Regression analysis with scikit-learn

HAS Tools - Sept 29

DISCLAIMER

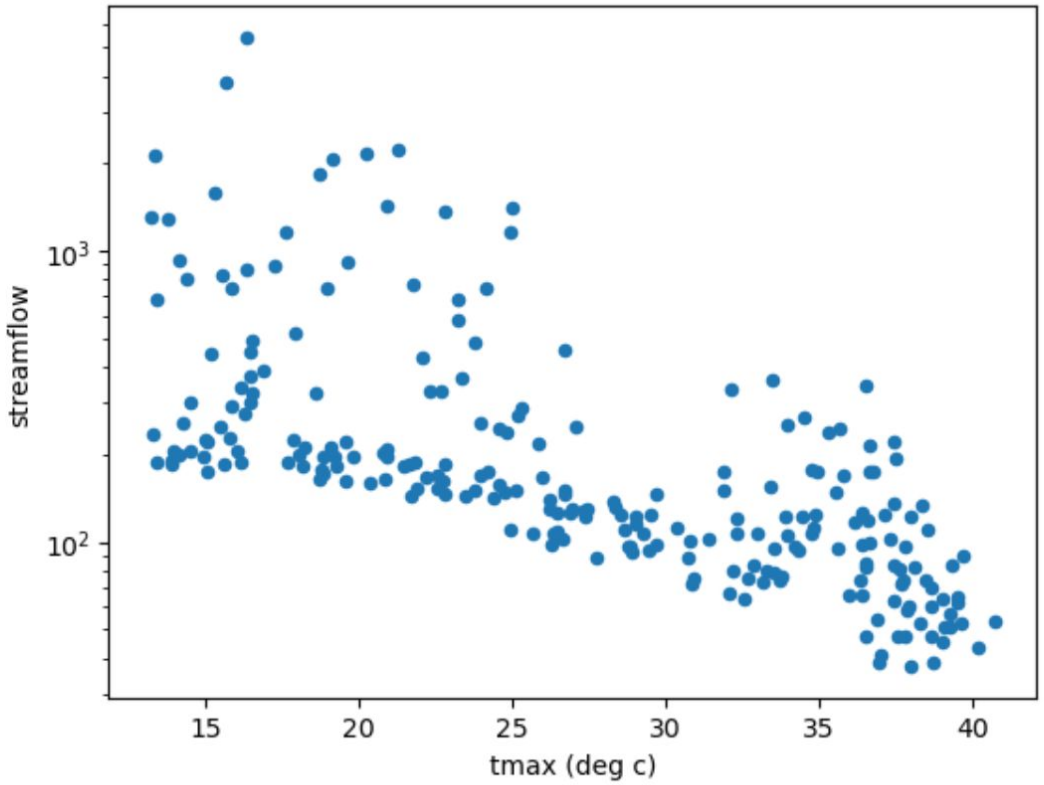
We are playing fast and loose with some stats today, please be sure to follow up on formal methods either by taking some real stats classes or using prior knowledge!!!

Where we left off on Tuesday

You saw how to load data from the USGS streamflow database for the Verde River

You also learned how to apply the same technique to the DayMet dataset, which lets us get things like temperature and precipitation

We finished up by making this plot



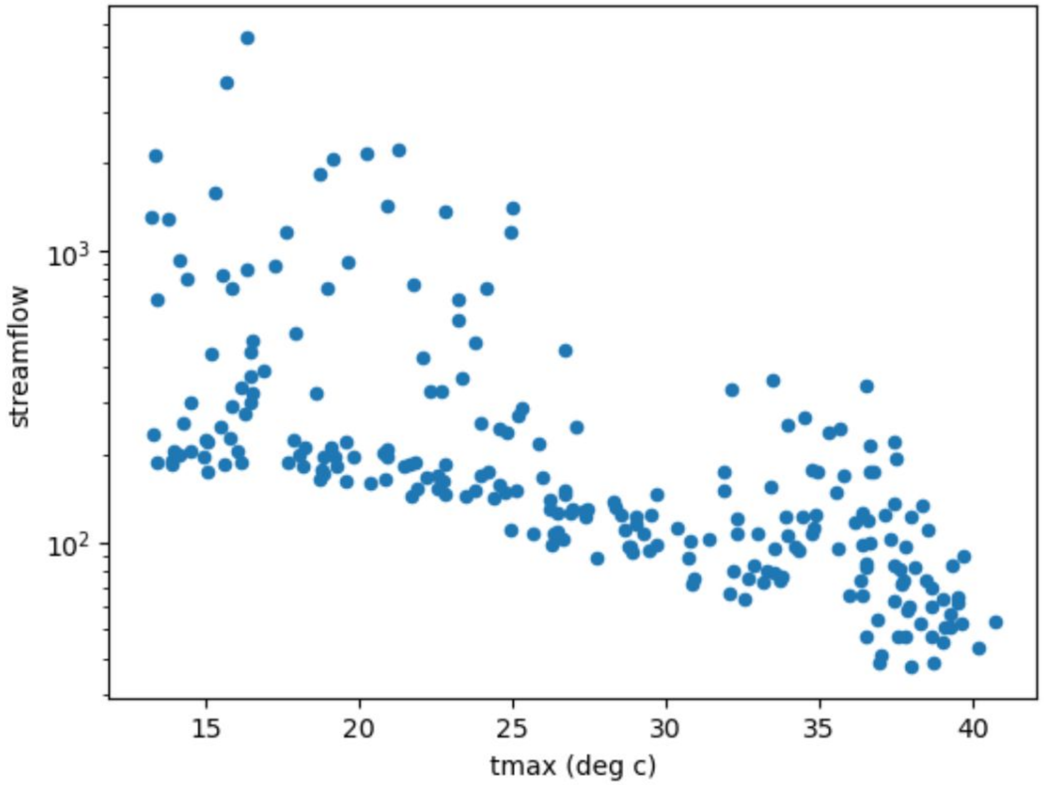
Where we left off on Tuesday

You saw how to load data from the USGS streamflow database for the Verde River

You also learned how to apply the same technique to the DayMet dataset, which lets us get things like temperature and precipitation

We finished up by making this plot

Clearly there's some relationship here, but can we quantify it?



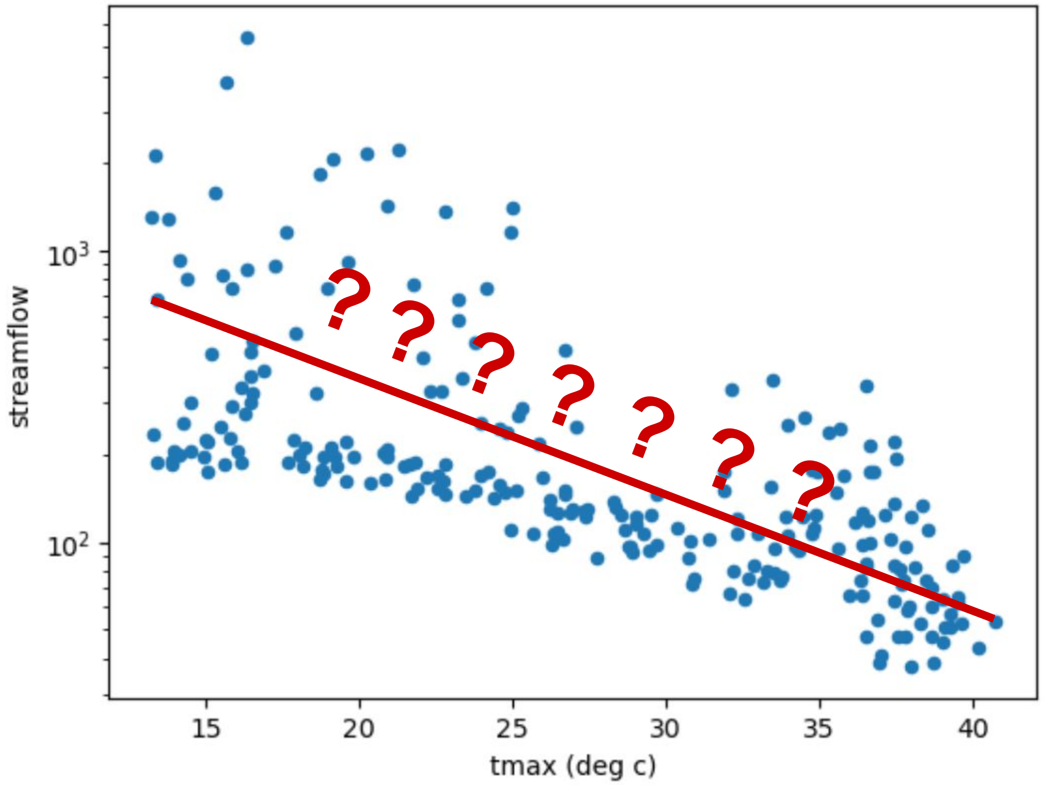
Where we left off on Tuesday

You saw how to load data from the USGS streamflow database for the Verde River

You also learned how to apply the same technique to the DayMet dataset, which lets us get things like temperature and precipitation

We finished up by making this plot

Clearly there's some relationship here, but can we quantify it?



Enter linear regression

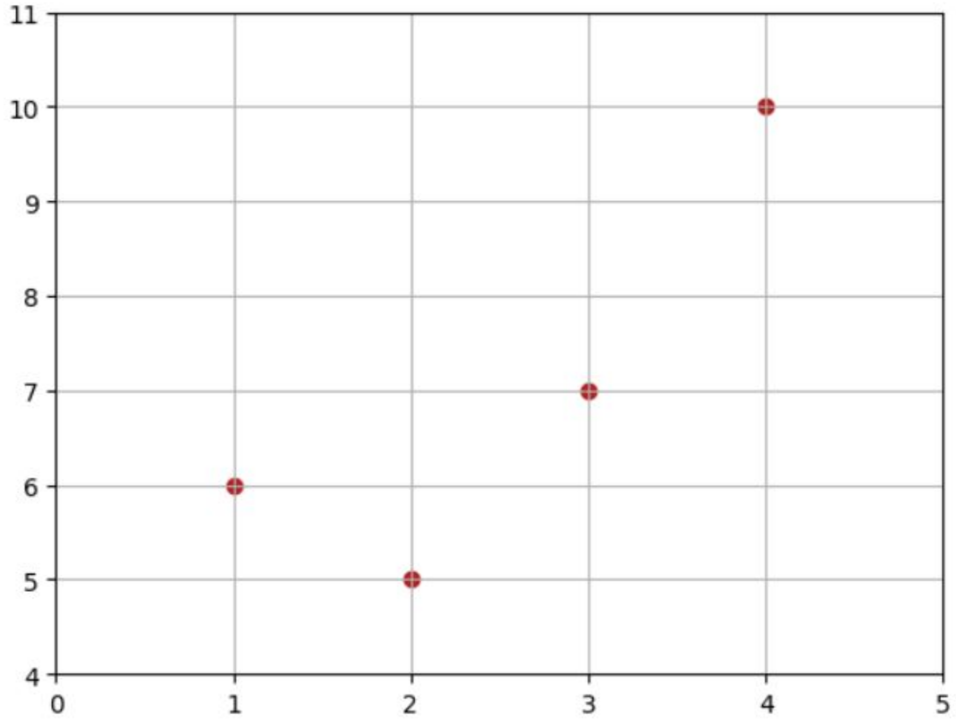
We're going to do just a little bit of theory here before

Suppose we have data with:

- `x=[1,2,3,4]`
- `y=[6,5,7,10]`

We will call x the "explanatory variable"

And we call y the "response variable'



Example courtesy wikipedia: https://en.wikipedia.org/wiki/Linear_least_squares

So how do we go about fitting a line through the data?

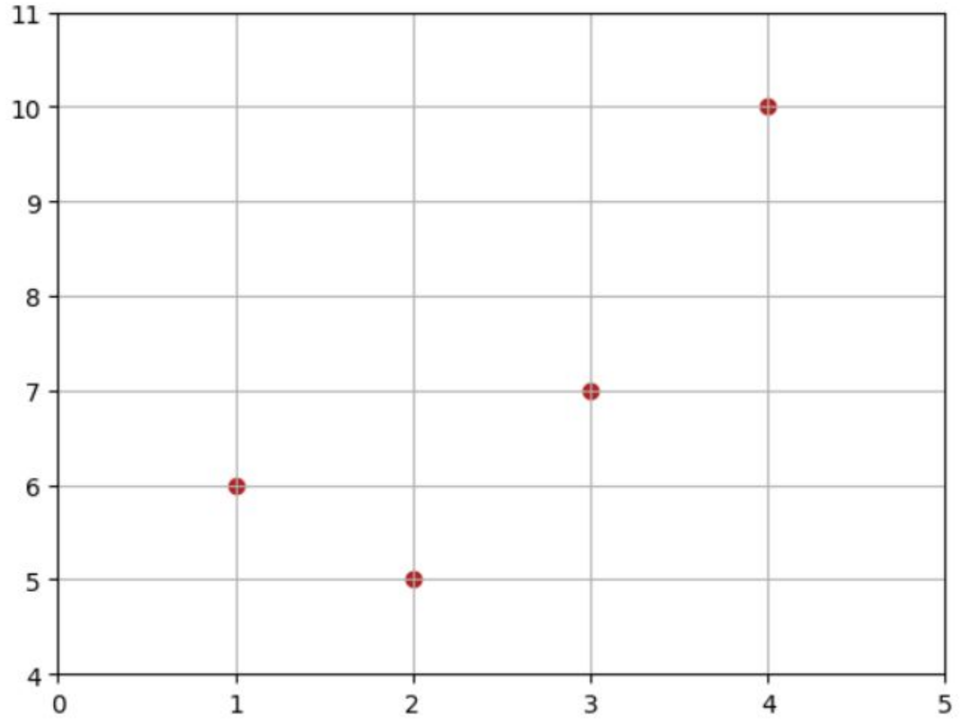
Recall that the equation for a line is:

$$y = \beta_2 x + \beta_1$$

Where:

β_2 Is the slope

β_1 Is the intercept



So how do we go about fitting a line through the data?

Recall that the equation for a line is:

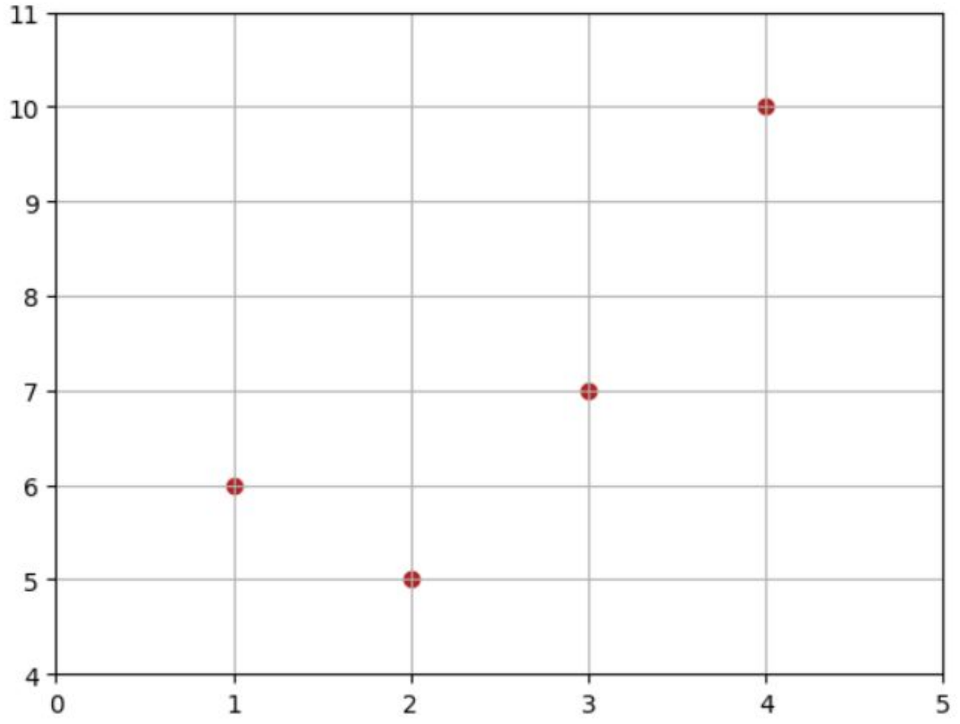
$$y = \beta_2 x + \beta_1$$

Where:

β_2 Is the slope

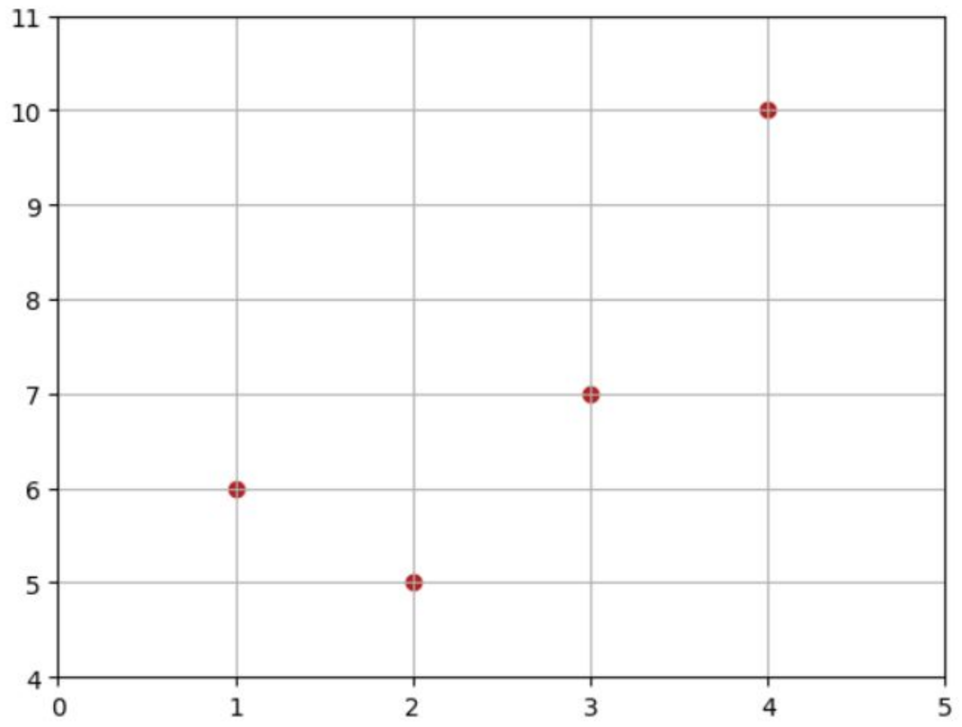
β_1 Is the intercept

r Is a residual term



Now we can frame things into a system of equations

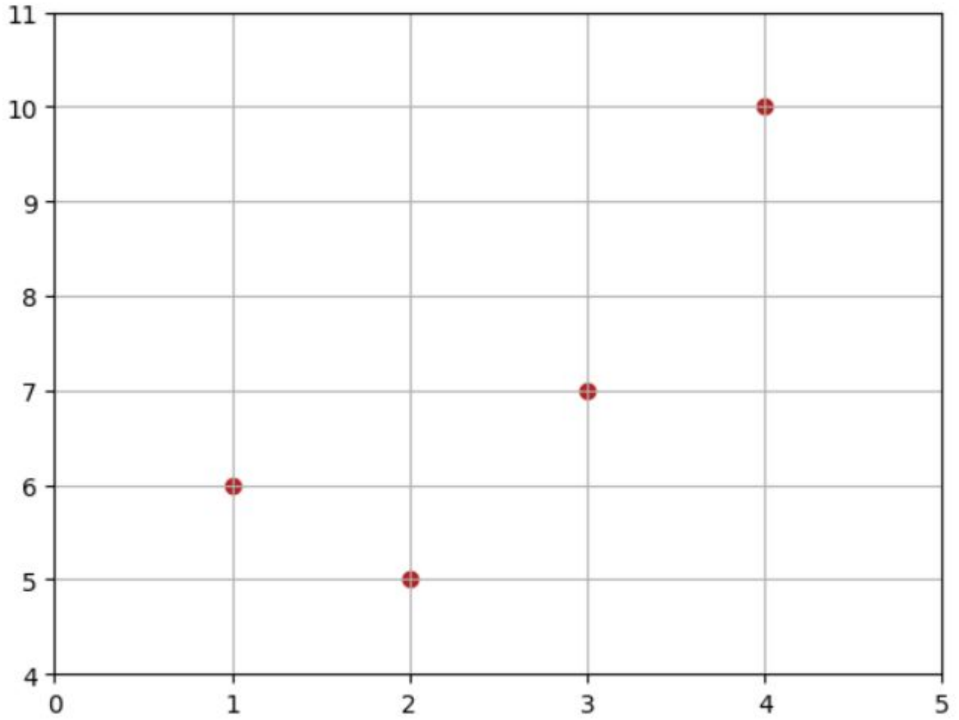
$$\begin{aligned}
\beta_1 + 1\beta_2 + r_1 &= 6 \\
\beta_1 + 2\beta_2 + r_2 &= 5 \\
\beta_1 + 3\beta_2 + r_3 &= 7 \\
\beta_1 + 4\beta_2 + r_4 &= 10
\end{aligned}$$



Now we can frame things into a system of equations

We want the value of these

$$\begin{aligned}
\boxed{\beta_1} + 1\boxed{\beta_2} + r_1 &= 6 \\
\beta_1 + 2\beta_2 + r_2 &= 5 \\
\beta_1 + 3\beta_2 + r_3 &= 7 \\
\beta_1 + 4\beta_2 + r_4 &= 10
\end{aligned}$$

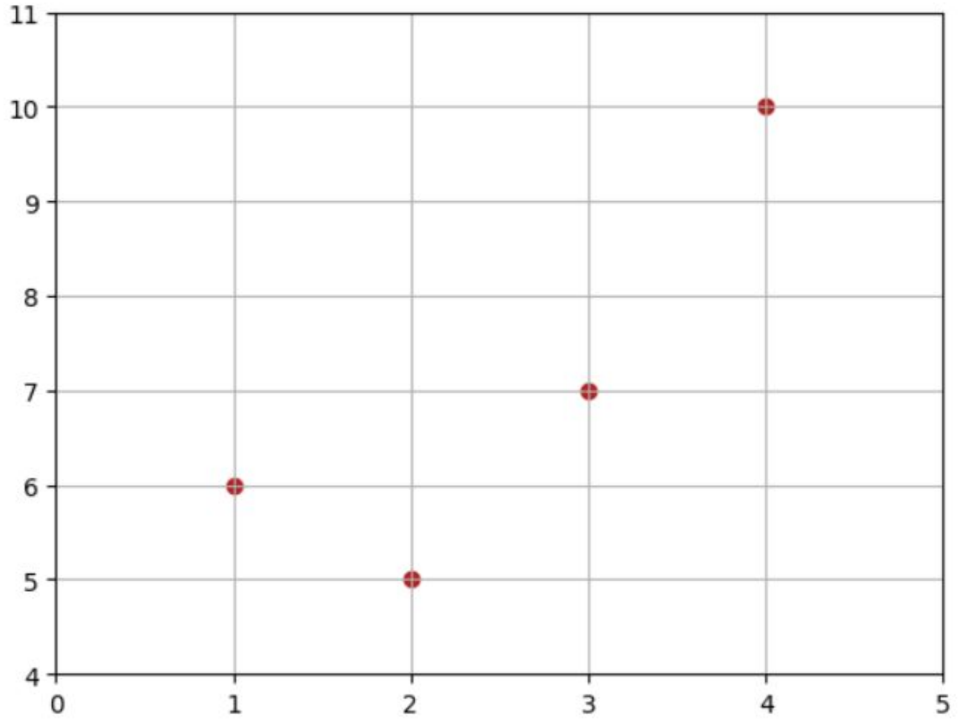


Now we can frame things into a system of equations

We want the value of these

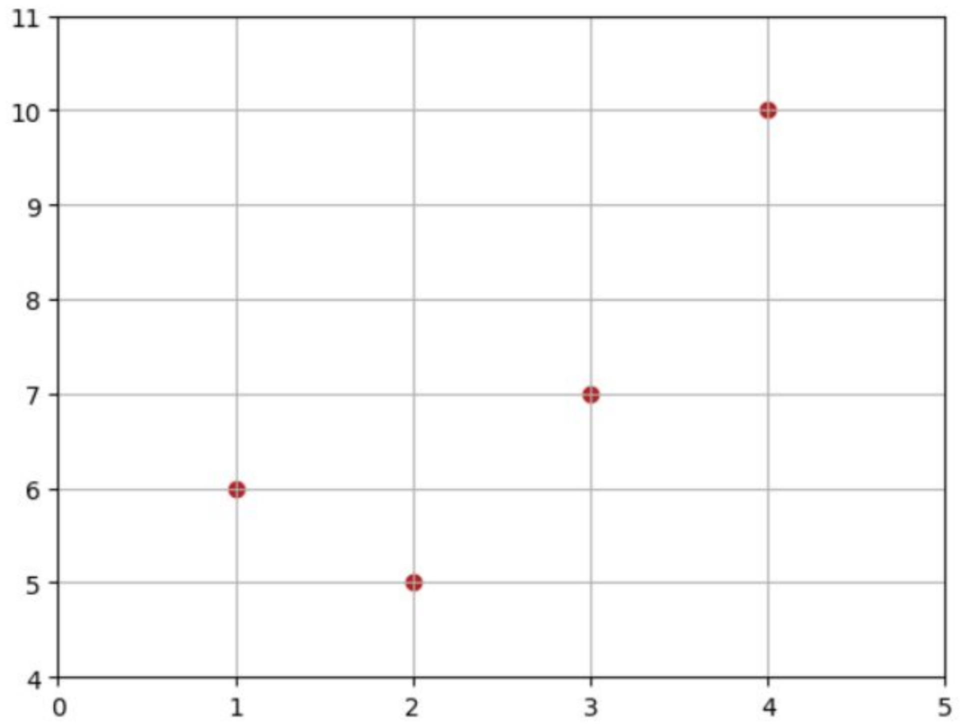
$$\begin{aligned}
\boxed{\beta_1} + 1\boxed{\beta_2} + \boxed{r_1} &= 6 \\
\beta_1 + 2\beta_2 + r_2 &= 5 \\
\beta_1 + 3\beta_2 + r_3 &= 7 \\
\beta_1 + 4\beta_2 + r_4 &= 10
\end{aligned}$$

And we want to minimize the “total” of these



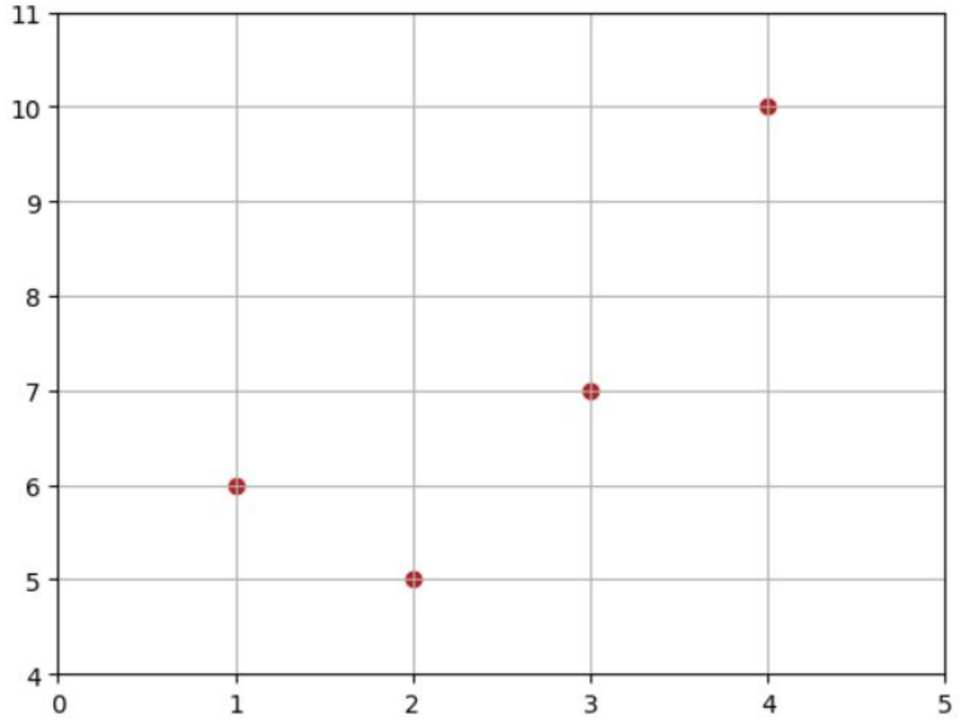
We rearrange to put the residual terms on the LHS

$$\begin{aligned}
r_1 &= \;\;6 - (\beta_1 + 1\beta_2) \\
r_2 &= \;\;5 - (\beta_1 + 2\beta_2) \\
r_3 &= \;\;7 - (\beta_1 + 3\beta_2) \\
r_4 &= 10 - (\beta_1 + 4\beta_2)
\end{aligned}$$



Then we reframe in terms of a sum of squares equation

$$
\begin{aligned}
S(\beta_1, \beta_2) &= r_1^2 + r_2^2 + r_3^2 + r_4^2 \\
&= [6 - (\beta_1 + 1\beta_2)]^2 \\
&\quad + [5 - (\beta_1 + 2\beta_2)]^2 \\
&\quad + [7 - (\beta_1 + 3\beta_2)]^2 \\
&\quad + [10 - (\beta_1 + 4\beta_2)]^2
\end{aligned}
$$



Then we reframe in terms of a sum of squares equation

$$S(\beta_1, \beta_2) = r_1^2 + r_2^2 + r_3^2 + r_4^2$$

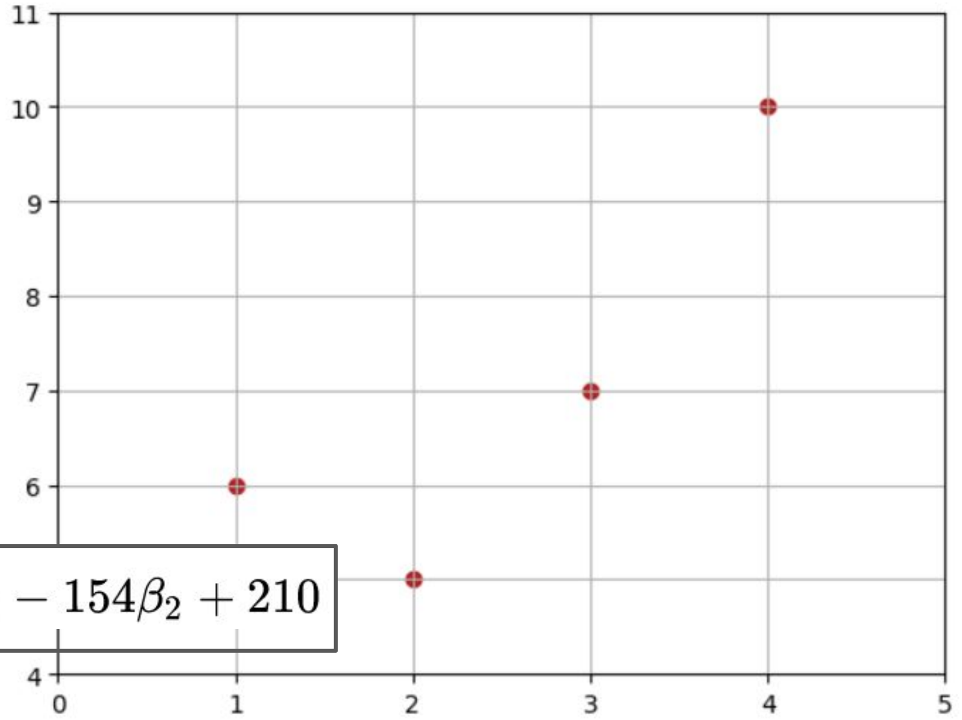
$$= [6 - (\beta_1 + 1\beta_2)]^2$$

$$+ [5 - (\beta_1 + 2\beta_2)]^2$$

$$+ [7 - (\beta_1 + 3\beta_2)]^2$$

$$+ [10 - (\beta_1 + 4\beta_2)]^2$$

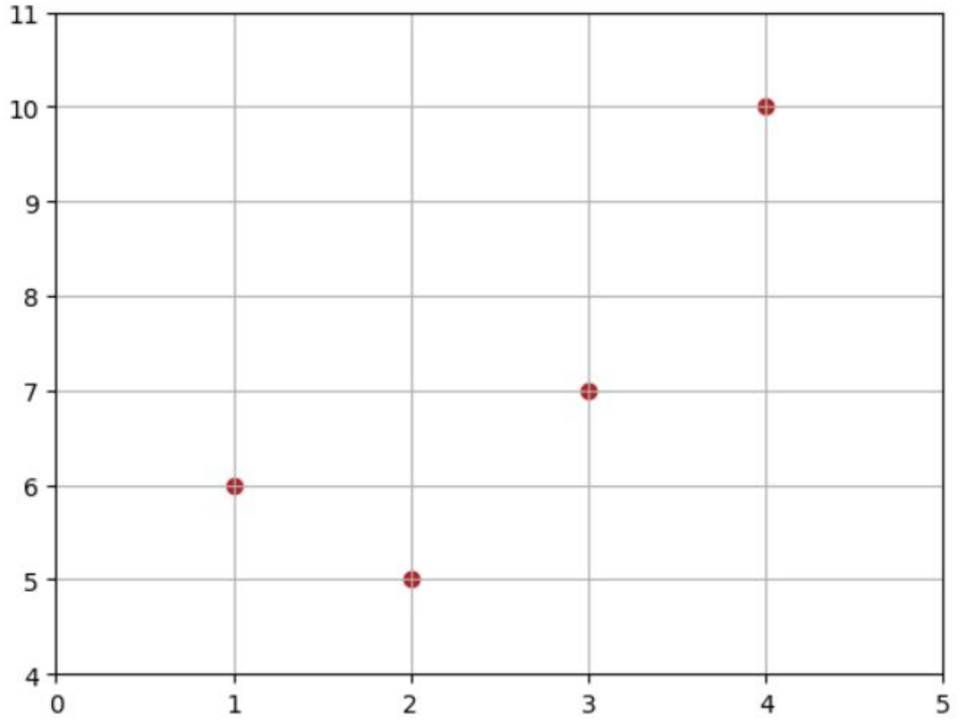
$$\boxed{= 4\beta_1^2 + 30\beta_2^2 + 20\beta_1\beta_2 - 56\beta_1 - 154\beta_2 + 210}$$



Set the partial derivatives equal to zero to find the “least squares error”

$$\frac{\partial S}{\partial \beta_1} = 0 = 8\beta_1 + 20\beta_2 - 56$$

$$\frac{\partial S}{\partial \beta_2} = 0 = 20\beta_1 + 60\beta_2 - 154.$$

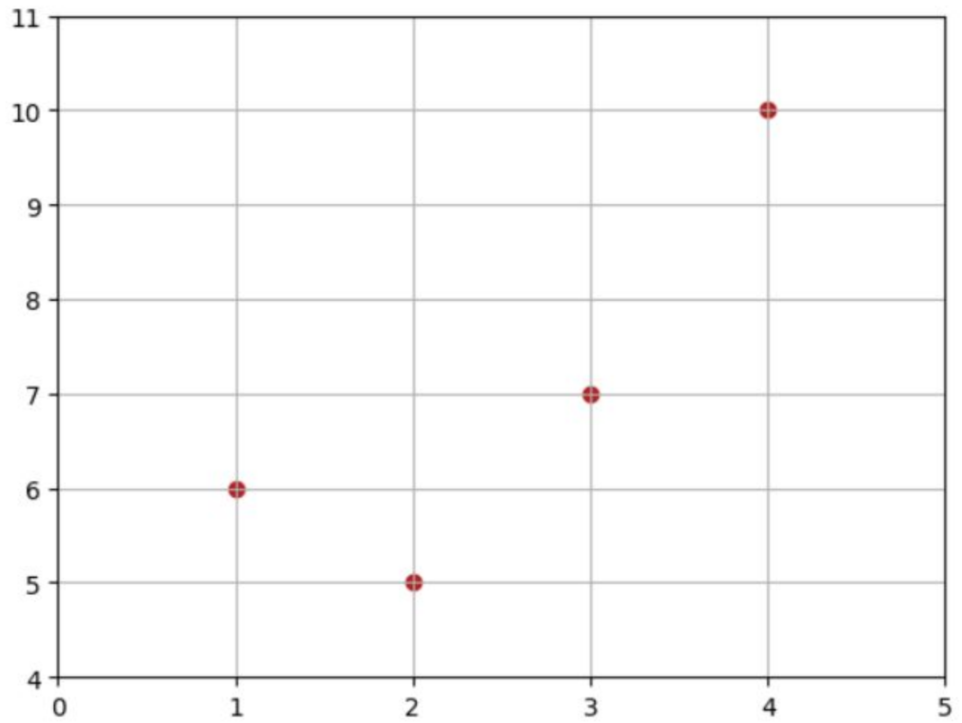


Set the partial derivatives equal to zero to find the “least squares error”

$$\frac{\partial S}{\partial \beta_1} = 0 = 8\beta_1 + 20\beta_2 - 56$$

$$\frac{\partial S}{\partial \beta_2} = 0 = 20\beta_1 + 60\beta_2 - 154.$$

2 equations, 2 unknowns, easy-peasy

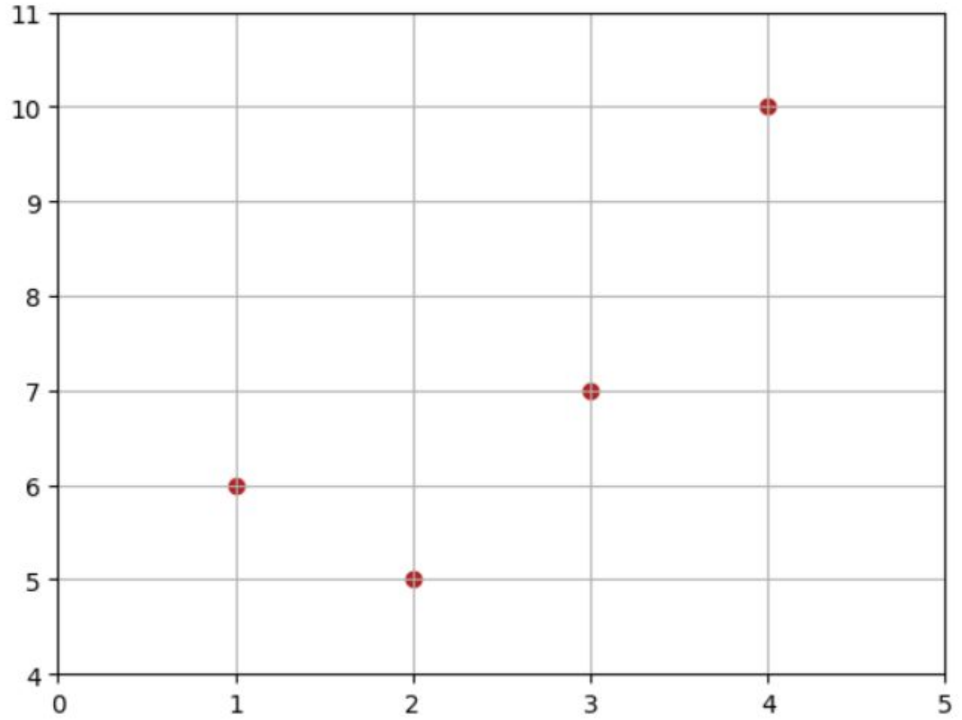


Set the partial derivatives equal to zero to find the "least squares error"

$$\beta_1 = 3.5$$

$$\beta_2 = 1.4$$

2 equations, 2 unknowns, easy-peasy

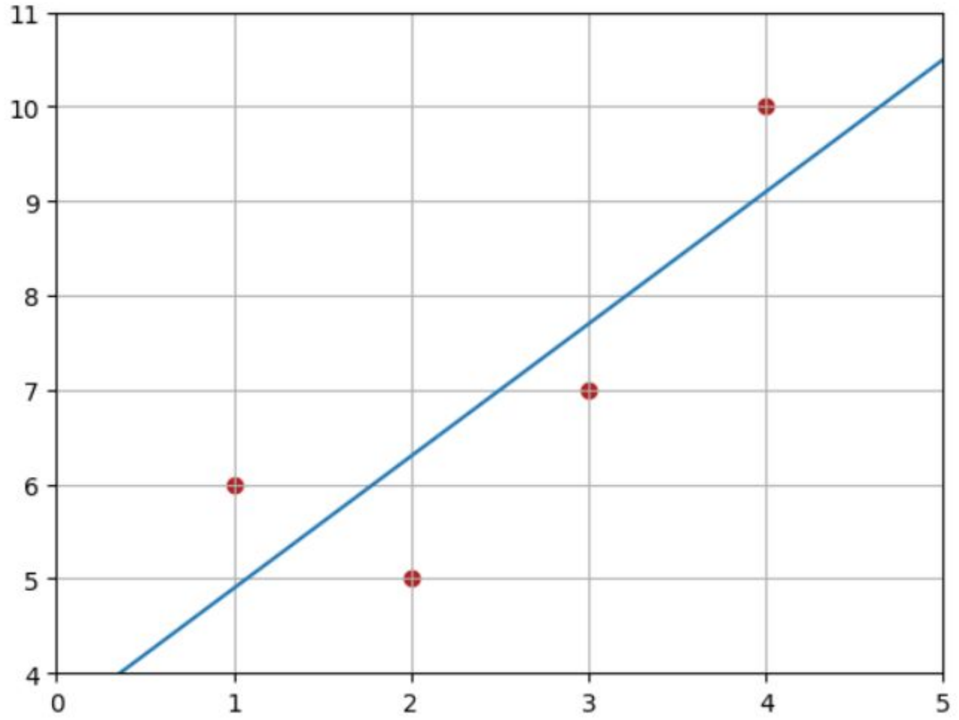


Set the partial derivatives equal to zero to find the “least squares error”

$$\beta_1 = 3.5$$

$$\beta_2 = 1.4$$

2 equations, 2 unknowns, easy-peasy

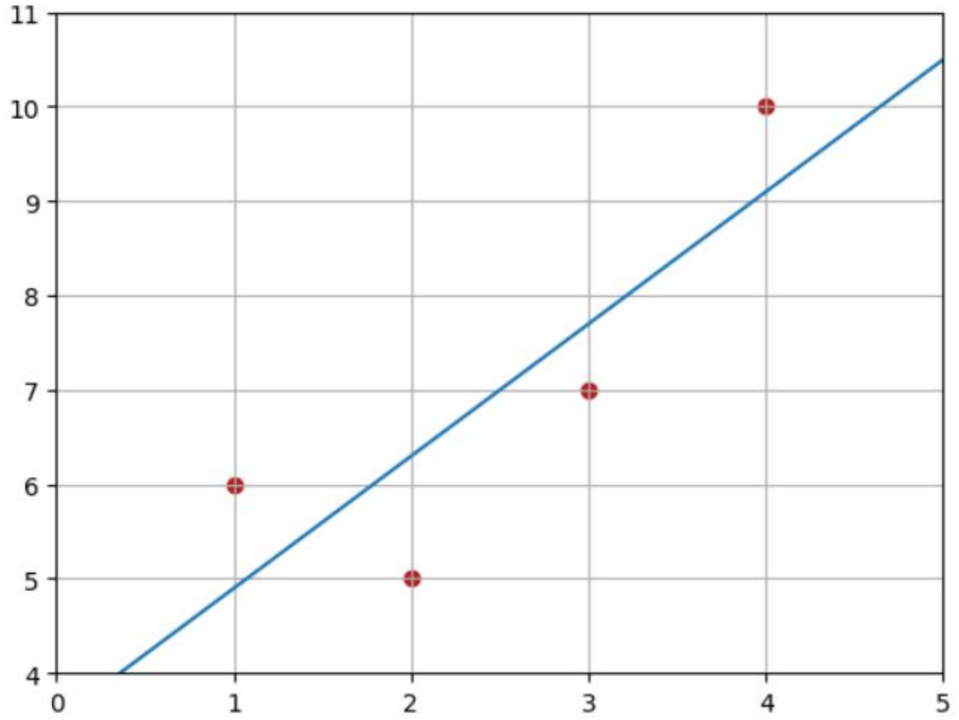


Looks pretty good! But I’m glossing over ***A TON***

The statistical procedure for fitting Ordinary Least Squares (OLS) models is generally understood

However there are many caveats, conditions, etc

And, particularly, linear least squares is not always the best model (actually almost never)

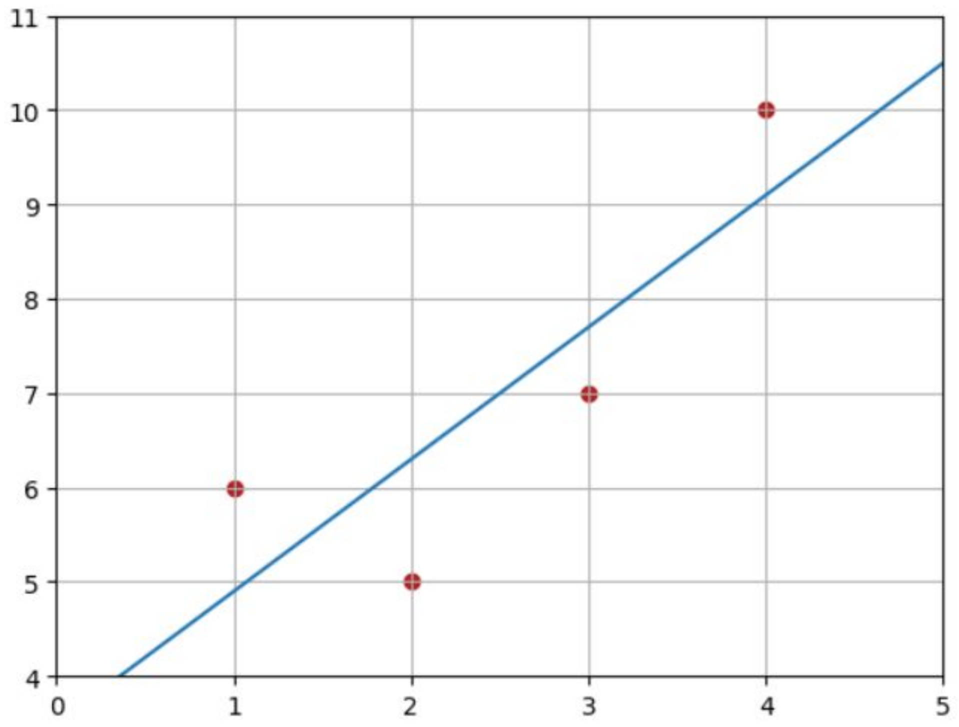


Looks pretty good! But I’m glossing over ***A TON***

The statistical procedure for fitting Ordinary Least Squares (OLS) models is generally understood

However there are many caveats, conditions, etc

And, particularly, linear least squares is not always the best model (actually almost never)



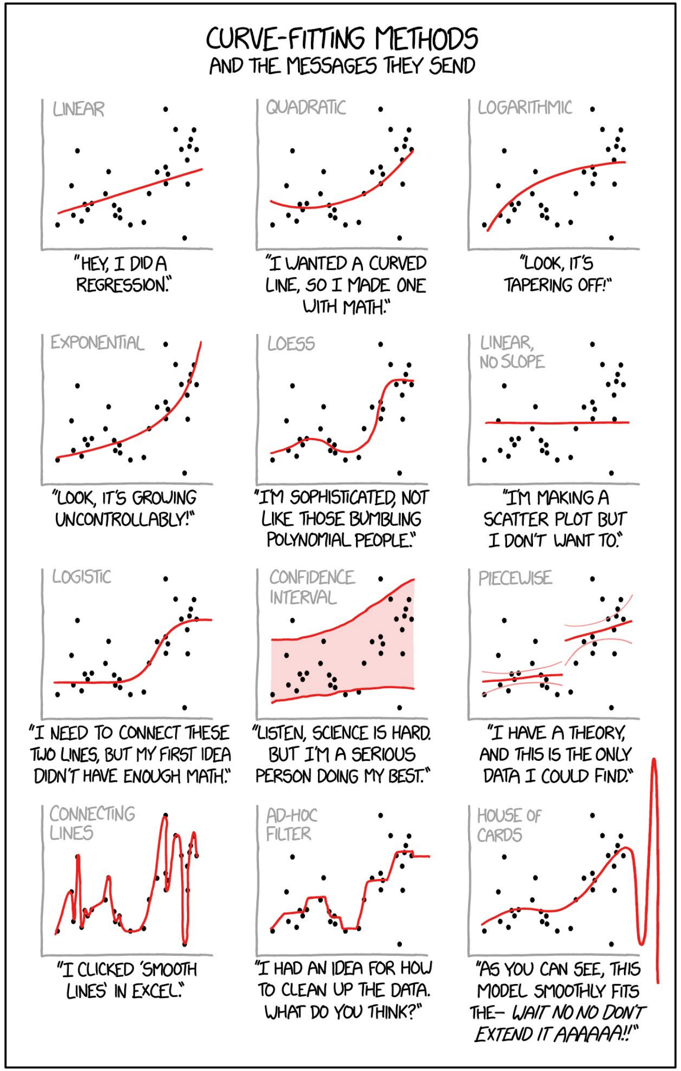
Looks pretty good! But I'm glossing over ***A TON***

The statistical procedure for fitting Ordinary Least Squares (OLS) models is generally understood

However there are many caveats, conditions, etc

And, particularly, linear least squares is not always the best model (actually almost never)

And then there's non-linear regression….


CURVE-FITTING METHODS
AND THE MESSAGES THEY SEND
LINEAR
"HEY, I DID A REGRESSION."
QUADRATIC
"I WANTED A CURVED LINE, SO I MADE ONE WITH MATH."
LOGARITHMIC
"LOOK, IT'S TAPERING OFF!"
EXPONENTIAL
"LOOK, IT'S GROWING UNCONTROLLABLY!"
LOESS
"I'M SOPHISTICATED, NOT LIKE THOSE BUMBLING POLYNOMIAL PEOPLE."
LINEAR, NO SLOPE
"I'M MAKING A SCATTER PLOT BUT I DON'T WANT TO."
LOGISTIC
"I NEED TO CONNECT THESE TWO LINES, BUT MY FIRST IDEA DIDN'T HAVE ENOUGH MATH."
CONFIDENCE INTERVAL
"LISTEN, SCIENCE IS HARD. BUT I'M A SERIOUS PERSON DOING MY BEST."
PIECEWISE
"I HAVE A THEORY, AND THIS IS THE ONLY DATA I COULD FIND."
CONNECTING LINES
"I CLICKED 'SMOOTH LINES' IN EXCEL."
AD-HOC FILTER
"I HAD AN IDEA FOR HOW TO CLEAN UP THE DATA. WHAT DO YOU THINK?"
HOUSE OF CARDS
"AS YOU CAN SEE, THIS MODEL SMOOTHLY FITS THE- WAIT NO NO DON'T EXTEND IT AAAAAA!!"

HOW DO WE DO THIS WITH PYTHON!?

The big 3

scikit learn

statsmodels

SciPy

The big 3

scikit learn

statsmodels

SciPy

Why scikit-learn?

It is well established, same for statsmodels & scipy

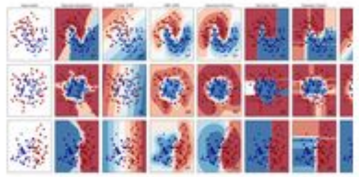
It has a consistent interface (or API) between model types

It has a number of other applications I hope you find useful scikit learn

Classification

Identifying which category an object belongs to.

Applications: Spam detection, Image recognition.
Algorithms: SVM, nearest neighbors, random forest, and more...

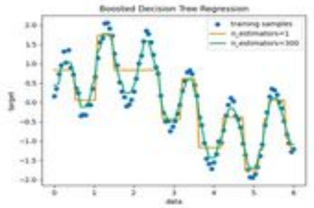


Examples

Regression

Predicting a continuous-valued attribute associated with an object.

Applications: Drug response, Stock prices.
Algorithms: SVR, nearest neighbors, random forest, and more...

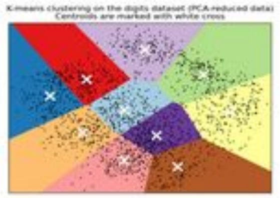


Examples

Clustering

Automatic grouping of similar objects into sets.

Applications: Customer segmentation, Grouping experiment outcomes
Algorithms: k-Means, spectral clustering, mean-shift, and more...

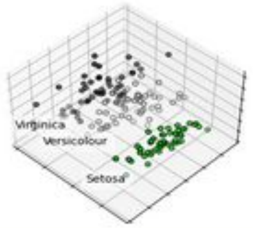


Examples

Dimensionality reduction

Reducing the number of random variables to consider.

Applications: Visualization, Increased efficiency
Algorithms: PCA, feature selection, non-negative matrix factorization, and more...

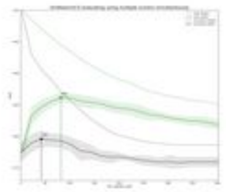


Examples

Model selection

Comparing, validating and choosing parameters and models.

Applications: Improved accuracy via parameter tuning
Algorithms: grid search, cross validation, metrics, and more...

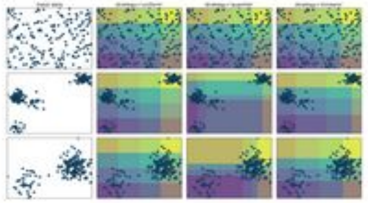


Examples

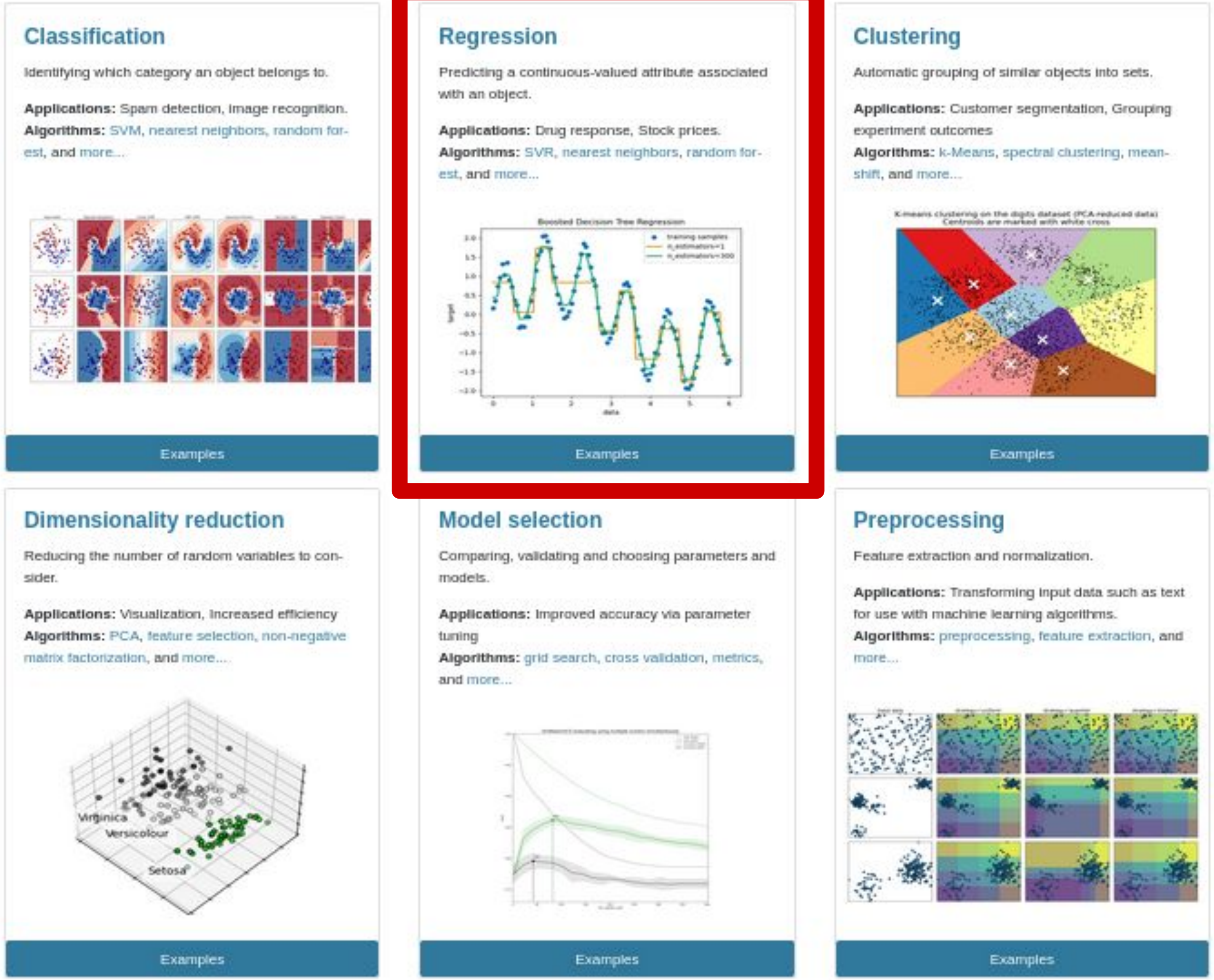
Preprocessing

Feature extraction and normalization.

Applications: Transforming input data such as text for use with machine learning algorithms.
Algorithms: preprocessing, feature extraction, and more...



Examples


Boosted Decision Tree Regression
training samples
n_estimators=1
n_estimators=300
data
target
K-means clustering on the digits dataset (PCA-reduced data)
Centroids are marked with white cross
Virginica
Versicolour
Setosa

Classification

Identifying which category an object belongs to.

Applications: Spam detection, Image recognition.
Algorithms: SVM, nearest neighbors, random forest, and more...

Examples

Regression

Predicting a continuous-valued attribute associated with an object.

Applications: Drug response, Stock prices.
Algorithms: SVR, nearest neighbors, random forest, and more...

Examples

Clustering

Automatic grouping of similar objects into sets.

Applications: Customer segmentation, Grouping experiment outcomes
Algorithms: k-Means, spectral clustering, mean-shift, and more...

Examples

Dimensionality reduction

Reducing the number of random variables to consider.

Applications: Visualization, Increased efficiency
Algorithms: PCA, feature selection, non-negative matrix factorization, and more...

Examples

Model selection

Comparing, validating and choosing parameters and models.

Applications: Improved accuracy via parameter tuning
Algorithms: grid search, cross validation, metrics, and more...

Examples

Preprocessing

Feature extraction and normalization.

Applications: Transforming input data such as text for use with machine learning algorithms.
Algorithms: preprocessing, feature extraction, and more...

Examples

https://scikit-learn.org/stable/auto_examples/linear_model/plot_ols.html