Data Analysis and Machine-Learning

Chapter 2.4:

Linear Regression Model (4): Inferential Statistics



1. Introduction: Sample OLS Summaries

#Import samples: diabetes

from sklearn import datasets

data = datasets.load\_diabetes()

#Split features and target, and make dataframes of Xs and Y

features = pd.DataFrame( data['data'], columns=data['feature\_names'] )

target = pd.DataFrame(data['target'], columns=['target'])

#Import statsmodels for statistical summaries

from statsmodels.formula.api import ols

model = ols("target ~ features", data).fit()

print(model.summary())

Output:

OLS Regression Results

==============================================================================

Dep. Variable: target R-squared: 0.518

Model: OLS Adj. R-squared: 0.507

Method: Least Squares F-statistic: 46.27

Date: Prob (F-statistic): 3.83e-62

Time: Log-Likelihood: -2386.0

No. Observations: 442 AIC: 4794.

Df Residuals: 431 BIC: 4839.

Df Model: 10

Covariance Type: nonrobust

===============================================================================

coef std err t P>|t| [0.025 0.975]

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

Intercept 152.1335 2.576 59.061 0.000 147.071 157.196

features[0] -10.0122 59.749 -0.168 0.867 -127.448 107.424

features[1] -239.8191 61.222 -3.917 0.000 -360.151 -119.488

features[2] 519.8398 66.534 7.813 0.000 389.069 650.610

features[3] 324.3904 65.422 4.958 0.000 195.805 452.976

features[4] -792.1842 416.684 -1.901 0.058 -1611.169 26.801

features[5] 476.7458 339.035 1.406 0.160 -189.621 1143.113

features[6] 101.0446 212.533 0.475 0.635 -316.685 518.774

features[7] 177.0642 161.476 1.097 0.273 -140.313 494.442

features[8] 751.2793 171.902 4.370 0.000 413.409 1089.150

features[9] 67.6254 65.984 1.025 0.306 -62.065 197.316

==============================================================================

Omnibus: 1.506 Durbin-Watson: 2.029

Prob(Omnibus): 0.471 Jarque-Bera (JB): 1.404

Skew: 0.017 Prob(JB): 0.496

Kurtosis: 2.726 Cond. No. 227.

==============================================================================

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

As the table depicts, the OLS summaries provide various statistical insights regarding the model including model’s general accuracy (R2, adjusted R2), standard errors, beta coefficients (weights of Xs), intercept, and insights regarding statistical significance (F-statistic, t-test, and P-values).

1. Inferential Statistics
   1. R-Squared

For intuitive understanding of the equation, consider following samples for illustrations.

#Generate Samples

x = np.array([60, 65, 69, 75, 82, 88])

y = np.array([165, 168, 174, 177, 179, 183])

x = x.reshape(-1,1)

#Scatter Plot of the samples

plt.plot(x, y, 'o')

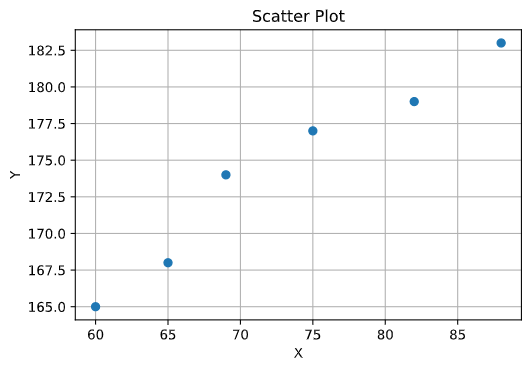
plt.title('Scatter Plot')

plt.xlabel('X')

plt.ylabel('Y')

plt.grid()

plt.show()



#Linear regression and visualization for explanation (1)

linearModel = LinearRegression().fit(x, y)

display(linearModel.coef\_)

display(linearModel.intercept\_)

fitline=linearModel.coef\_\*x+linearModel.intercept\_

meanline = np.mean(y)+0\*x

plt.figure(figsize=(10,5))

plt.subplot(1,2,1)

plt.plot(x,y,'o')

plt.plot(x, fitline, label='Linear Model')

plt.xlabel('X')

plt.ylabel('Y')

plt.legend()

plt.subplot(1,2,2)

plt.plot(x, y, 'o')

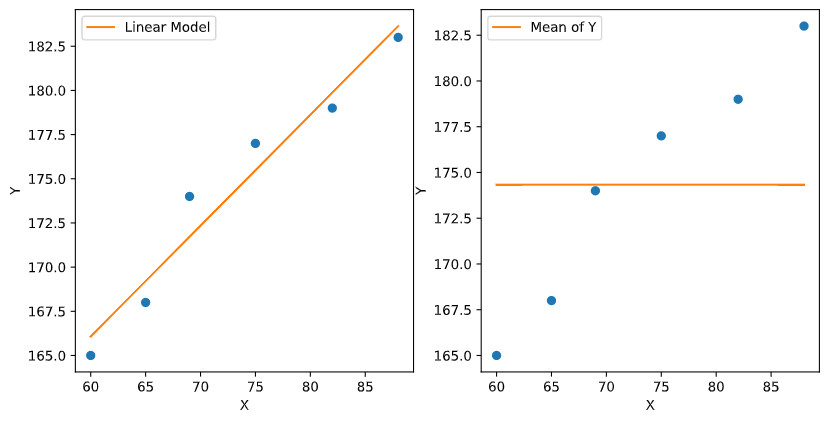
plt.plot(x, meanline, label='Mean of Y')

plt.xlabel('X')

plt.ylabel('Y')

plt.legend()

plt.show()



Now, going back to the equation of R-Squared, can be depicted as the sum of distances between the line in the second graph (Mean of Y) and the actual values of Y data. Dividing the result by N (number of samples), we can get the variance of the result.

Similarly, can be depicted as the sum of distances between the line in the first graph (Linear Model) and the actual values of Y data. Dividing the result by N, we get the variance of the result.

As the denominator is set as (or the variance of it), we can assume that if the model explains the data well (i.e., if the variation of Y is explained by taking X into account), the value of R-squared will become closer to 1, because the subtracted value on the numerator () will get smaller. Thus as implied from the equation, the R-squared value depicts the extent of variation in Y explained by taking X into account.

For a clear understanding, now consider the following illustrations:

y2 = linearModel.predict(x)

meanline2 = np.mean(y2) + 0\*x

plt.figure(figsize=(10,5))

plt.subplot(1,2,1)

plt.plot(x,y2,'o')

plt.plot(x, fitline, label='Linear Model')

plt.xlabel('X')

plt.ylabel('Y')

plt.legend()

plt.subplot(1,2,2)

plt.plot(x, y2, 'o')

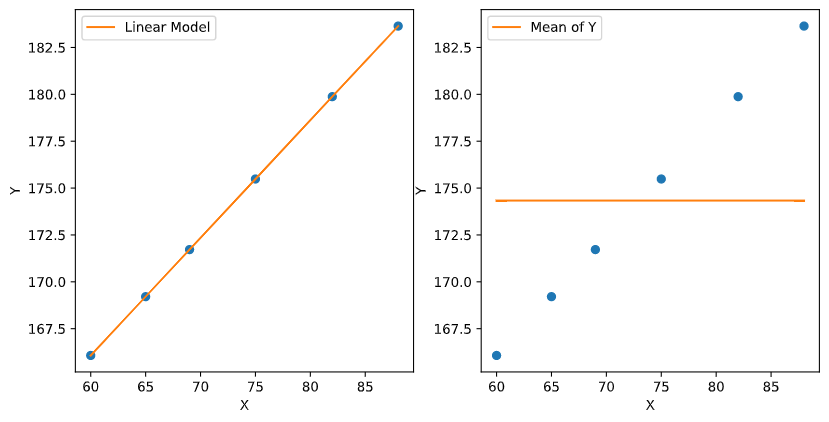
plt.plot(x, meanline2, label='Mean of Y')

plt.xlabel('X')

plt.ylabel('Y')

plt.legend()

plt.show()



What would be the R-squared value for the linear model generated this time (graph 1)? By now, you will know that the R-squared value for this model would be 1, because the subtraction value () is 0 (because there is no distance between each sample and the fitted linear line). Then, this would mean that the variation of Y is 100% explained by taking X into account.

* 1. P-Value

R-squared values are especially beloved in machine-learning as the evaluation method of the model, as “predictions” become the chief objective for machine learning. If you are an economist or a social scientist, however, statistical significance (and the statistical significance of the calculated R-squared value) is another major evaluation criterion, because R-squared (or even adjusted R-squared) value has some blind spots as well. For example, changes in the hours of sleep you have per day may somehow be highly correlated to the changes in the number of cars produced in California, which would result in high R-squared value. However, this result is most likely coming from random contingency, with no actual statistical meaning. This is why, representatively, statisticians and social scientists use P-value as another criterion for the evaluation of a model.

* + 1. Differences between P-value and Probability

A typical mistaken notion regarding P-value comes from confusing between the concept of Probability and P-value. Let us first resolve this misunderstanding by directly calculating both probability and P-value using a coin-toss example.

If you flip a coin two times, the probability of getting two tails would be: outcome of having two tails / the total number of outcomes = 1/4. Thus the probability of getting two tails out of two flips would be 25%.

P-value (or, two-tailed P-value), on the other hand, is calculated by adding up the probability of obtaining both equally probable and more extreme outcomes to the original probability. In this case, therefore, the P-value would be 1/4 + 1/4 = 1/2 (0.5).

* + 1. Null Hypothesis

It is worth mentioning that P-value always comes with hypothesis. As regards to the example above, the hypothesis would be: “the coin used in this coin-toss, which generated the outcome of two tails in a row, is not as ordinary as the other ordinary coins.” If we want our hypothesis to be right, our goal would be to reject the null hypothesis (the opposite of our hypothesis), which would be that the coin used in this coin-toss is just as ordinary as the other coins. The typical threshold in the area of social science for rejecting the null hypothesis is when p-value is smaller than 0.05 (or 0.1, depending on the cases), and is often denoted with \* sign as: [\*p<0.01, \*\*p<0.05, \*\*\*p<0.01] depending on each threshold. In case of the coin-toss example earlier, null hypothesis cannot be rejected since the calculated value is 0.5, which is much larger than the accepted threshold.

* + 1. Probability Density and P-value

Let us generate a probability density function of a normal distribution for better understanding.

stdRv = norm(loc=0, scale=1)

plt.plot(x, stdRv.pdf(x))

plt.axvline(x=4.7, ymax=0.004, color='r', linewidth=20)

plt.axvline(x=4.35, ymax=0.2, color='g', linewidth=2, linestyle=':')

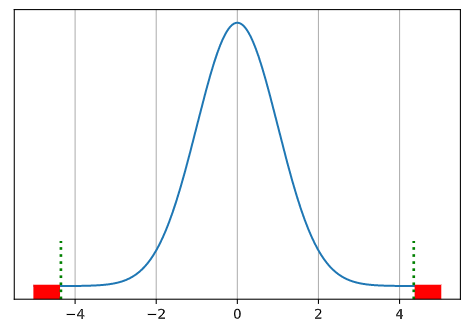
plt.axvline(x=-4.7, ymax=0.004, color='r', linewidth=20)

plt.axvline(x=-4.35, ymax=0.2, color='g', linewidth=2, linestyle=':')

plt.yticks([])

plt.grid()

plt.show()



We have learned previously that the area under the distribution curve as above depicts the probability of getting the corresponding range of X values. Let us suppose that the 99.8% of the area under the curve is between the X values of -4.35 and 4.35. This means that whenever we pick a random float number within the range of total X values, there is a 99.8% chance that the picked number is going to be between the X values of -4.35 and 4.35.

Let us assume that we observed the X value of -4.35. What we want to know is if this X value is coming from the distribution above with average X value of 0, or:

stdRv = norm(loc=0, scale=1)

x = np.linspace(-5, 5, 1000)

x2 = np.linspace(-9.35, 0.65, 1000)

plt.plot(x, stdRv.pdf(x))

plt.axvline(x=4.7, ymax=0.004, color='r', linewidth=20)

plt.axvline(x=4.35, ymax=0.2, color='g', linewidth=2, linestyle=':')

plt.axvline(x=-4.7, ymax=0.004, color='r', linewidth=20)

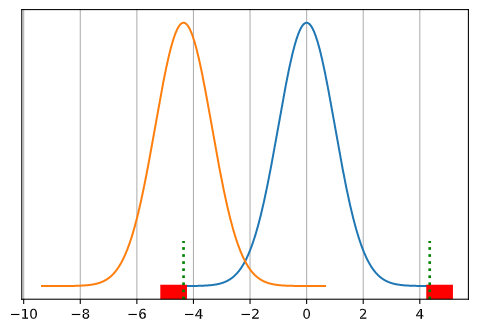
plt.axvline(x=-4.35, ymax=0.2, color='g', linewidth=2, linestyle=':')

plt.plot(x2, stdRv.pdf(x))

plt.yticks([])

plt.grid()

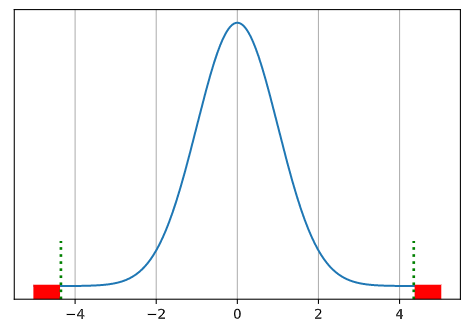
plt.show()



an alternative distribution such as the one above (orange). The null hypothesis would be that the observed X value -4.35 is indeed coming from the original distribution (blue). We know that the area marked as red (less than -4.35 and larger than 4.35) is 0.2% of the total area under the curve. As calculated earlier, P-value is obtained by adding up the possibilities of having equally probable and more extreme outcomes (and thus, the area of both ends colored as red, i.e., X <= -4.35 and X >= 4.35). Thus we get the P-value of 0.002 (0.001 + 0.001), which is considered as statistically significant P-value (\*\*\*p<0.01). This implies that we can reject the null hypothesis and conclude that the observed X value is more likely coming from other forms of distribution such as the orange one.

* + 1. Applications of P-Value

Now that we understand the whole concepts and mechanism of P-value, what would be the example usage of it? Consider the distribution graph we drew earlier:



Now, suppose that the X values indicate the average GDP growth rates of countries at year 2020. Using the same X example we used earlier, let us assume that country A exhibited the growth rate of -4.35 (%). The P-value for this country’s growth rate is less than 0.01. Thus given the distribution of GDP growth rates of all countries, we can statistically state that the country A’s GDP growth rate of -4.35 (%) was highly unusual, compared to other countries, at year 2020 (which then also calls for various other analyses to investigate why country A’s GDP growth rate was unusually low compared to other countries at year 2020).

Now that we have covered the concepts of R2 and P-value, some enthusiastic readers might wonder: “what if the p-value is below 0.05 while the R2 score is too low (e.g., below 10%)?” As implied so far, P-value should always be considered as the priority when it comes to the regression equation, because it indicates the very **significance** of the equation per se. R2, on the other hand, indicates the **ratio of contribution (i.e., explanatory power)**. As these concepts imply, the independent variable and the equation created with that variable with significant p-value indicates that the equation is statistically relevant (significant); having confirmed that the variable used for this equation is statistically relevant, the fact that the R2 score is low indicates that additional independent variables are required to increase the explanatory power of the relevant equation.

* + 1. Theoretical Backbone of Inferential Statistics, P-Value, and Significance Levels:

Central Limit Theorem

Regardless of the shapes and types of the population distribution, if the size of sample is sufficiently large and the random variables satisfy the i.i.d. supposition, the distribution of sample ***means*** will be approximately normally distributed. This becomes the theoretical basis for significance tests in inferential statistics. P-value and the level of significance (e.g., 5%, 1%), for instance, are validated under the premise that the distribution of sample means approximately forms normal distribution. Throughout creating a sample simulation and visualization, as follows, it is possible to check if the distribution of sample means approximates normal distribution. Let us suppose a uniform distribution for our sample:

#Demonstration of the Central Limit Theorem

from scipy.stats import uniform

rv = uniform()

rv.rvs(3)

n = 20000

m = 10000

mean = []

for \_ in range(m):

    x = rv.rvs(n)

    mean.append(x.mean())

import seaborn as sns

import matplotlib.pyplot as plt

sns.distplot(mean)

plt.grid()

plt.show()

output:

