metrics

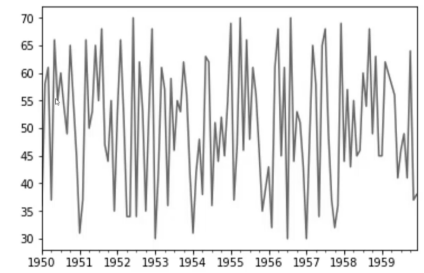| | |
|---|---|
| from sklearn.metrics import mean_squared_error, mean_absolute_error | import the necessary libraries |
| mean_squared_error(test, predictions) | calculate the MSE |
| np.sqrt(mean_squared_error(test, predictions)) | calculate the RMSE |

now that we saw our model was not that far off (if that's the case), we retrain our model on the entire dataset and we can plot it to show the future behaviour of our data
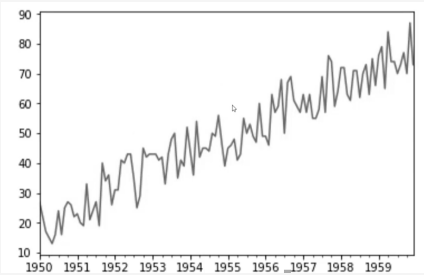
## IMPORTANT CONCEPTS

**STATIONARY** data: these kinds of data do not exhibit trends or seasonality.
**NON-STATIONARY** data: these kinds of data exhibit trends or seasonality.

## - stationary data



## - non stationary data



## - via code...

| | |
|---|---|
| from statmodels.tsa.statespace.tools import diff | import libraries |
| diff(df["timeseries col"], k_diff=1) | use the diff() func to check stationarity |