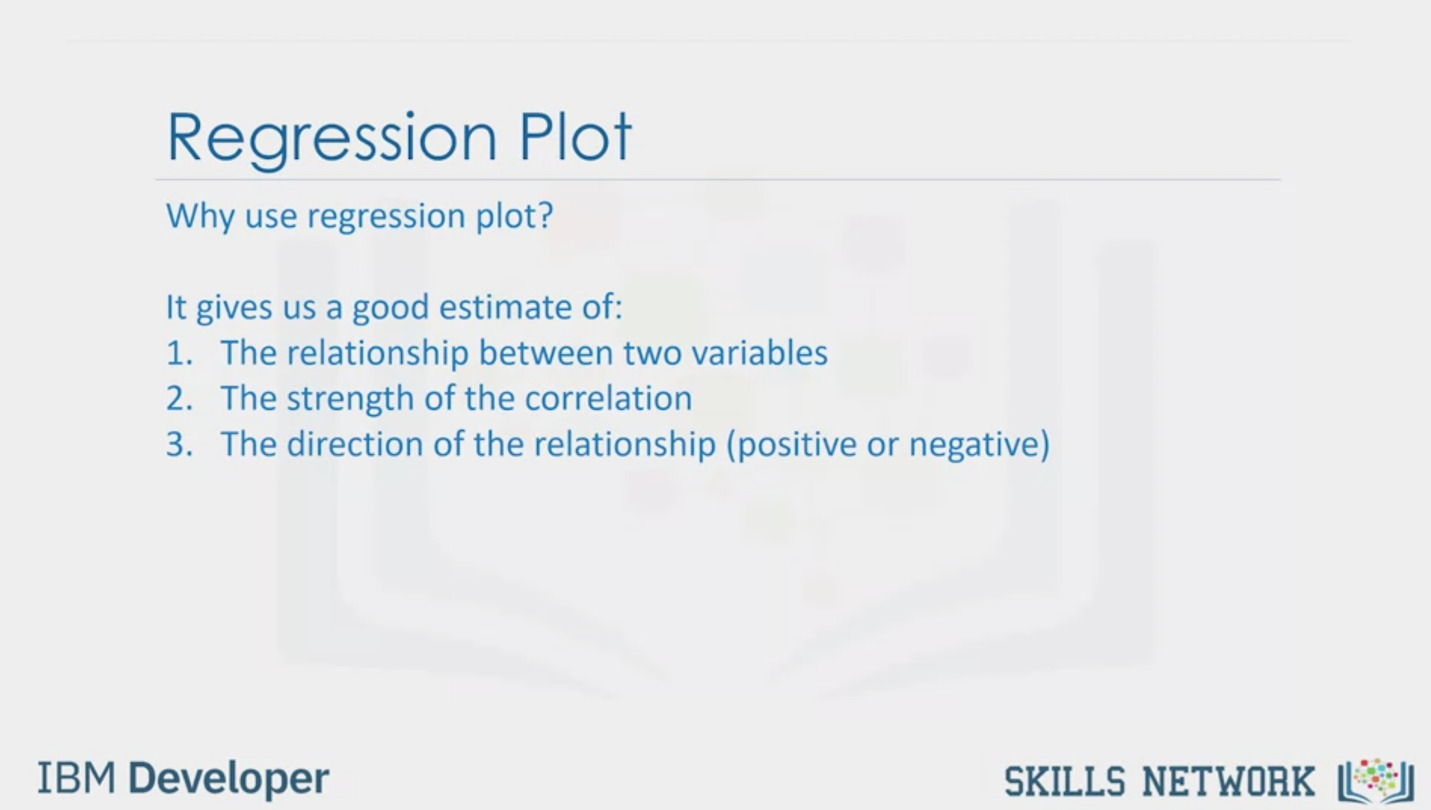


Model Evaluation

Using Visualization

IBM Developer

SKILLS NETWORK



Regression Plot

Why use regression plot?

It gives us a good estimate of:

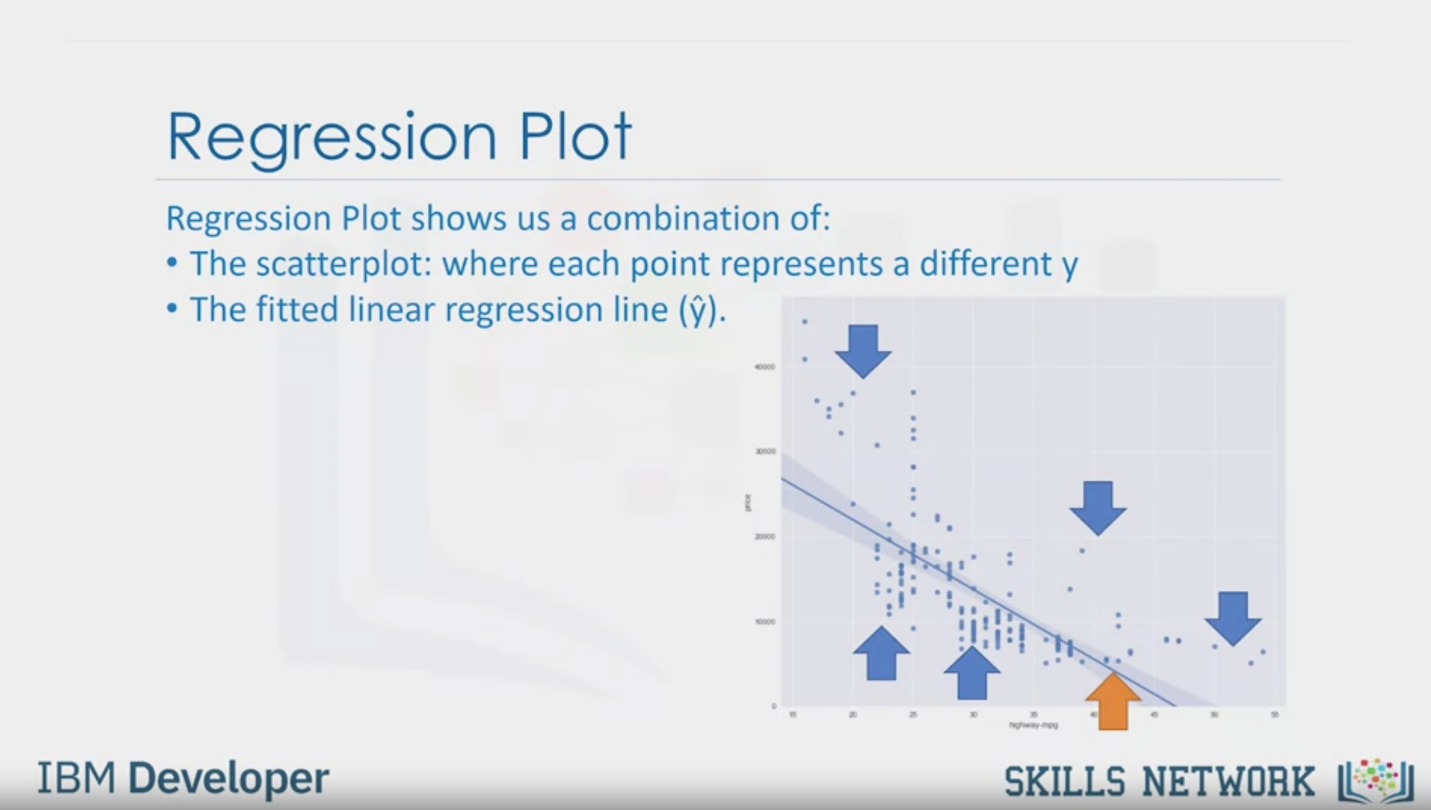
1. The relationship between two variables

2. The strength of the correlation

3. The direction of the relationship (positive or negative)

IBM Developer

SKILLS NETWORK



Regression Plot

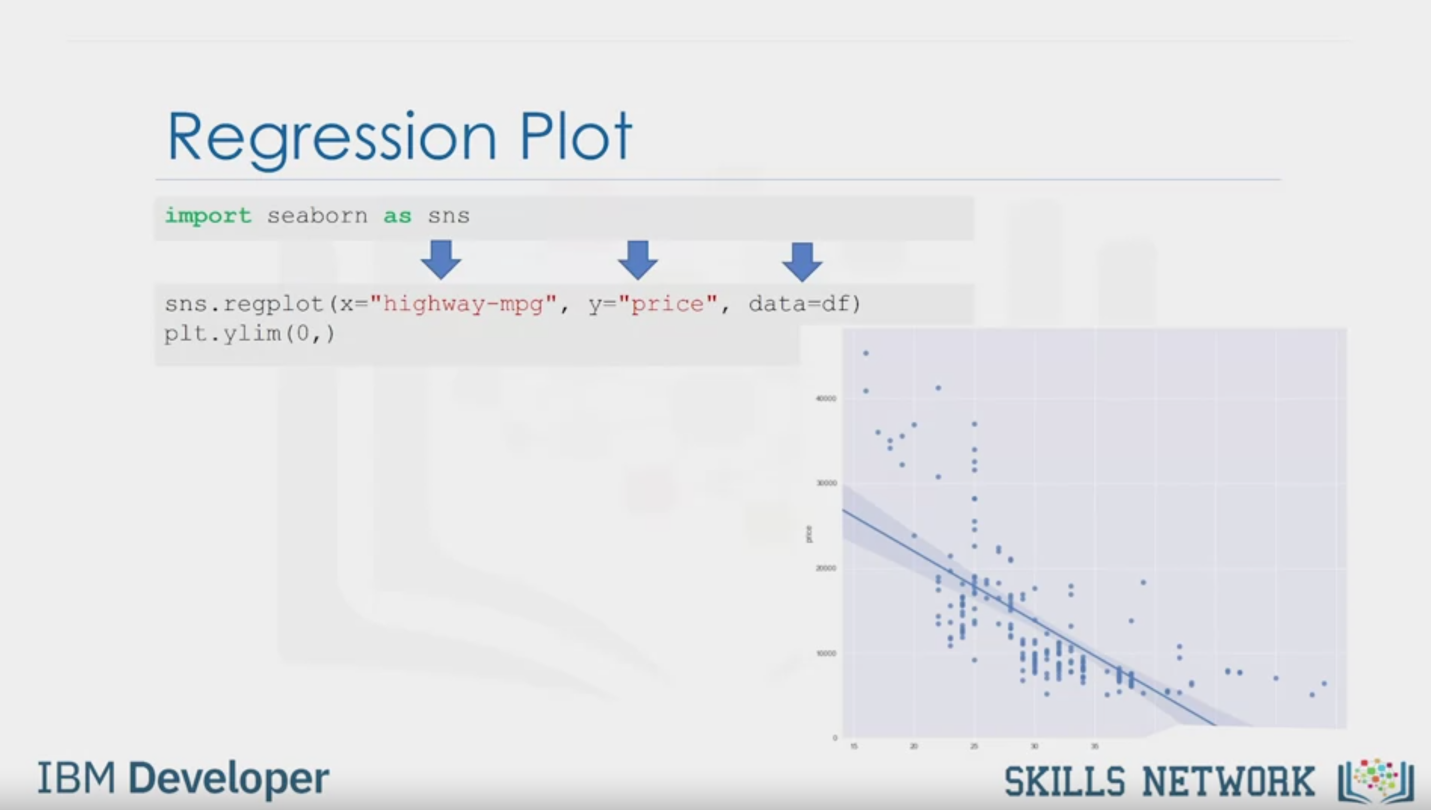
Regression Plot shows us a combination of:

• The scatterplot: where each point represents a different y

• The fitted linear regression line ().

IBM Developer

SKILLS NETWORK



Regression Plot

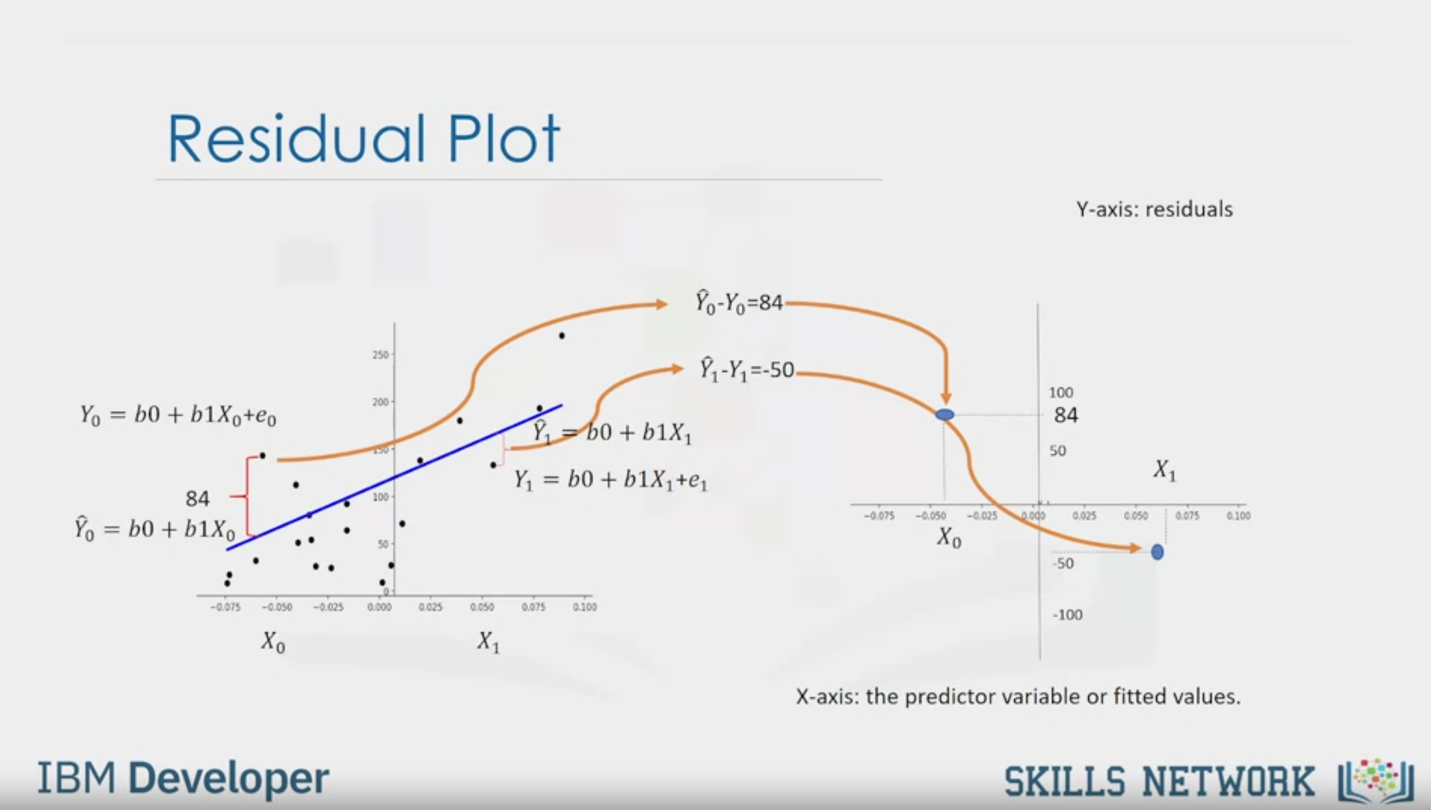
import seaborn as sns

sns.regplot (x="highway-mpg", y="price", data=df)

plt.ylim(0,)

IBM Developer

SKILLS NETWORK



Residual Plot

Y-axis: residuals

Y0 = b0 + b1X0 + e0

84

0  = b0 + b1X0

X0

X1

1  = b0 + b1X1

Y1 = b0 + b1X0 + e1

0  - Y0 = 84

1  - Y1 = -50

X0

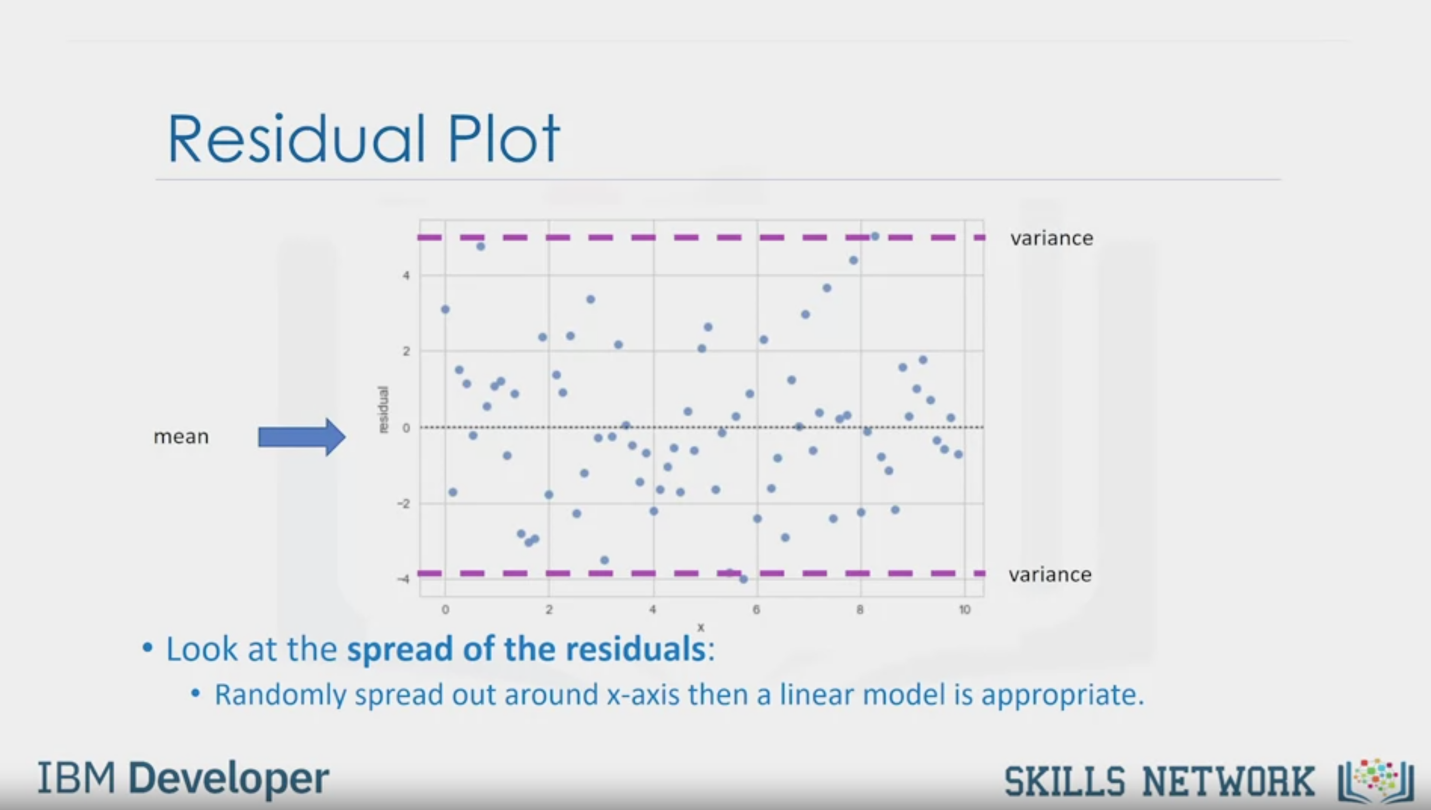
84

X1

X-axis: the predictor variable or fitted values.

IBM Developer

SKILLS NETWORK



Residual Plot

variance

mean

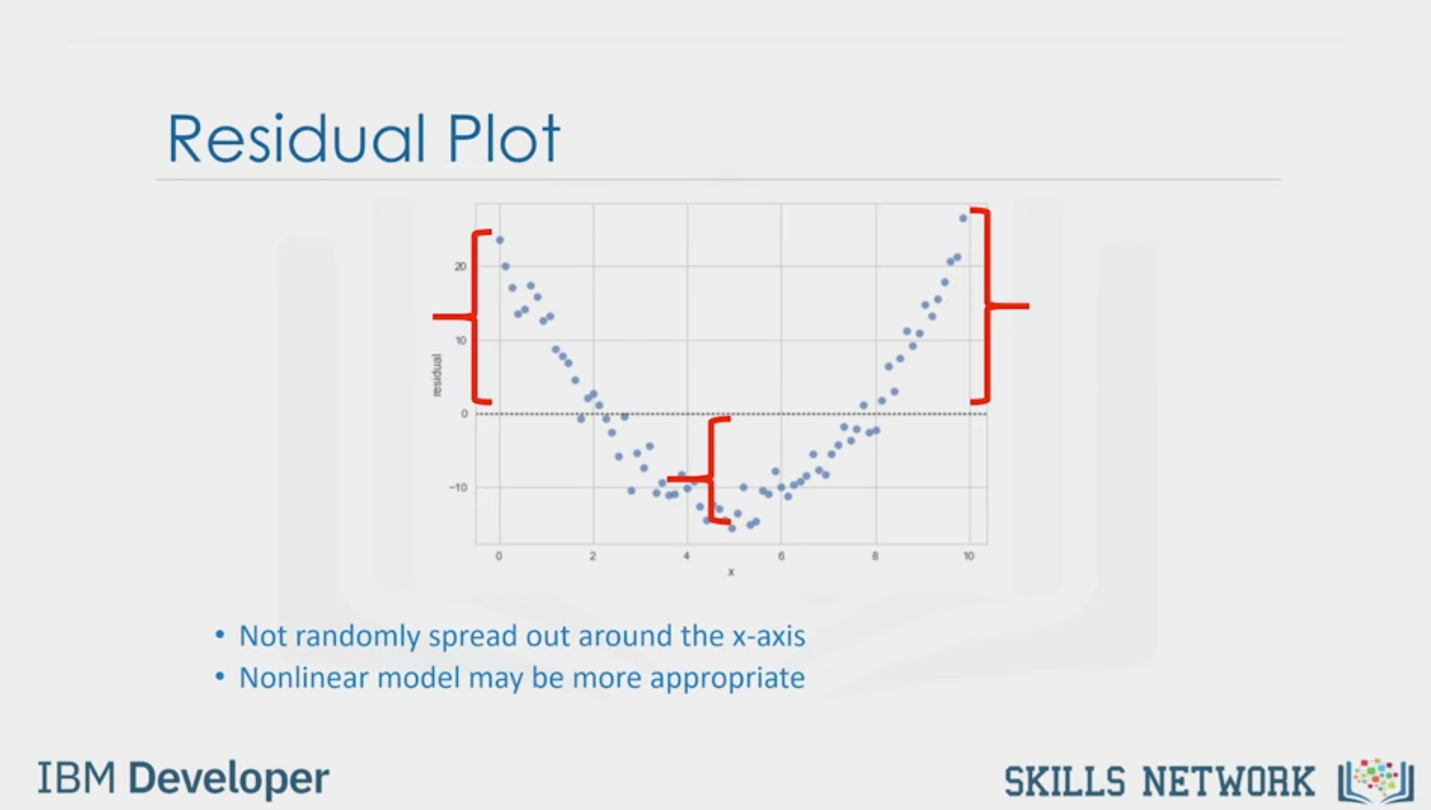
variance

• Look at the spread of the residuals:

• Randomly spread out around -axis then a linear model is appropriate.

IBM Developer

SKILLS NETWORK



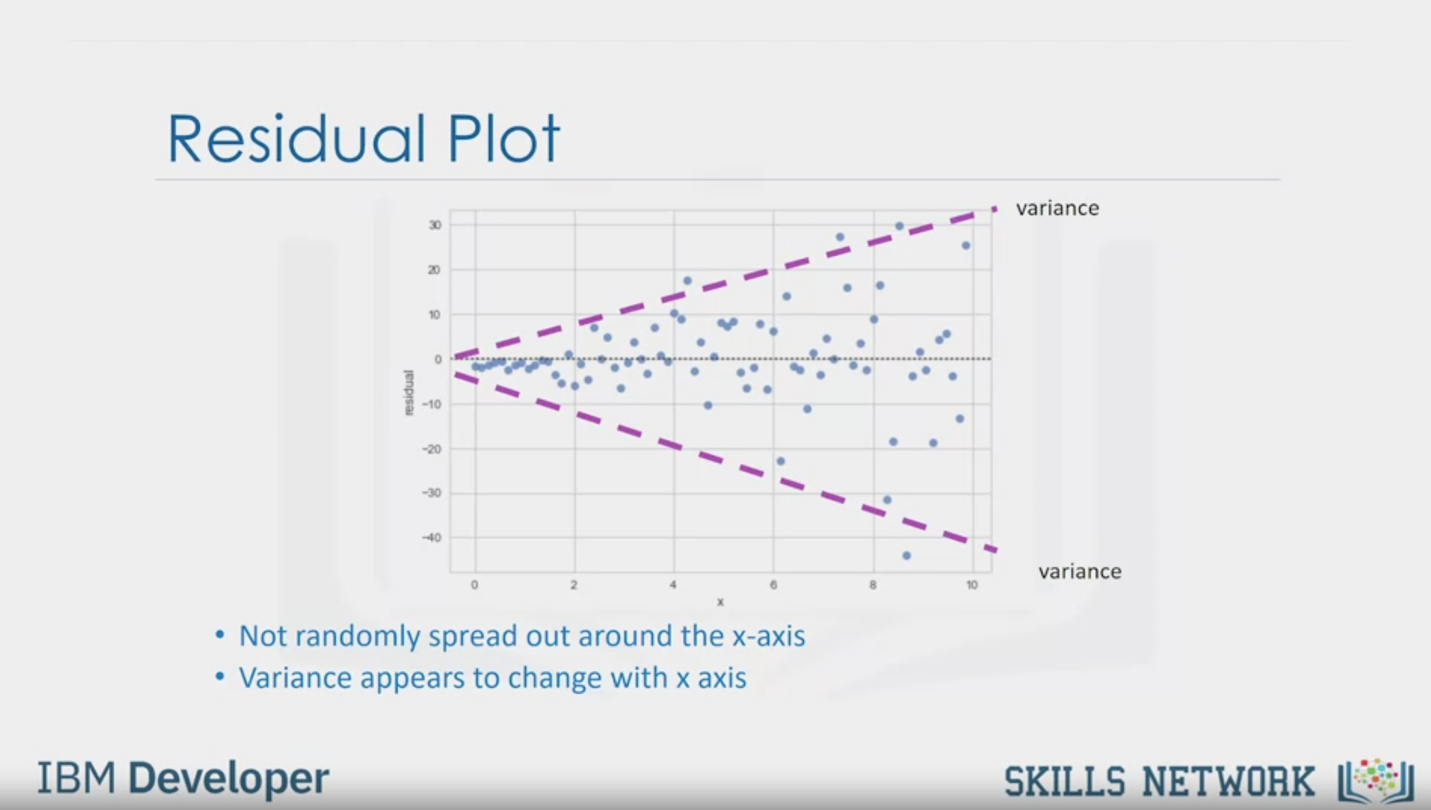
Residual Plot

• Not randomly spread out around the x-axis

• Nonlinear model may be more appropriate

IBM Developer

SKILLS NETWORK



Residual Plot

variance

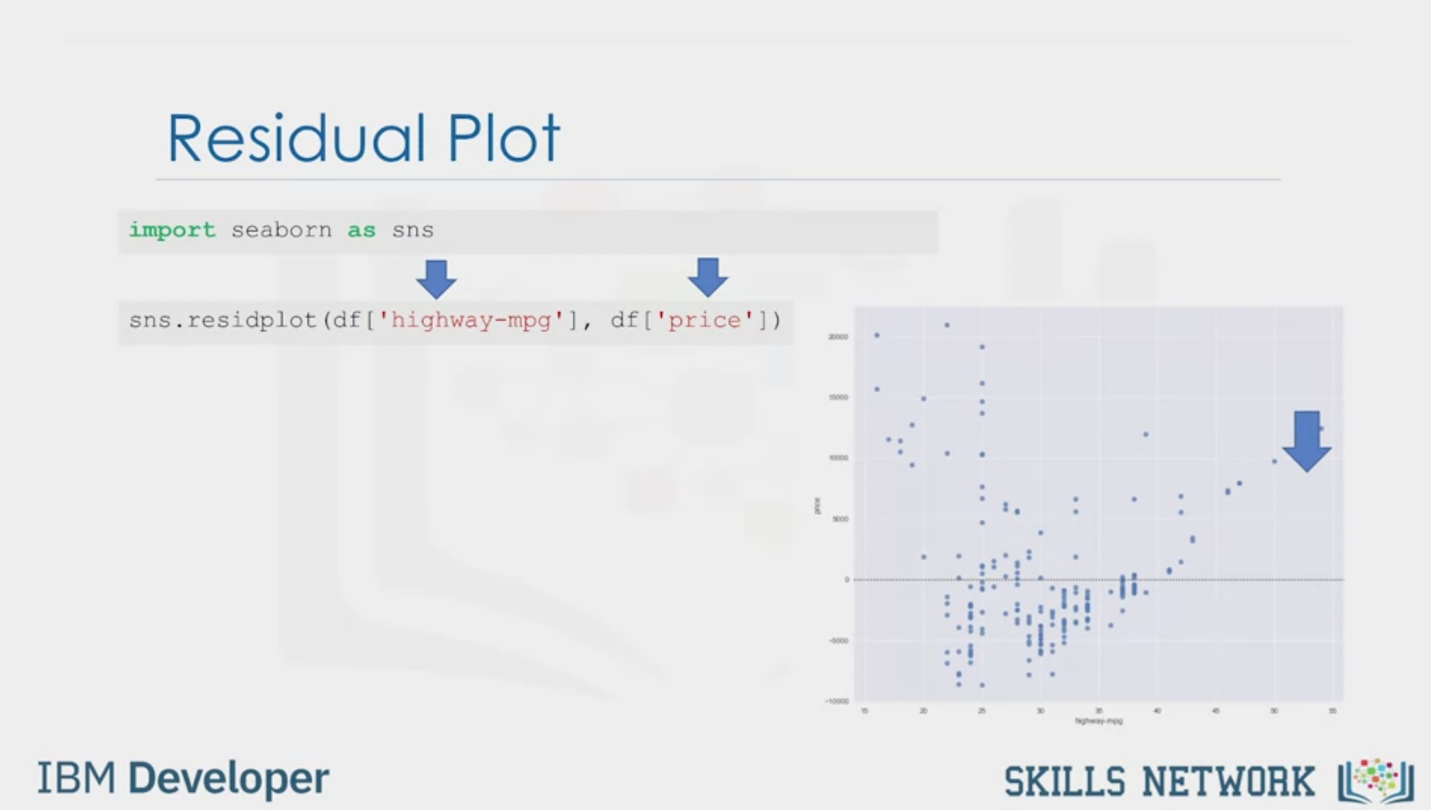
variance

• Not randomly spread out around the x-axis

• Variance appears to change with x axis

IBM Developer

SKILLS NETWORK



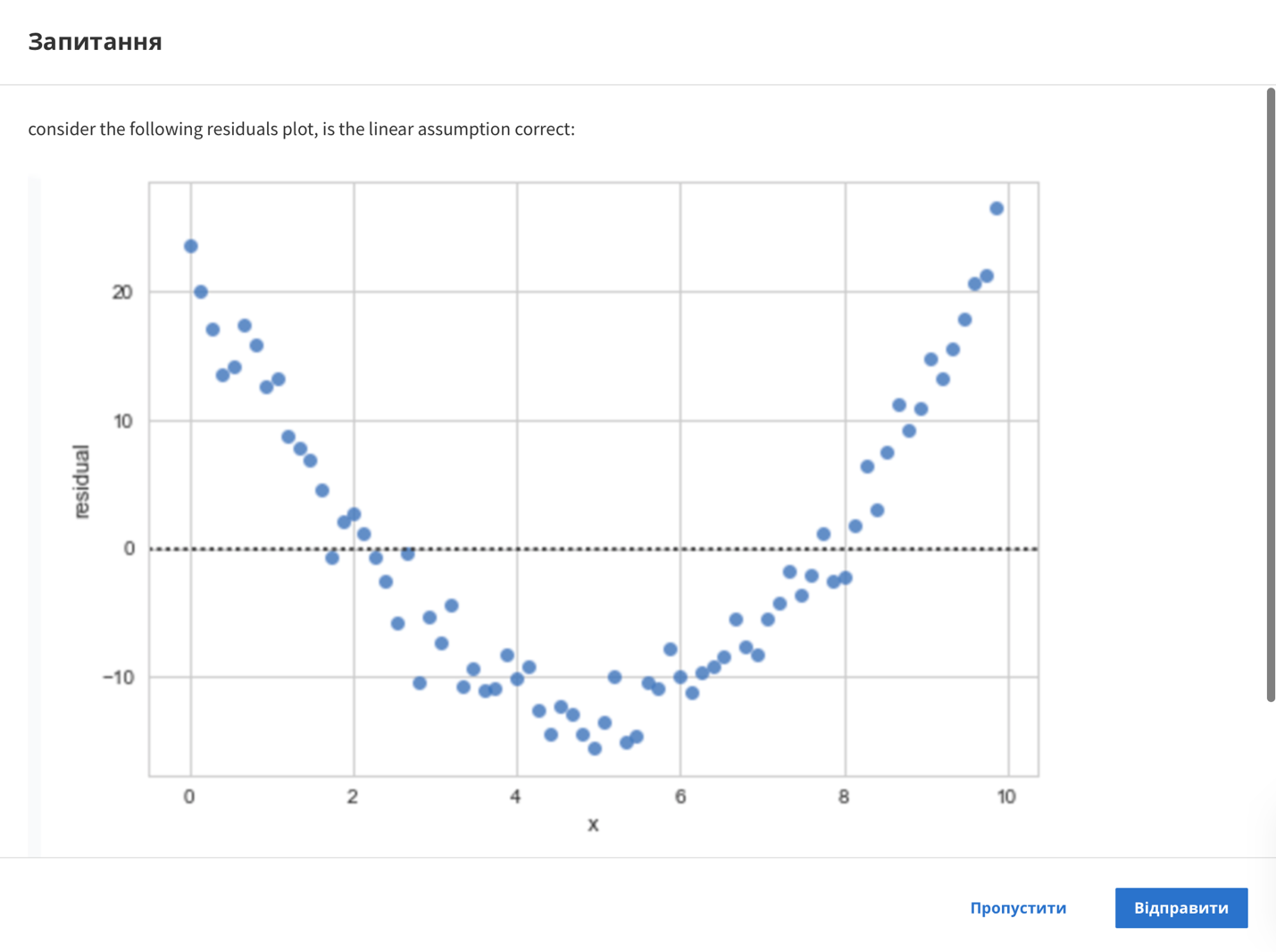
Residual Plot

import seaborn as sns

sns.residplot (df ['highway-mpg'], df['price"])

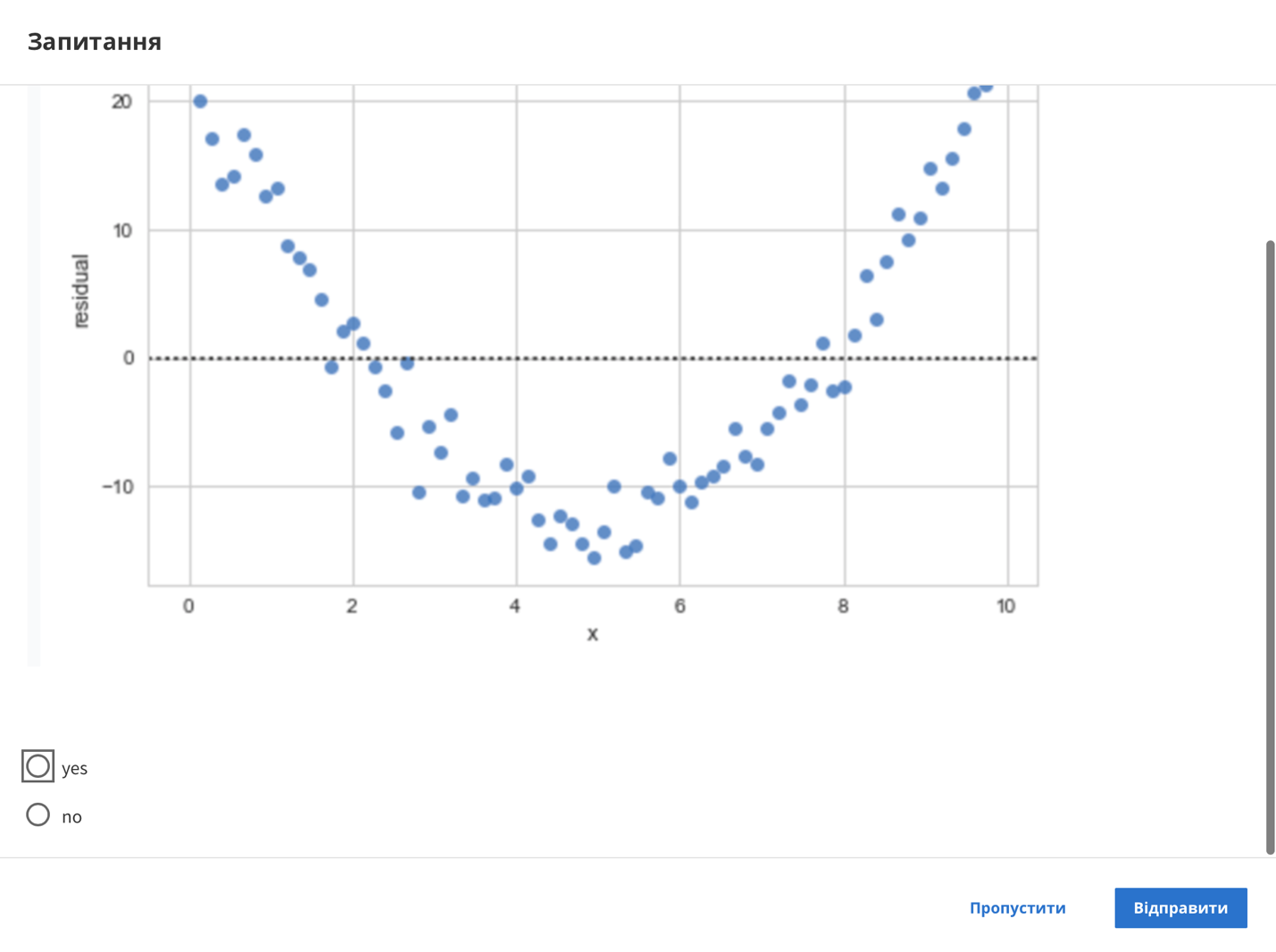
IBM Developer

SKILLS NETWORK



Question

consider the following residuals plot, is the linear assumption correct:

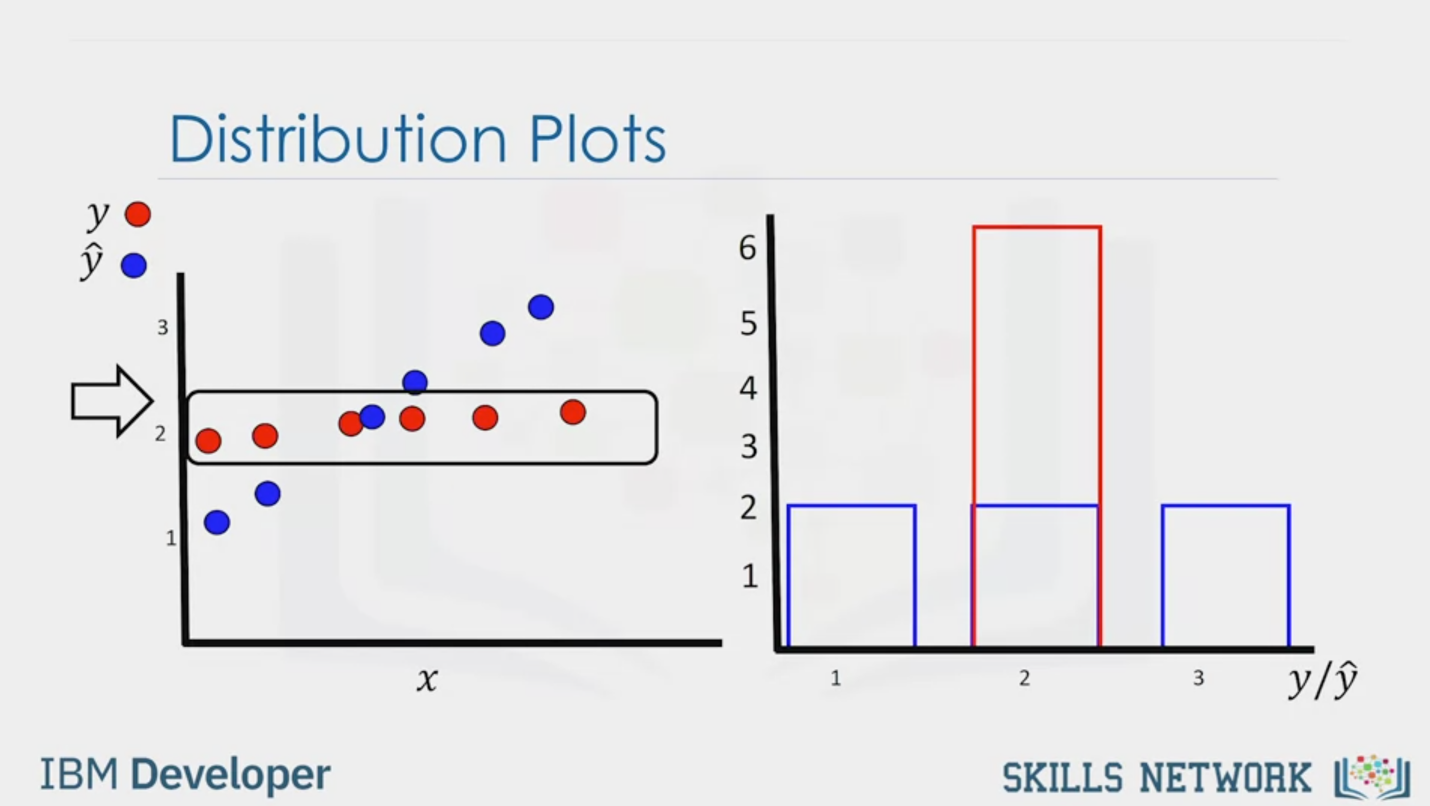


yes

no

Skip

Send



Distribution Plots

y

3

2

1

X

6

5

4

3

2

1

1

2

3

y/

IBM Developer

SKILLS NETWORK



Distribution Plots

1

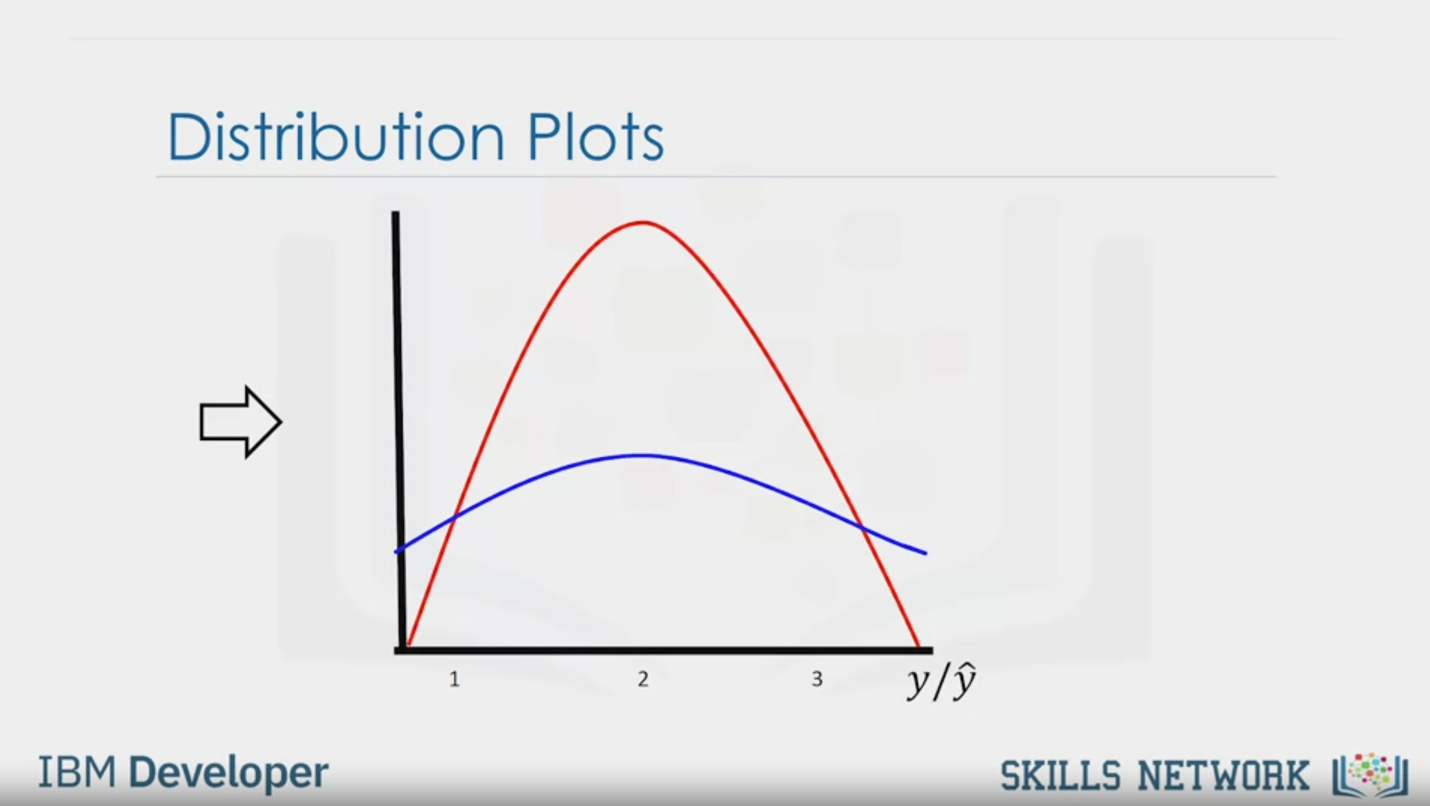
2

3

y/

IBM Developer

SKILLS NETWORK



Distribution Plots

1

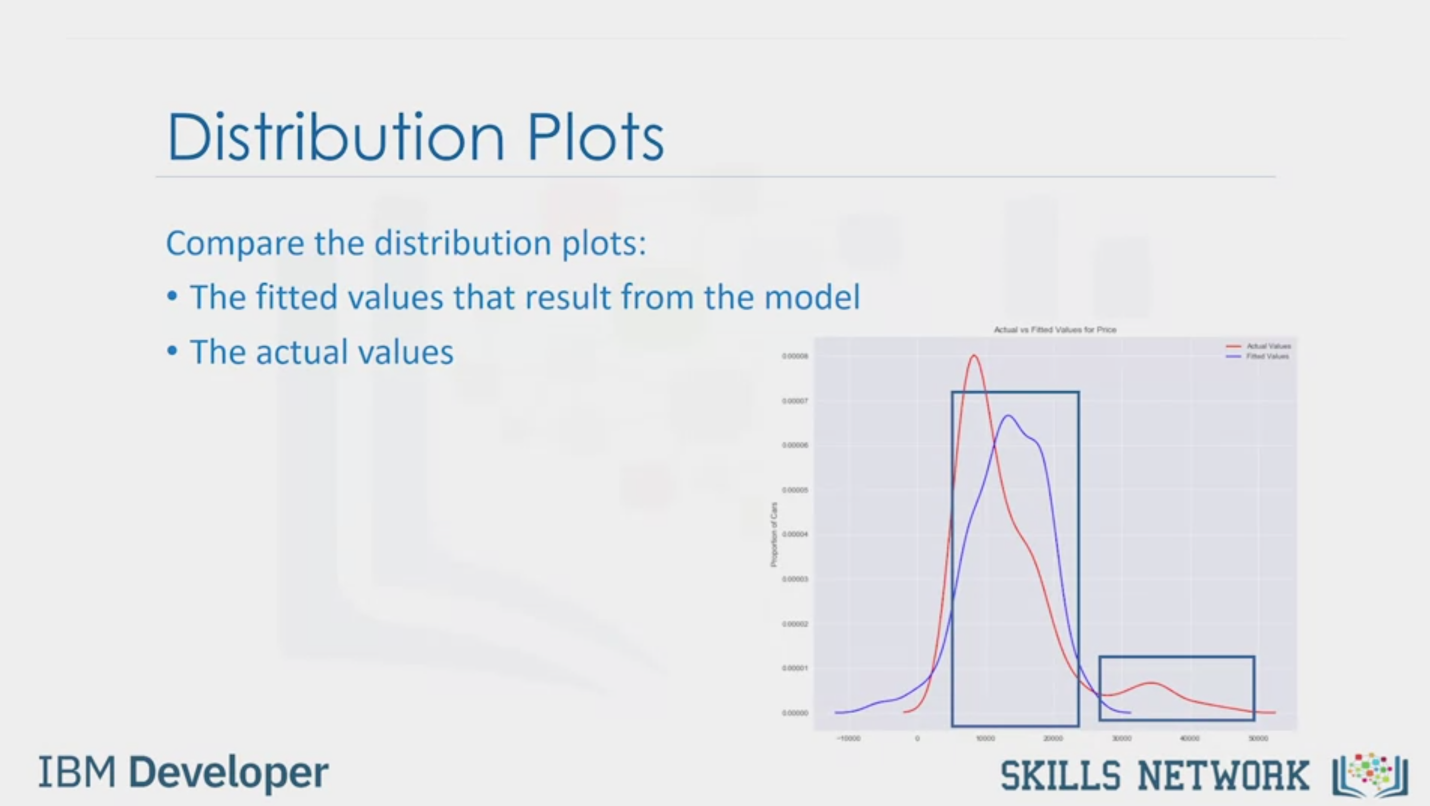
2

3

y/

IBM Developer

SKILLS NETWORK



Distribution Plots

Compare the distribution plots:

• The fitted values that result from the model

• The actual values

IBM Developer

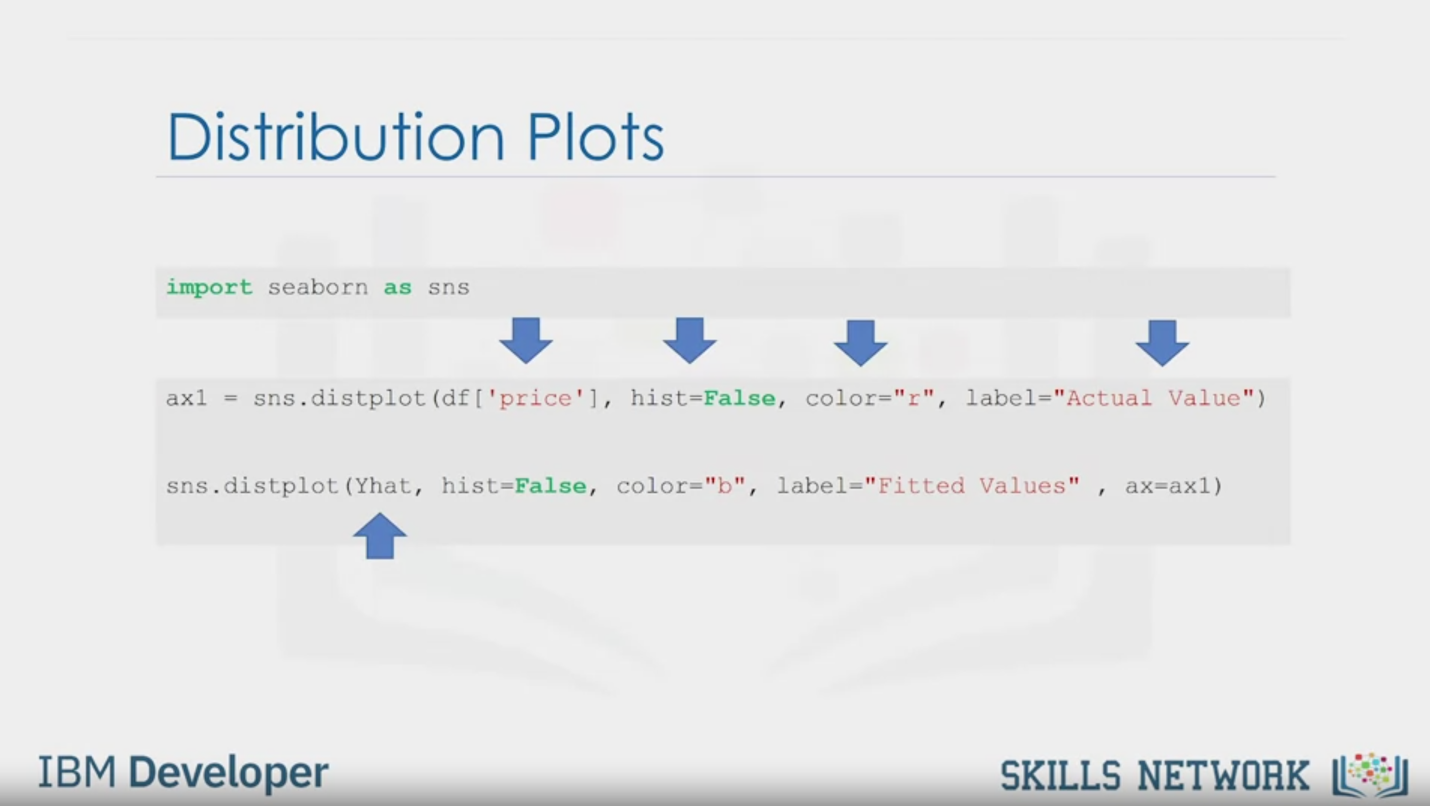
SKILLS NETWORK



MLR - Distribution Plots

IBM Developer

SKILLS NETWORK



Distribution Plots

import seaborn as sns

axl = sns.distplot (df ['price'], hist=False, color="r", label="Actual Value")

sns.distplot (Yhat, hist=False, color="b", label="Fitted Values" , ax=ax1)

IBM Developer

SKILLS NETWORK



SKILLS NETWORK

IBM Developer

In this video, we'll look at Model Evaluation using Visualization.

Regression plots are a good estimate of the relationship between two variables,

the strength of the correlation,

and the direction of the relationship (positive or negative).

The horizontal axis is the independent variable.

The vertical axis is the dependent variable.

Each point represents a different target point.

The fitted line represents the predicted value.

There are several ways to plot a regression plot.

A simple ways to use regplot from the seaborn library.

First, "import seaborn."

Then use the "regplot" function.

The parameter x is the name of the column that

contains the independent variable or feature.

The parameter y, contains the name of the column that

contains the name of the dependent variable or target.

The parameter data is the name of the dataframe.

The result is given by the plot.

The residual plot represents the error between the actual value.

Examining the predicted value and actual value we see a difference.

We obtain that value by subtracting the predicted value,

and the actual target value.

We then plot that value on

the vertical axis with the independent variable as the horizontal axis.

Similarly, for the second sample,

we repeat the process.

Subtracting the target value from the predicted value.

Then plotting the value accordingly.

Looking at the plot gives us some insight into our data.

We expect to see the results to have zero mean,

distributed evenly around the x axis with similar variance.

There is no curvature.

This type of residual plot suggests a linear plot is appropriate.

In this residual plot, there is a curvature.

The values of the error change with x.

For example, in the region,

all the residual errors are positive.

In this area, the residuals are negative.

In the final location,

the error is large.

The residuals are not randomly separated.

This suggests the linear assumption is incorrect.

This plot suggests a nonlinear function.

We will deal with this in the next section.

In this plot, we see the variance of the residuals increases with x.

Therefore, our model is incorrect.

We can use seaborn to create a residual plot.

First, "import seabourn."

We use the "residplot" function.

The first parameter is a series of dependent variable or feature.

The second parameter is a series of dependent variable or target.

We see in this case, the residuals have a curvature.

A distribution plot counts the predicted value versus the actual value.

These plots are extremely useful for visualizing

models with more than one independent variable or feature.

Let's look at a simplified example.

We examined the vertical axis.

We then count and plot the number of

predicted points that are approximately equal to one.

We then, count and plot the number of

predicted points that are approximately equal to two.

We repeat the process.

For predicted points, they are approximately equal to three.

Then we repeat the process for the target values.

In this case, all the target values are approximately equal to two.

The values of the targets and predicted values are continuous.

A histogram is for discrete values.

Therefore, pandas will convert them to a distribution.

The vertical axis is scaled to make the area under the distribution equal to one.

This is an example of using a distribution plot.

The dependent variable or feature is price.

The fitted values that result from the model are in blue.

The actual values are in red.

We see the predicted values for prices in the range from 40,000 to 50,000 are inaccurate.

The prices in the region from 10,000 to 20,000 are much closer to the target value.

In this example, we use multiple features or independent variables.

Comparing it to the plot on the last slide,

we see predicted values are much closer to the target values.

Here's the code to create a distribution plot.

The actual values are used as a parameter.

We wanted distribution instead of a histogram.

So we want the hist parameters set to false.

The color is red. The label is also included.

The predicted values are included for the second plot.

The rest of the parameters are set accordingly.