





# Gentle Introduction to Machine Learning: Regression

## A Reproducible Research Workshop

Simon Stone

*Research Data Services*

*Dartmouth College*





# About the Reproducible Research Group

- Joint venture of **Research Computing @ ITC** and **Research Data Services @ Library**
- Consult with **experts** on
  - research data management,
  - data visualization,
  - biomedical research support,
  - spatial data and GIS,
  - high performance and research computing,
  - statistical analysis,
  - economics and social sciences data
- **Meet** the people on campus that support your reproducible research lifecycle
- **Engage** in community discussions to learn from other researchers on campus
- Attend a workshop to **learn** practical tools and tips



# About Research Data Services

## Research Data Management

Data Management Plans (DMPs) for sponsored projects

Finding and using 3rd party data

Collection and cleaning of data

Organization and documentation

Publishing and Repositories

## Data Analysis/Visualization

Textual, numeric, spatial data

Reproducible research workflows

Scripting in R: tidyverse core package (i.e., ggplot, dplyr, tydr, tibble, etc.)

Scripting in Python: NumPy, SciPy, Pandas, Scikit-learn, Matplotlib, Seaborn, (OpenCV, PyTorch, TensorFlow, Tesseract, NLTK, etc.)

## Computational Scholarship

Computational project planning

Collections as Data

Storytelling with data and visualizations

Text and data mining

Digital Humanities support

Computational Pedagogy



# Work with us

[ResearchDataHelp@groups.dartmouth.edu](mailto:ResearchDataHelp@groups.dartmouth.edu)

**Jeremy Mikecz**

Research Data Science Specialist  
[jeremy.m.mikecz@dartmouth.edu](mailto:jeremy.m.mikecz@dartmouth.edu)  
[dartgo.org/jeremyappts](https://dartgo.org/jeremyappts)

**Simon Stone**

Research Data Science Specialist  
[simon.stone@dartmouth.edu](mailto:simon.stone@dartmouth.edu)  
[dartgo.org/meetwithsimon](https://dartgo.org/meetwithsimon)

**Lora Leligdon**

Head of Research Data Services  
[lora.c.leligdon@dartmouth.edu](mailto:lora.c.leligdon@dartmouth.edu)  
[dartgo.org/lora](https://dartgo.org/lora)

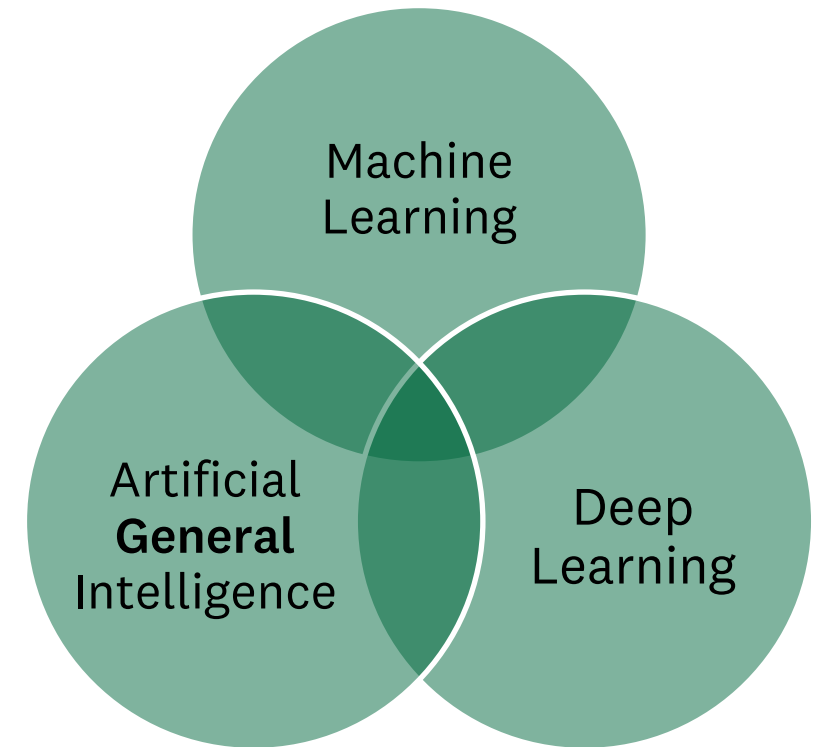
# Gentle Introduction to Machine Learning

## Intro

Machine Learning is "the field of study that gives computers the ability to learn without explicitly being programmed."

- Arthur Samuel, 1959 (paraphrased)

Machine Learning is what we should call (the current) Artificial Intelligence!





# Gentle Introduction to Machine Learning

## Intro

### Aims of this series:

- 🔮 **Demystify** the field a bit and give context to the buzzwords
- 💡 A working **mental model** how machine learning algorithms ~~think~~ calculate
- 🤔 To provide enough knowledge to **think critically** about “A.I.”
- 😎 To inspire you to **confidently use machine learning** in your work and personal life



# Gentle Introduction to Machine Learning

## Intro

- **Statistics** (April 12)

-  A brief survey of the fundamentals for Machine Learning

- **Regression** (April 25)

-  How can an algorithm find relationships between two variables?

- **Classification** (May 9)

-  How can an algorithm put a label on a real-world object?





# Gentle Introduction to Machine Learning

## Intro

### Feedback from Intro to Statistics:



Generally very positive



Most frequent request: More "how", more math!



# Basics

## Intro

Every machine learning model, ever:

Trained model:





# Basics

## Intro

### Training process:

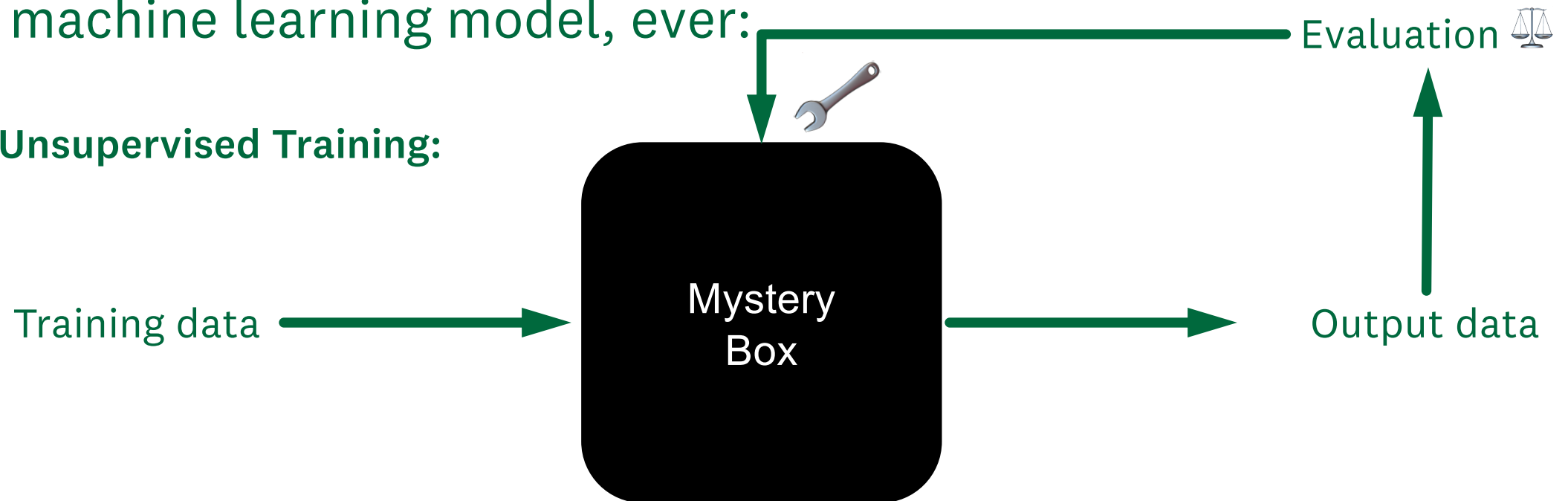
- Choose a model structure for your problem
  - Every model has parameters that can be changed to better fit the problem at hand
- Show the model plenty of data
- Adapt the model's parameters to best fit the data
- Challenge: How do we know what is the “best fit”?

# Basics

## Intro

Every machine learning model, ever:

During Unsupervised Training:

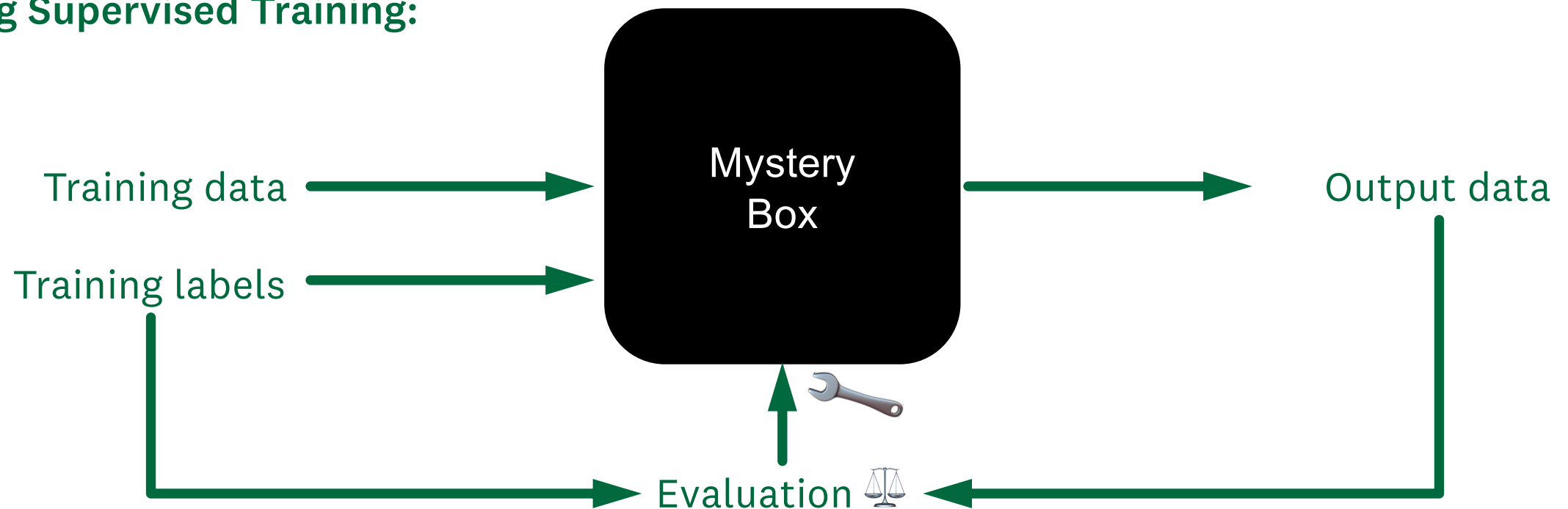


# Basics

## Intro

Every machine learning model, ever:

During Supervised Training:





# Gentle Introduction to Machine Learning

## Intro

Supervised  
Learning

Classification

Regression

Unsupervised  
Learning

Clustering

Anomaly  
Detection

Semi-supervised  
Learning








Mix of  
labeled and  
unlabeled  
data

Self-supervised  
Learning

Language  
modeling

# Gentle Introduction to Machine Learning

## Outline

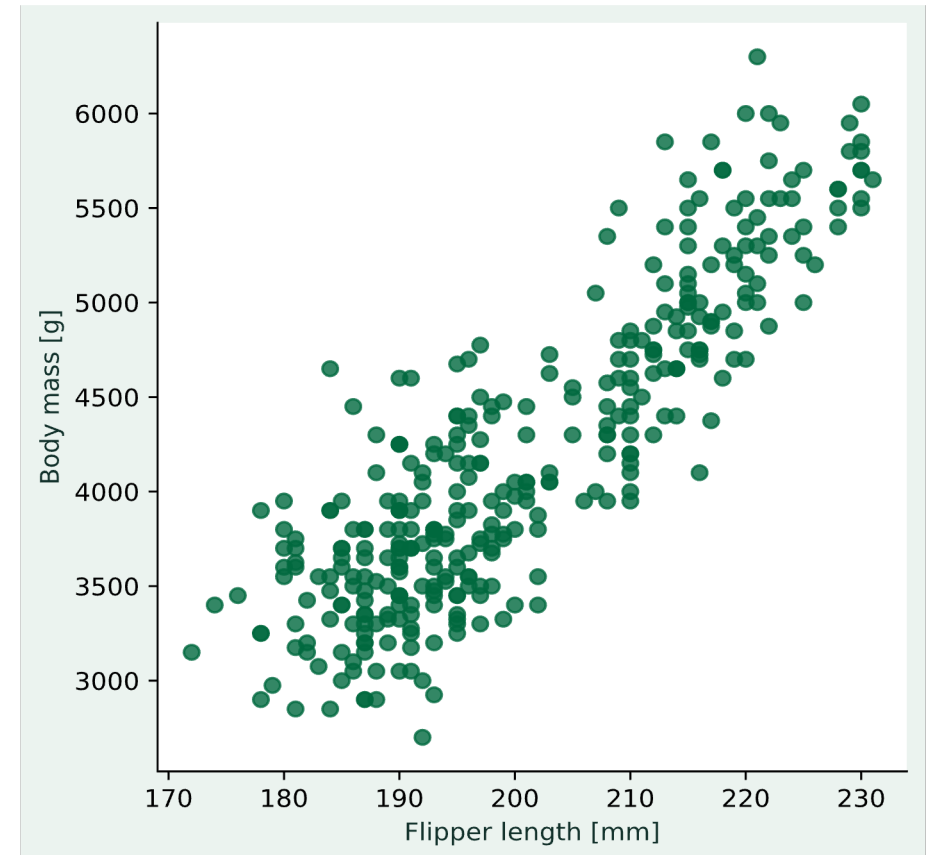
-  What is “regression”?
-  Best-fit regression models: Linear Regression
-  The many, many, MANY kinds of regression models
-  An artificial neural network for regression problems
-  Case study: Diamond price prediction
-  Critical Thinking
-  Summary

## Basics

# What is “regression”?

Consider the following problem:

- 🐧 We have measurements of the flipper length and the body mass of penguins
- 🐧 We suspect that there is some kind of relationship between the two variables
- 🐧 Formulating a causal, biologically-motivated model for this relationship seems difficult
- 🐧 Instead, we try to somehow learn the “rules” for the association from the data itself (supervised learning)







## Basics

# Regression problem or not?

### Definition:

We use regression models to associate **one or more independent variables** with a **continuous-valued dependent variable** without knowing the rules for this association.

### Activity:

Which of these problems could we tackle with a regression model?

- Predicting if a customer will buy a product based on their previous purchase history.
- Predicting a student's college GPA based on their high school GPA and SAT score.
- Predicting the distance traveled by a spaceship based on time.
- Predict the load of a server based on time.
- Predicting a company's revenue based on its advertising expenditure.
- Predict the value of a house based on its features.

## Best-fit models

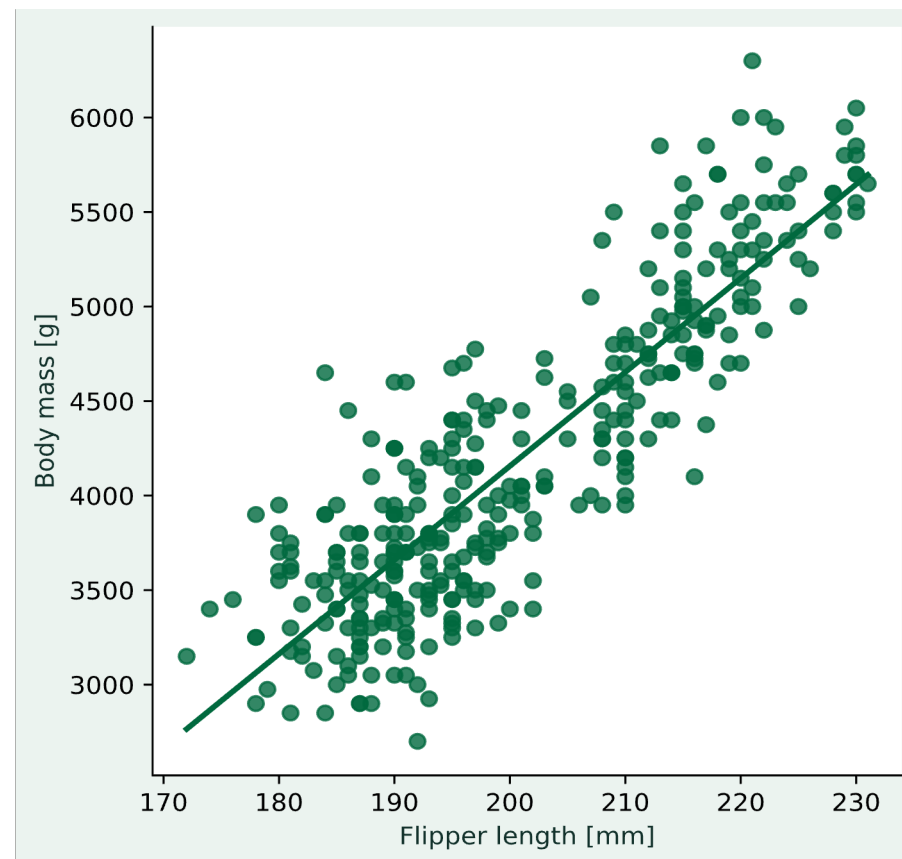
# Linear Regression – The Art of Drawing a Line

Consider the following problem:

- 🐧 We have measurements of the flipper length and the body mass of penguins
- 🐧 We suspect that there is some kind of relationship between the two variables
- 🐧 Formulating a causal, biologically-motivated model for this relationship seems difficult
- 🐧 Intuitively, we would suspect some kind of linear relationship plus some noise:

$$\text{mass}_i = \beta_0 + \beta_1 \cdot \text{length}_i + \varepsilon_i$$

How to draw the best line?



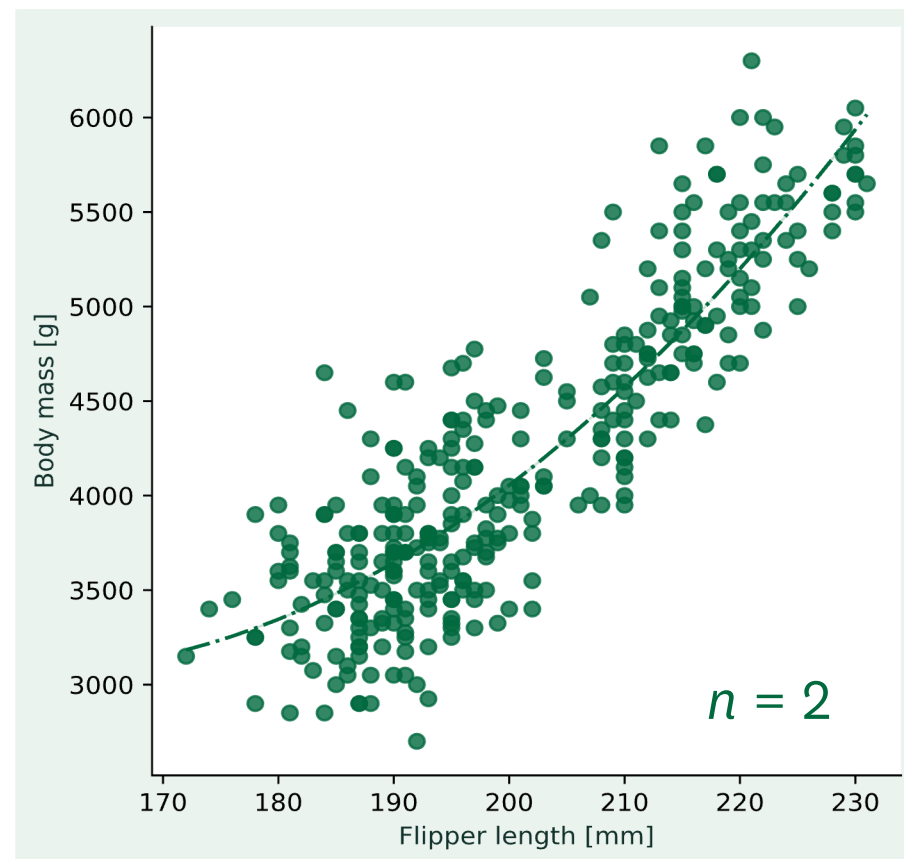
## Best-fit models

# Linear Regression – The Art of Drawing a Line

We could also assume more complex relationships between the independent  $x$  (predictor) and the dependent  $y$  (response, target):

- Quadratic:

$$y_i = \beta_0 + \beta_1 \cdot x_i + \beta_2 \cdot x_i^2 + \varepsilon_i$$



## Best-fit models

# Linear Regression – The Art of Drawing a Line

We could also assume more complex relationships between the independent  $x$  (predictor) and the dependent  $y$  (response, target):

- Quadratic:

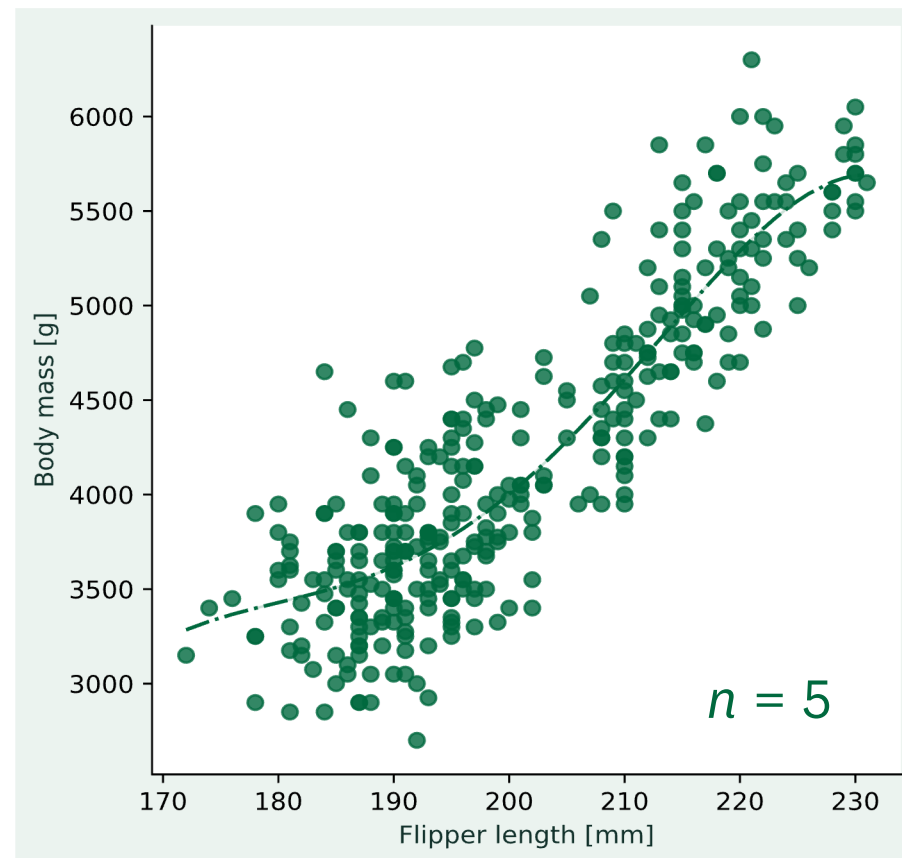
$$y_i = \beta_0 + \beta_1 \cdot x_i + \beta_2 \cdot x_i^2 + \varepsilon_i$$

- Polynomial of degree  $n$ :

$$y_i = \beta_0 + \beta_1 \cdot x_i + \beta_2 \cdot x_i^2 + \dots + \beta_n \cdot x_i^n + \varepsilon_i$$

- Any **linear** combination of some basis function, really:

$$y_i = \beta_0 + \sum_{k=1}^n \beta_k \cdot \phi_k(x_i) + \varepsilon_i$$



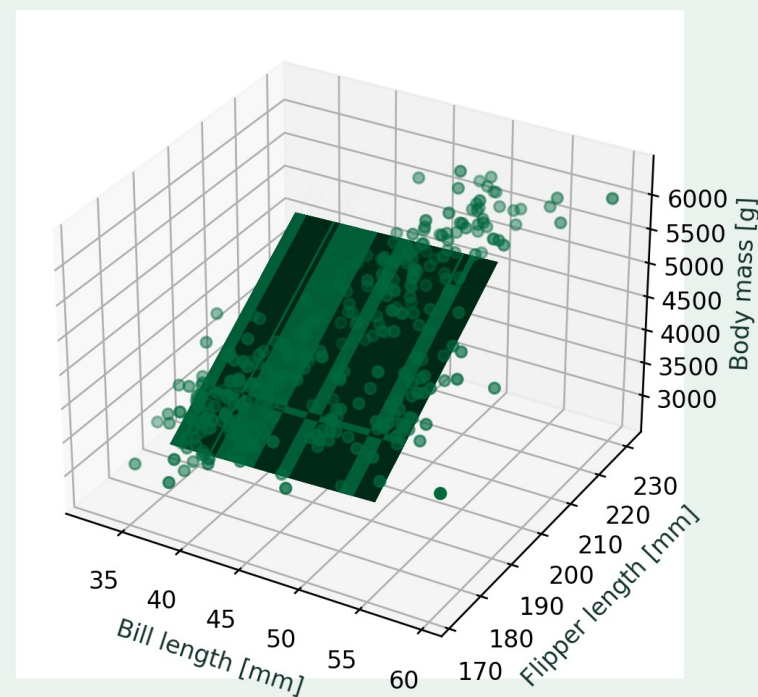
## Best-fit models

# Linear Regression – The Art of Drawing a Line

We could also use multiple independent variables with any basis function (here, degree 1):

$$y_i = \beta_0 + \beta_1 \cdot x_{1,i} + \beta_2 \cdot x_{2,i} + \varepsilon_i$$

- “Multiple linear regression”
- Theoretically arbitrarily many predictors possible
- Usually just a single response per model

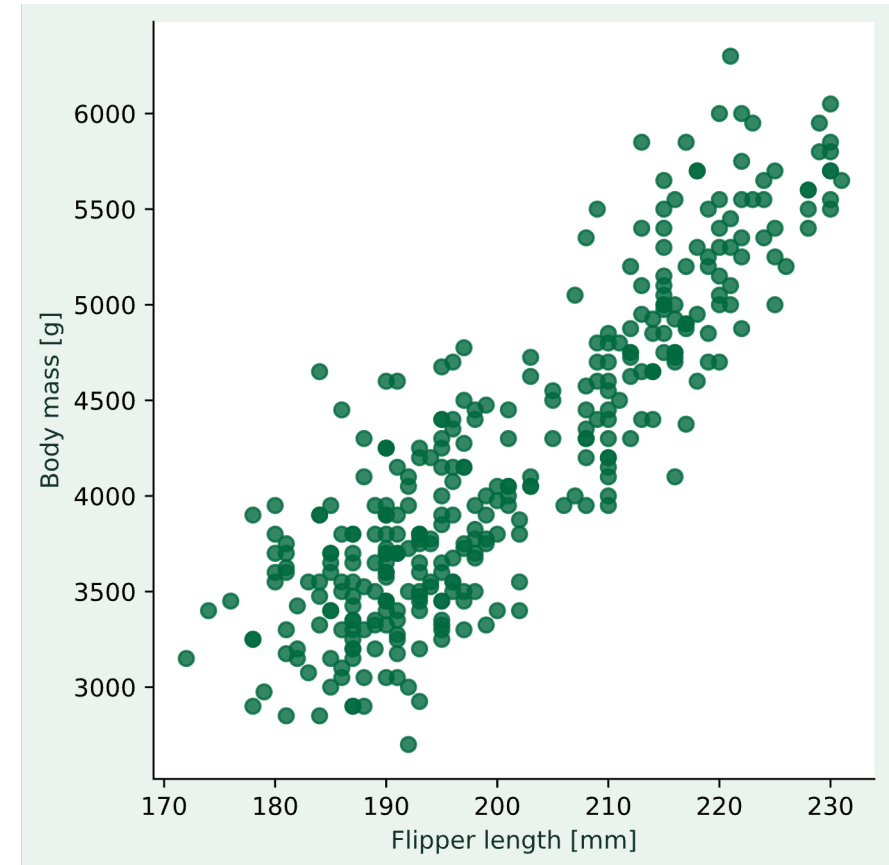


## Best-fit models

# Linear Regression – Survival of the best fit

### Activity:

- Where would you draw the line?
- How could you be sure that it is “the best line”?



## Best-fit models

# Linear Regression – Survival of the best fit

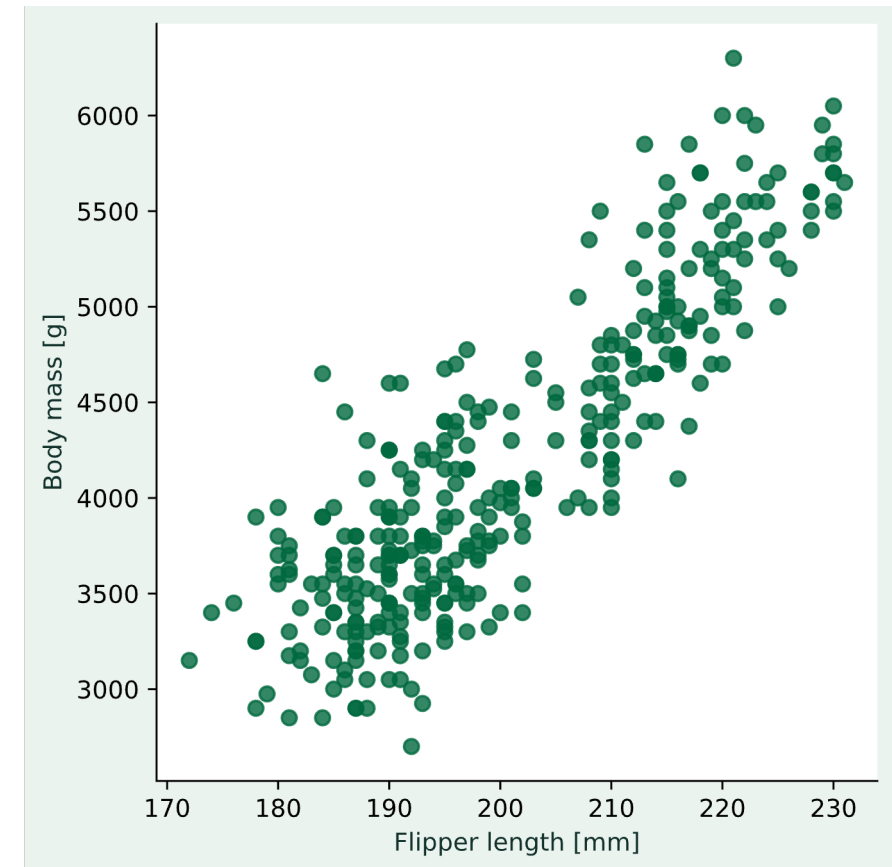
### Assumption:

$$y_i = \beta_0 + \beta_1 \cdot x_i + \varepsilon_i$$

*Drawing a line* means finding the values for the parameters  $\beta_i$ .

We accept that we cannot model the  $\varepsilon_i$  (random error).

The *best fit* would be the one that makes the smallest errors.



## Best-fit models

# Linear Regression – Survival of the best fit

The true observations are:

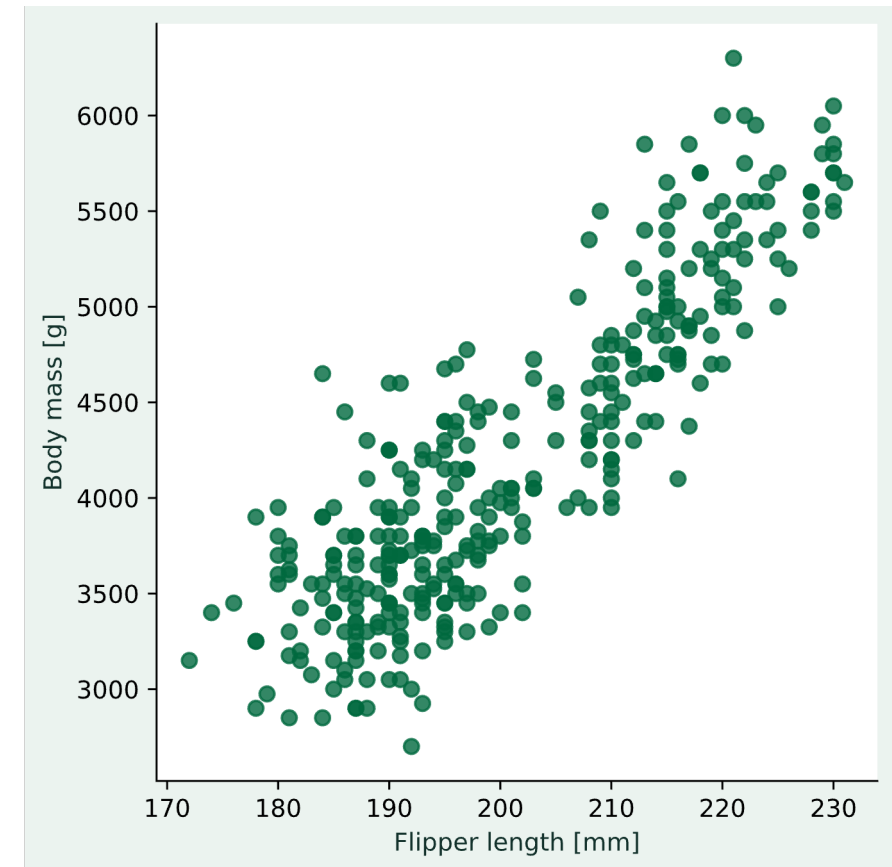
$$y_i = \beta_0 + \beta_1 \cdot x_i + \varepsilon_i$$

Our slightly erroneous predictions are:

$$\hat{y}_i = \beta_0 + \beta_1 \cdot x_i$$

The error is therefore:

$$|\hat{y}_i - y_i| = \sqrt{(\hat{y}_i - y_i)^2}$$



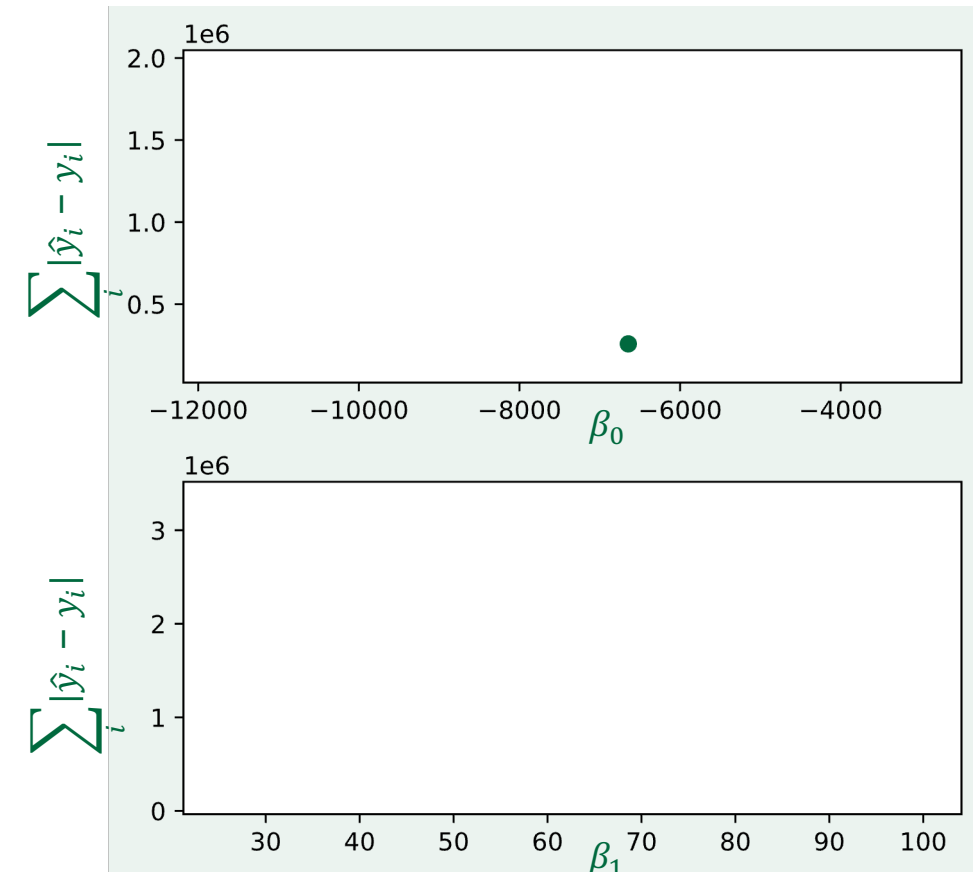




## Best-fit models

# Linear Regression – Survival of the best fit

1. Pick some values for  $\beta_0$  and  $\beta_1$
2. Calculate the error and plot it
3. Pick some other values and repeat

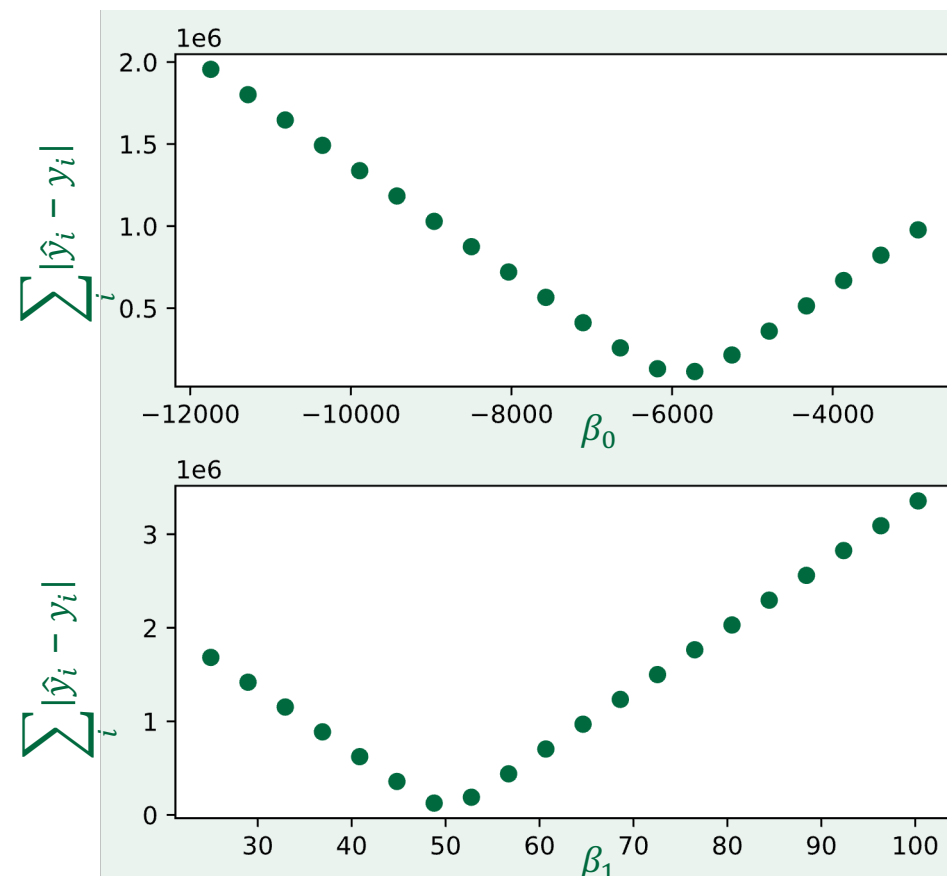


## Best-fit models

# Linear Regression – Survival of the best fit

1. Pick some values for  $\beta_0$  and  $\beta_1$
2. Calculate the error and plot it
3. Pick some other values and repeat
4. We can see a minimum, so there is a best fit!

Can we find the optimum without trial & error?



# Best-fit models

## Linear Regression – Survival of the best fit

Find the minimum using analysis:

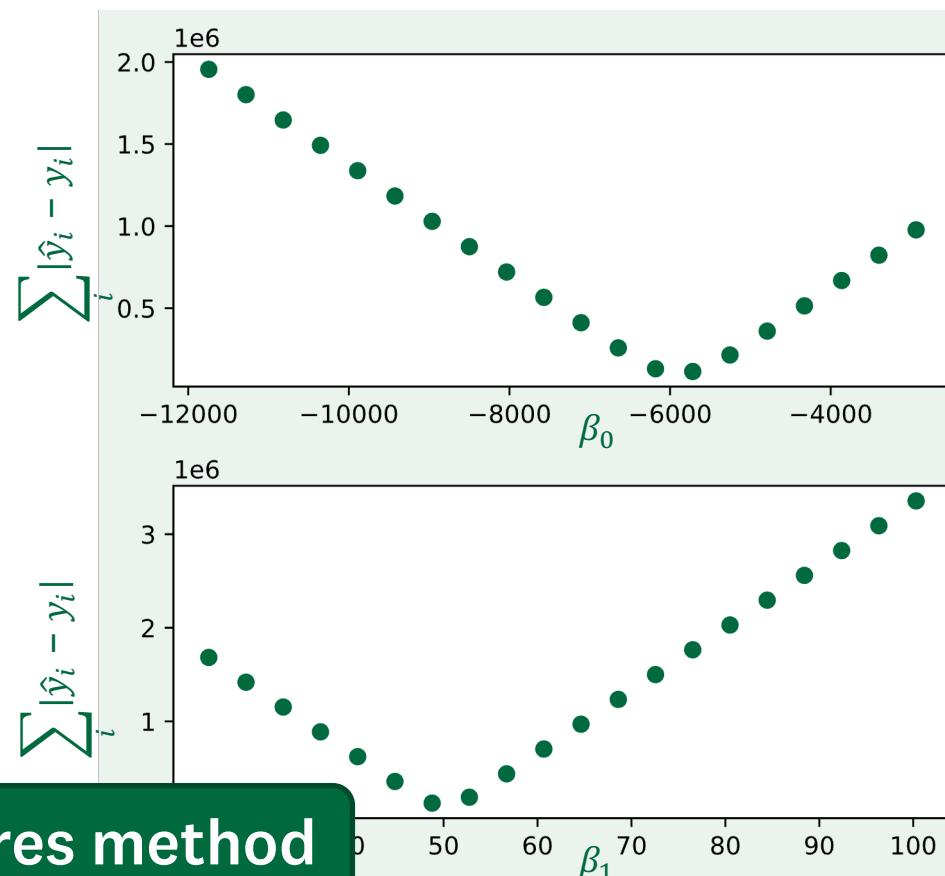
1. Define an error function (a.k.a. *loss*):

$$e(\beta_0, \beta_1, y_i) = \sum_i (\hat{y}_i - y_i)^2 = \sum_i (\beta_0 + \beta_1 \cdot x_i - y_i)^2$$

2. The derivative of this function must be 0 at the minimum:

$$\frac{\partial e(\beta_0, \beta_1)}{\partial \beta_0 \partial \beta_1} = \frac{\partial}{\partial \beta_0 \partial \beta_1} \sum_i (\beta_0 + \beta_1 \cdot x_i - y_i)^2 \equiv 0$$

3. Solve this equation w.r.t.  $\beta_0$  and  $\beta_1$



**Least squares method**



## Best-fit models

## Linear Regression – Survival of the best fit

$$\frac{\partial}{\partial \beta_0} \sum_{i=0}^N (\beta_0 + \beta_1 \cdot x_i - y_i)^2 = 2 \cdot \sum_{i=0}^N (\beta_0 + \beta_1 \cdot x_i - y_i) = 0$$

$$0 = \sum_{i=0}^N (\beta_0 + \beta_1 \cdot x_i - y_i) = \sum_{i=0}^N \beta_0 + \sum_{i=0}^N \beta_1 \cdot x_i - \sum_{i=0}^N y_i$$

$$\sum_{i=0}^N y_i - \sum_{i=0}^N \beta_1 \cdot x_i = \sum_{i=0}^N \beta_0$$

$$\sum_{i=0}^N y_i - \beta_1 \cdot \sum_{i=0}^N x_i = N\beta_0$$

$$\frac{\sum_{i=0}^N y_i}{N} - \beta_1 \frac{\sum_{i=0}^N x_i}{N} = \beta_0$$

$$\beta_0 = \bar{y} - \beta_1 \bar{x}$$

$$\frac{\partial}{\partial \beta_1} \sum_{i=0}^N (\beta_0 + \beta_1 \cdot x_i - y_i)^2 = 2 \cdot \sum_{i=0}^N x_i \cdot (\beta_0 + \beta_1 \cdot x_i - y_i) = 0$$

$$0 = \sum_{i=0}^N (\beta_0 x_i + \beta_1 \cdot x_i^2 - y_i x_i) = \sum_{i=0}^N \beta_0 x_i + \sum_{i=0}^N \beta_1 \cdot x_i^2 - \sum_{i=0}^N y_i x_i$$

$$0 = \beta_0 \cdot \sum_{i=0}^N x_i + \beta_1 \cdot \sum_{i=0}^N x_i^2 - \sum_{i=0}^N y_i x_i$$

$$\sum_{i=0}^N y_i x_i - \beta_0 \cdot \sum_{i=0}^N x_i = \beta_1 \cdot \sum_{i=0}^N x_i^2$$

$$\sum_{i=0}^N y_i x_i - \left( \frac{\sum_{i=0}^N y_i}{N} - \beta_1 \frac{\sum_{i=0}^N x_i}{N} \right) \cdot \sum_{i=0}^N x_i = \beta_1 \cdot \sum_{i=0}^N x_i^2$$

$$\sum_{i=0}^N y_i x_i - \frac{\sum_{i=0}^N y_i \sum_{i=0}^N x_i}{N} + \beta_1 \frac{\left( \sum_{i=0}^N x_i \right)^2}{N} = \beta_1 \cdot \sum_{i=0}^N x_i^2$$

$$\sum_{i=0}^N y_i x_i - \frac{\sum_{i=0}^N y_i \sum_{i=0}^N x_i}{N} = \beta_1 \cdot \sum_{i=0}^N x_i^2 - \beta_1 \frac{\left( \sum_{i=0}^N x_i \right)^2}{N}$$

$$\sum_{i=0}^N y_i x_i - \frac{\sum_{i=0}^N y_i \sum_{i=0}^N x_i}{N} = \beta_1 \cdot \left( \sum_{i=0}^N x_i^2 - \frac{\left( \sum_{i=0}^N x_i \right)^2}{N} \right)$$

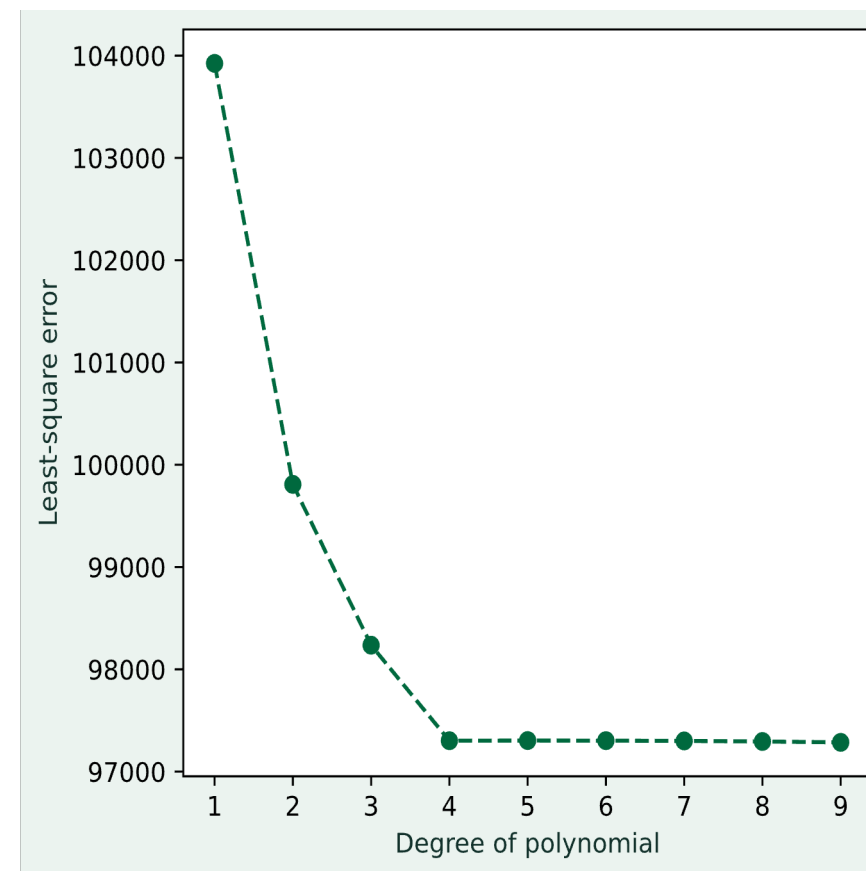
$$\frac{\sum_{i=0}^N y_i x_i - \frac{\sum_{i=0}^N y_i \sum_{i=0}^N x_i}{N}}{\sum_{i=0}^N x_i^2 - \frac{\left( \sum_{i=0}^N x_i \right)^2}{N}} = \beta_1$$

## Best-fit models

# Linear Regression – Survival of the best fit

### How to find the optimal basis function?

1. Try out all candidate basis functions
2. Calculate the least-squares error for each
3. Pick the candidate with the smallest error

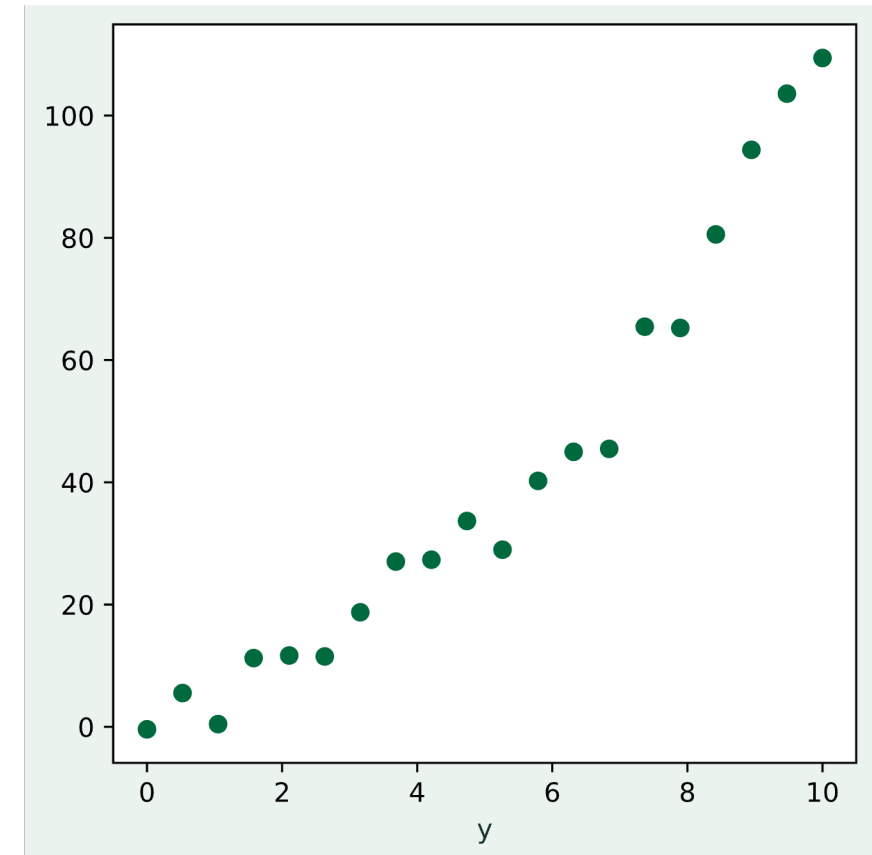




## Best-fit models

# Underfitting and overfitting

If we choose a basis function that does not have enough flexibility (i.e., parameters), it may not be able to match the shape of the data distribution (*underfitting*).

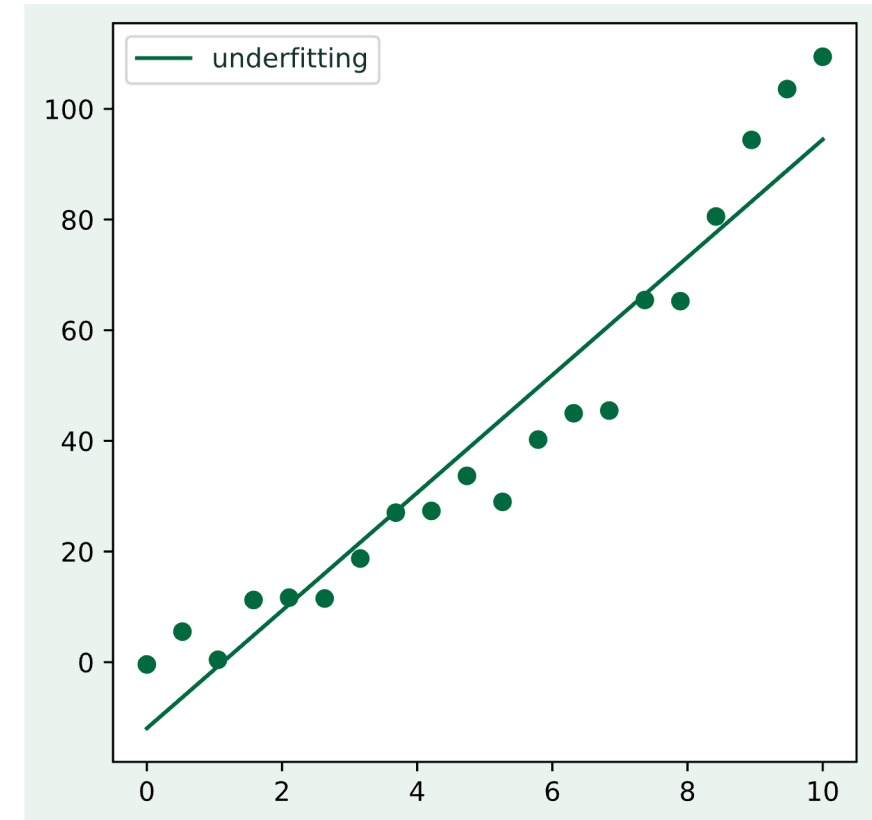




## Best-fit models

# Underfitting and overfitting

If we choose a basis function that does not have enough flexibility (i.e., parameters), it may not be able to match the shape of the data distribution (*underfitting*)



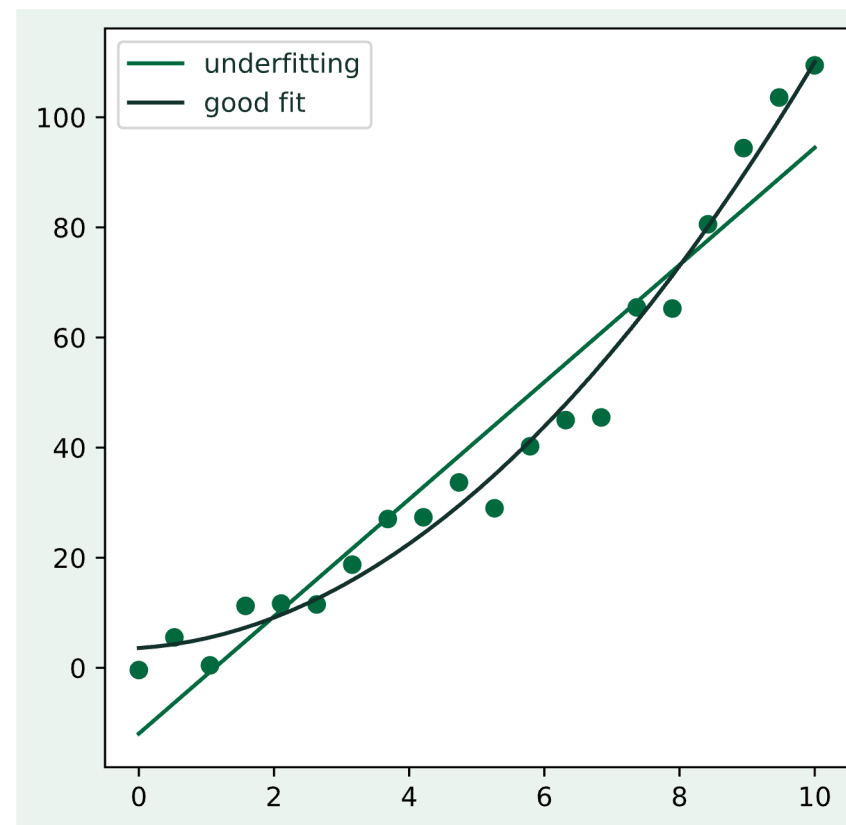


## Best-fit models

# Underfitting and overfitting

If we choose a basis function that does not have enough flexibility (i.e., parameters), it may not be able to match the shape of the data distribution (*underfitting*)

A good-fitting basis function should be smoothly going passing closely through or nearby the observations, not relying to strongly on individual points.





# Best-fit models

## Underfitting and overfitting

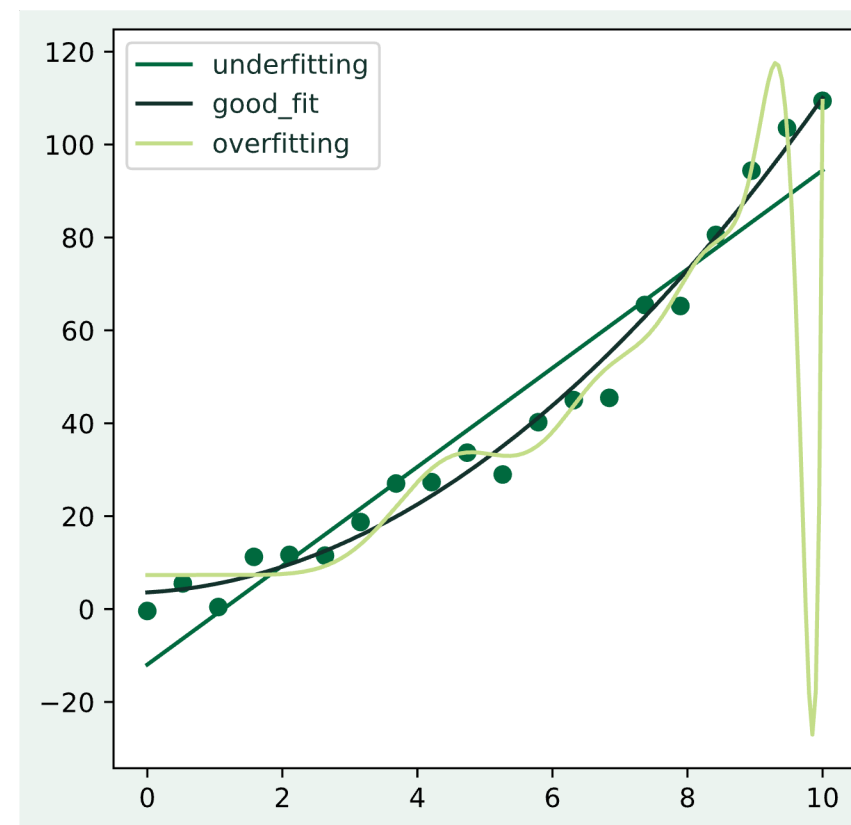
If we choose a basis function that does not have enough flexibility (i.e., parameters), it may not be able to match the shape of the data distribution (*underfitting*)

A good-fitting basis function should be smoothly going passing closely through or nearby the observations, not relying to strongly on individual points.

If the basis function "tries too hard" to fit individual points, the behavior between or beyond the observations can get weird (*overfitting*).

We need an optimal bias-variance trade-off!

We can check our model's performance on data we did not use for training (the *test data*)





Beyond this session

## The many, many, MANY kinds of regression models

- Ordinary least squares
- Regularized least squares
  - Ridge, LASSO, Elastic-Net
- Support Vector Regression
- Nearest Neighbors
- Gaussian Process Regression
- Decision Trees
- Ensemble methods (“model zoo”)

They each have individual strengths and weaknesses with regards to:

- Amount of data needed
- Computational complexity during training
- Computational complexity when making predictions
- Adaptability to new observations

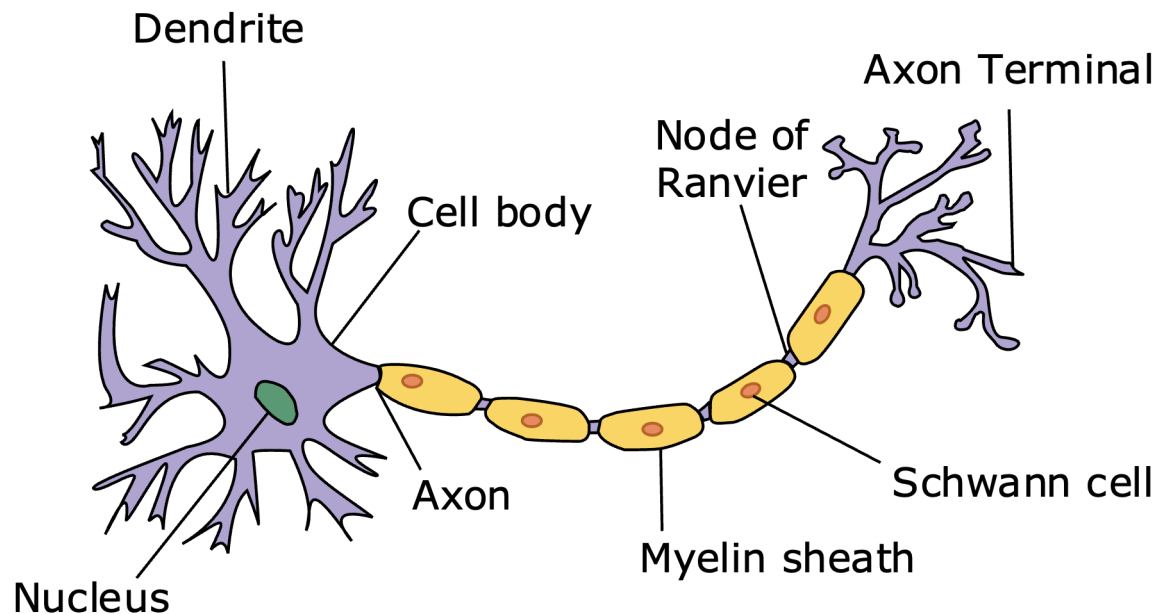
**In most cases, nobody can say in advance which model will have the highest accuracy.**

**The only way to know for sure is to try them all out!**

# Neural Networks for regression problems

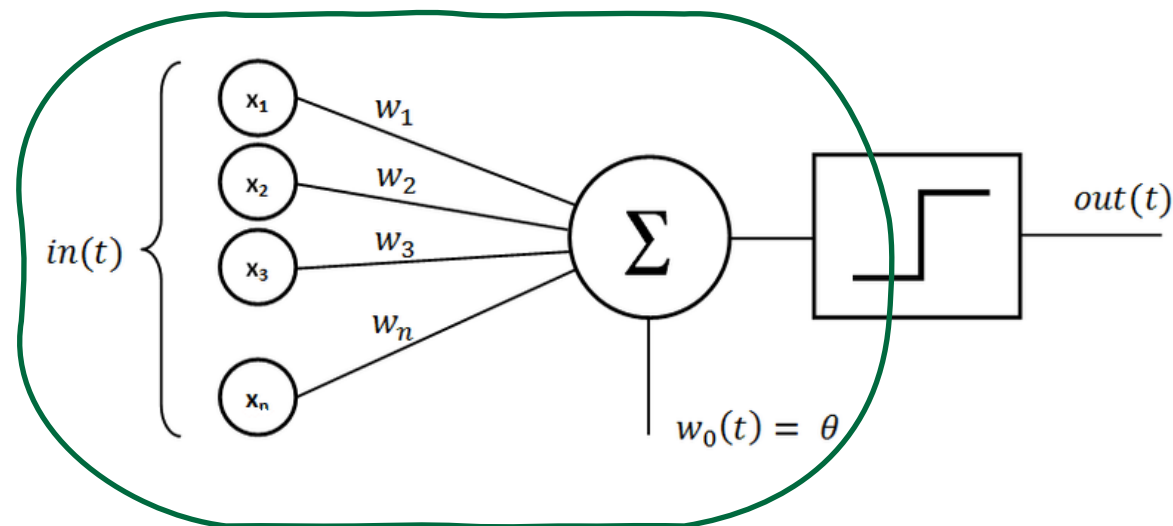
## The Perceptron

### A biological neuron



User:Dhp1080, [CC BY-SA 3.0](#), via Wikimedia Commons

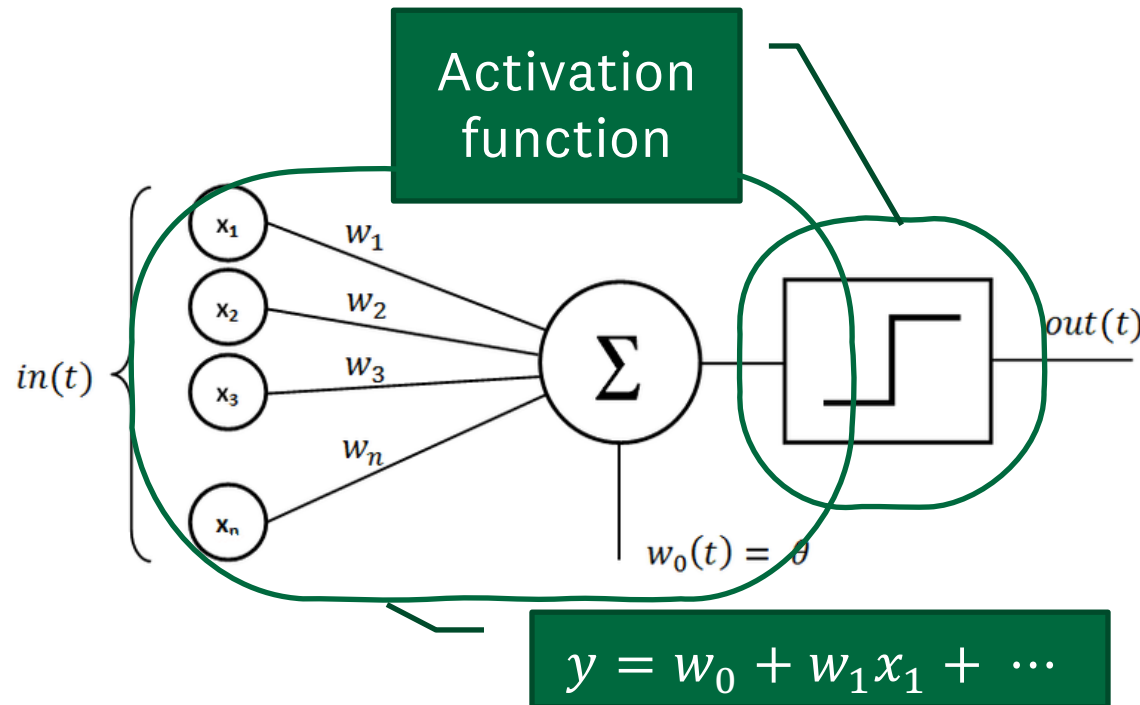
### Rosenblatt's Perceptron (1985)



Mayranna, [CC BY-SA 3.0](#), via Wikimedia Commons

# Neural Networks for regression problems

## The Perceptron



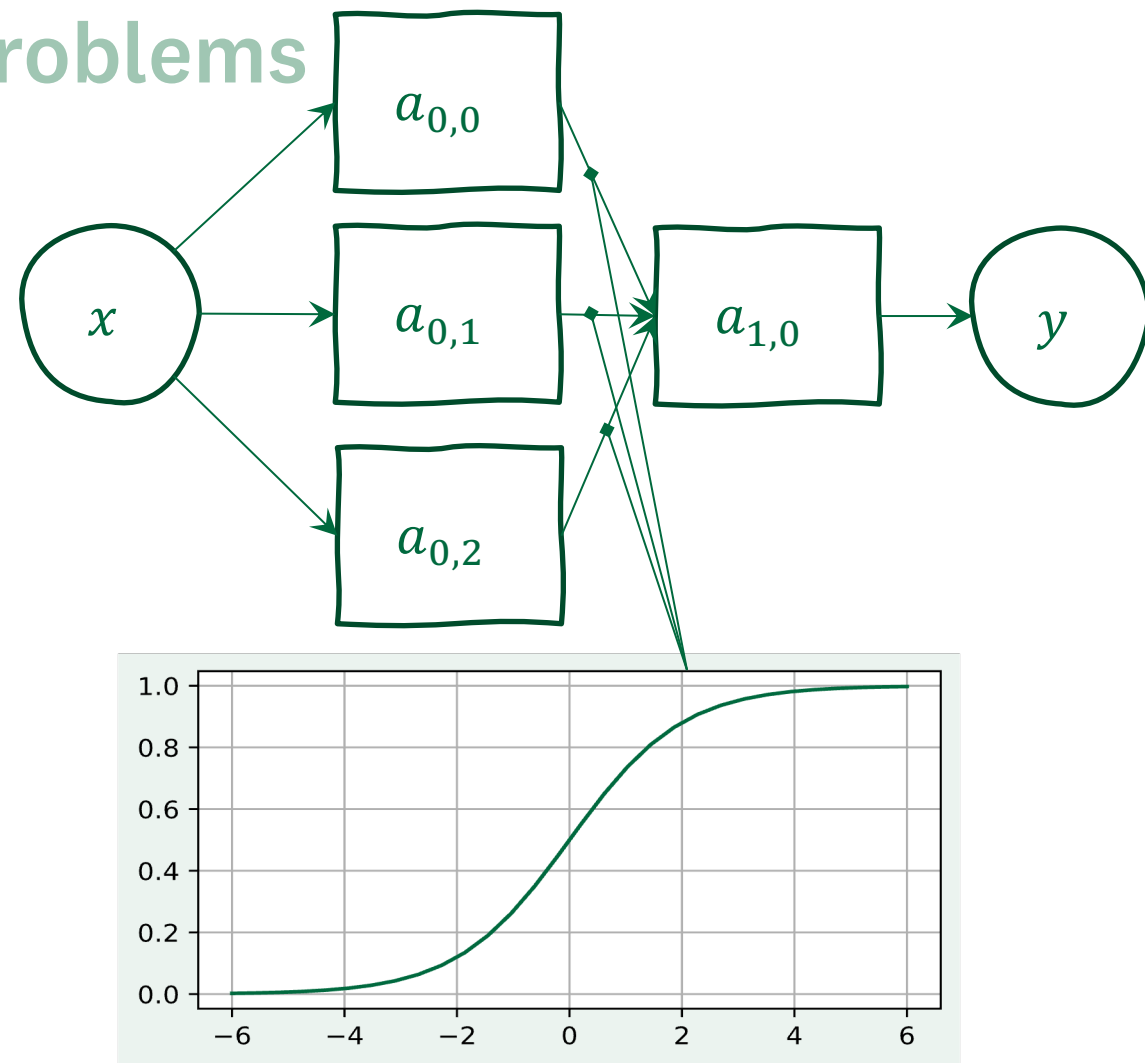
- The activation function is a non-linear function
- Without it, every artificial neuron would just be a linear regression model
- Because of the non-linearity, every neuron's output is like a unique puzzle piece
- We can bring all these pieces together by combining the outputs of several neurons and feed them to another neuron (or several more) to form a neural network
- A neural network is a universal function approximator!\*

\*given enough neurons

# Neural Networks for regression problems

## Why non-linearity matters

- A popular non-linear activation function is the logistic function
- Think of these functions as gates for the “linear regression”-like part of each neuron:
  - If the “predicted” value is within a certain range, it gets scaled
  - If the “predicted” value is below a certain threshold, it gets blocked (output 0)
  - If it is above a certain threshold, it adds a constant value
- This gating mechanism shapes the “puzzle pieces”, that later layers in the network can “put together”





# Neural Networks for regression problems

## Training and limitations

- Neural networks are trained using gradient-based methods (backpropagation and gradient descent)
- Despite the similarity to linear regression, the optimal solution cannot be found in one set of equations because of the layered structure
- The training works iteratively instead is thus time-intensive
- The flexibility comes at the cost of larger number of parameters, which requires large amounts of data (rule of thumb: one observation per parameter)
- With readily-available compute power and large amounts of data, neural networks are a frequent choice in machine learning problems



# Case study

## Diamond price prediction

Demo



## Critical thinking

# Regression

Is the model's basis function a reasonable choice?

Is the model complexity adequate?

Is there evidence for overfitting?

Is there enough data to generalize?

Can the model be interpreted?

Are the conclusions plausible?





## Summary

### Key take-aways

- Regression models describe a relationship between independent variables and a continuous dependent variable
- The relationship is described by an analytic function plus noise
- The parameters of this analytic function are found using training data (they are “learned” from data)
- Many models exist with individual strengths and weaknesses
- Neural networks are popular because they are *universal function approximators*

# References

Great intro to linear regression:

- Rumsey. (2011). *Statistics for dummies* (2nd ed.). Wiley.

In-depth book:

- Bishop. (2006). *Pattern recognition and machine learning*. Springer.

Great video series on neural networks:

- [https://www.youtube.com/playlist?list=PLZHQObOWTQDNU6R1\\_67000Dx\\_ZCJB-3pi](https://www.youtube.com/playlist?list=PLZHQObOWTQDNU6R1_67000Dx_ZCJB-3pi)