import numpy as np

import pandas as pd

import statsmodels.api as sm

df = pd.read\_csv('http://vincentarelbundock.github.io/Rdatasets/csv/datasets/AirPassengers.csv', index\_col=0)

df.head()

y = df.AirPassengers # response

X = df.time # predictor

X = sm.add\_constant(X) # Adds a constant term to the predictor

X.head()

est = sm.OLS(y, X)

est = est.fit()

est.summary()

est.params

%pylab inline

X\_prime = np.linspace(X.time.min(), X.time.max(), 100)[:, np.newaxis]

X\_prime = sm.add\_constant(X\_prime) # add constant as we did before

y\_hat = est.predict(X\_prime)

plt.scatter(X.time, y, alpha=0.3) # Plot the raw data

plt.xlabel("Time of Travel")

plt.ylabel("Total Air Passengers")

plt.plot(X\_prime[:, 1], y\_hat, 'r', alpha=0.9) # Add the regression line, colored in red

bokeh

from bokeh.plotting import figure, HBox, output\_file, show, VBox

from bokeh.models import Range1d

------------------------------------------------------------------------------------------------

################################################################################################

# import formula api as alias smf

import statsmodels.formula.api as smf

# formula: response ~ predictors

est = smf.ols(formula='AirPassengers ~ time', data=df).fit()

est.summary()

# Fit the no-intercept model

est\_no\_int = smf.ols(formula='AirPassengers ~ time - 1', data=df).fit()

# We pick 100 hundred points equally spaced from the min to the max

X\_prime\_1 = pd.DataFrame({'time': np.linspace(X.time.min(), X.time.max(), 100)})

X\_prime\_1 = sm.add\_constant(X\_prime\_1) # add constant as we did before

y\_hat\_int = est.predict(X\_prime\_1)

y\_hat\_no\_int = est\_no\_int.predict(X\_prime\_1)

fig = plt.figure(figsize=(8,4))

splt = plt.subplot(121)

splt.scatter(X.time, y, alpha=0.3) # Plot the raw data

plt.ylim(100, 700) # Set the y-axis to be the same

plt.xlabel("Time of Travel")

plt.ylabel("Total Air Passengers")

plt.title("With intercept")

splt.plot(X\_prime[:, 1], y\_hat\_int, 'r', alpha=0.9) # Add the regression line, colored in red

splt = plt.subplot(122)

splt.scatter(X.time, y, alpha=0.3) # Plot the raw data

plt.xlabel("Time of Travel")

plt.title("Without intercept")

splt.plot(X\_prime[:, 1], y\_hat\_no\_int, 'r', alpha=0.9) # Add the regression line, colored in red

pip install bokeh

import bokeh