

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 = ExponentialS-   create and
moothing(train,              fit the
trend='mul or add',          model
seasonal='mul or add',
seasonal_periods=n_unit).fit()
```

```
predictions = fitted_mod.fore- forecast
cast(n of units)
```

```
train.plot()                plot
test.plot()                 forecasted
predictions.plot()          values
                             together
                             with train
                             and test
                             data
```

Evaluation metrics

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

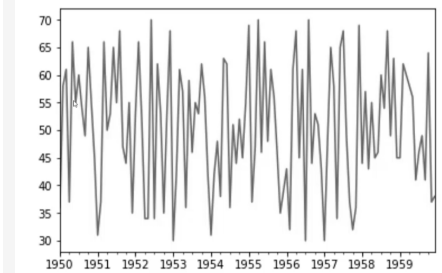
now that we saw our model was not that far off (if that's the case), we retrain our model on the entire dataset and we can plot it to show the future behaviour of our data

IMPORTANT CONCEPTS

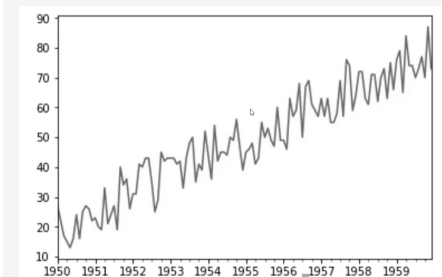
STATIONARY data: these kinds of data do not exhibit trends or seasonality.

NON-STATIONARY data: these kinds of data exhibit trends or seasonality.

- stationary data



- non stationary data



- via code...

```
from statmodels.tsa.s-   import libraries
tatespace.tools import
```

```
diff                      use the diff() func
diff(df["timeseries col"], k_diff=1)  to check statio-
                                      narity
```



By **DarioPittera** (aggialavura)

cheatography.com/aggialavura/
www.dariopittera.com

Not published yet.

Last updated 2nd June, 2022.

Page 1 of 1.

Sponsored by **CrosswordCheats.com**

Learn to solve cryptic crosswords!

<http://crosswordcheats.com>