

# Programming

## Numerical Data Analysis & Visualization

Daniel Dörr

Faculty of Technology, Bielefeld University

```
332
333
334     if extrapolate is None:
335         extrapolate = self.extrapolate
336     x = np.asarray(x)
337     x_shape, x_ndim = x.shape, x.ndim
338     x = np.ascontiguousarray(x.ravel(), dtype=np
339
340     # With periodic extrapolation we map x to the
341     # [self.t[k], self.t[n]].
342     if extrapolate == 'periodic':
343         n = self.t.size - self.k - 1
344         x = self.t[self.k] + (x - self.t[self.k]) *
345
346         extrapolate = False
347
348     out = np.empty((len(x), prod(self.c.shape[1:])),
349                   dtype=self.c.dtype)
350     self._ensure_c_contiguous()
351     self._evaluate(x, nu, extrapolate, out)
352     out = out.reshape(x_shape + self.c.shape[1:])
353
354     if self.axis != 0:
355         # transpose to move the calculated values to 0
356         l = list(range(out.ndim))
357         l = l[x_ndim:x_ndim+self.axis] + l[:x_ndim] +
358             l[x_ndim+self.axis:]
359         out = out.transpose(l)
360
361     return out
362
363 def _evaluate(self, xp, nu, extrapolate, out):
364     _bspl.evaluate_spline(self.t, self.c.reshape(self.c
365
366     self.k, xp, nu, extrapolate, out)
367
368 def _ensure_c_contiguous(self):
369     """
370     Ensure that the C array is contiguous. The Cython code
371     c and t may be modified by the user. The Cython code
372     ensures that they are C contiguous.
373
374     """
375     if not self.c.flags.c_contiguous:
376         self.c = np.ascontiguousarray(self.c)
377     if not self.t.flags.c_contiguous:
378         self.t = np.ascontiguousarray(self.t)
```

# Recap

# Reading from file

Dynamic: read from file with name requested by prompt

```
1 fName = input('Input file: ')
2 lines = list()
3 f = open(fName)
4 for line in f:
5     lines.append(line)
```

# File formats

## Unstructured data

- Text

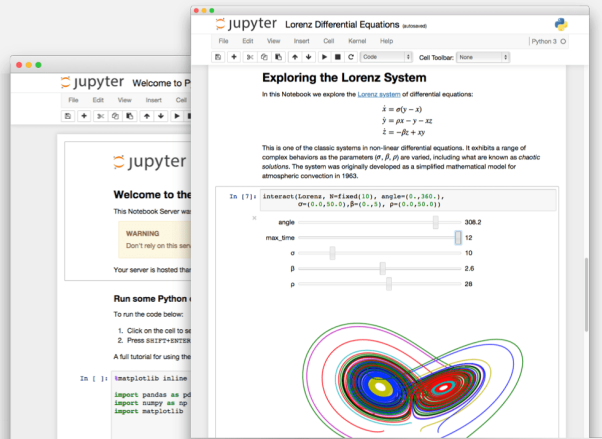
## Structured data

- XML
- JSON
- Tables
- Matrices

# Jupyter Notebook

## Why use Jupyter Notebook in Data Science?

- Simultaneous documentation & analysis
- Step-by-step processing
- Ensures reproducibility

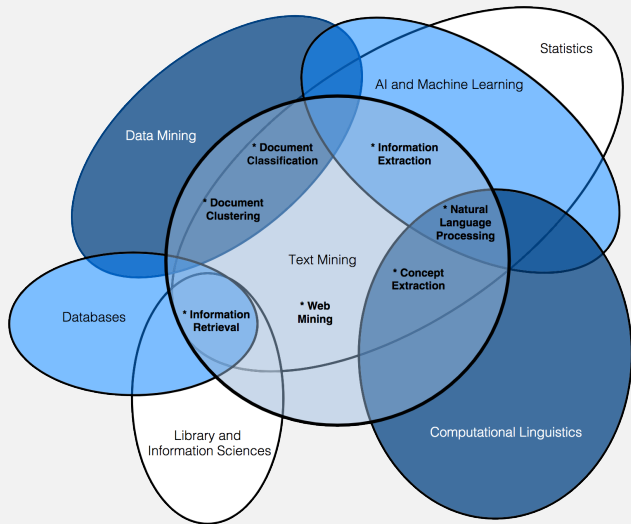


# Text mining

Relies on *Natural Language Processing* (NLP)

Main (constitutive) tasks:

- ❖ Document summarization, clustering & classification
- ❖ Information extraction
- ❖ Information discovery



source: Miner, Gary. *Practical Text Mining and Statistical Analysis for Non-structured Text Data Applications*. 1st ed. Amsterdam: Academic Press, 2012.

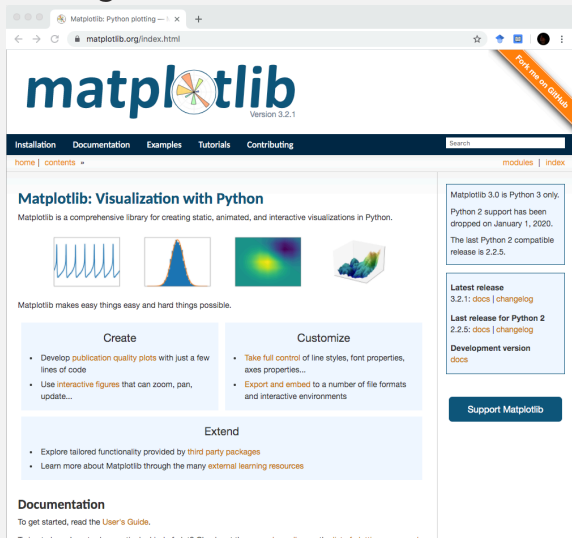
**Data  
Visualization**

**Numerical  
Data Analysis  
with NumPy**

**Modeling  
Experimental  
Data**

# Matplotlib: Visualization with Python

- ❖ de-facto standard library for scientific visualizations
- ❖ many third party packages built on top of Matplotlib
- ❖ comprehensive library for creating static, animated, and interactive visualizations



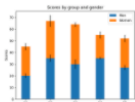
The screenshot shows the Matplotlib website homepage. At the top, the logo "matplotlib" is displayed in a large, blue, sans-serif font, with a circular icon containing a colorful plot. Below the logo, the version "Version 3.2.1" is noted. A navigation bar includes links for "Installation", "Documentation", "Examples", "Tutorials", and "Contributing". A search bar is located on the right side of the navigation bar. Below the navigation bar, the main content area features the heading "Matplotlib: Visualization with Python" and a sub-heading "Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python." Four small images illustrate different types of plots: a line plot, a histogram, a heatmap, and a 3D surface plot. Below these images, the text "Matplotlib makes easy things easy and hard things possible." is displayed. Two main sections, "Create" and "Customize", provide bullet points describing the library's capabilities. The "Create" section mentions developing publication quality plots and using interactive figures. The "Customize" section highlights taking full control of line styles and font properties, and exporting plots to various file formats. Below these sections, an "Extend" section mentions exploring tailored functionality provided by third party packages and learning more about Matplotlib through external learning resources. A "Documentation" section at the bottom suggests reading the User's Guide. On the right side of the page, there are two boxes: one for "Matplotlib 3.0 is Python 3 only" and another for "Latest release 3.2.1: docs | changelog". A "Support Matplotlib" button is located at the bottom right of the page.

source: <https://matplotlib.org/>

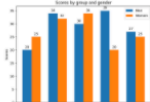


# Matplotlib: Visualization with Python

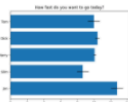
## Lines, bars and markers



Stacked bar chart



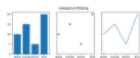
Grouped bar chart  
with labels



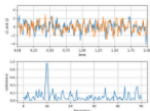
Horizontal bar chart



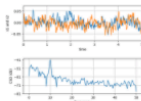
Broken Barh



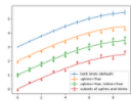
Plotting categorical  
variables



Plotting the  
coherence of two  
signals



CSD Demo

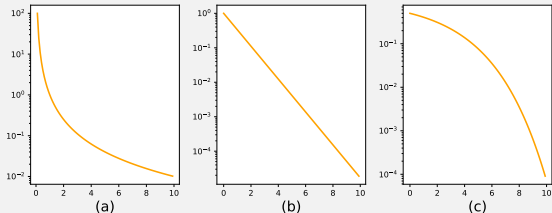


Errorbar limit  
selection

source: <https://matplotlib.org/gallery/index.html>

# Quiz

- Assign the decay functions to their shapes when plotted on a logarithmic scale:



exponential

polynomial

super-exponential

- Which of these `matplotlib (plt)` functions can be used to draw a scatter plot?

`plt.dot()`

`plt.plot()`

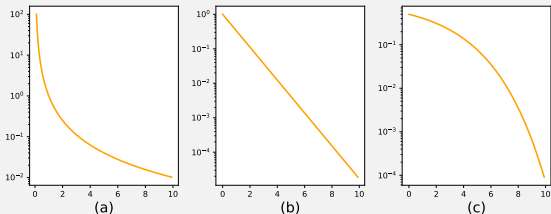
`plt.draw()`

`plt.show()`

- Specify the `matplotlib (plt)` command to draw a dotted blue line defined by lists `X` and `Y`.

# Quiz

- Assign the decay functions to their shapes when plotted on a logarithmic scale:



exponential (b)

polynomial (a)

super-exponential (c)

- Which of these `matplotlib` (`plt`) functions can be used to draw a scatter plot?

`plt.dot()`

`plt.plot()` ✓

`plt.draw()`

`plt.show()`

- Specify the `matplotlib` (`plt`) command to draw a dotted blue line defined by lists `X` and `Y`.

```
plt.plot(X, Y, ':', color='blue')
```

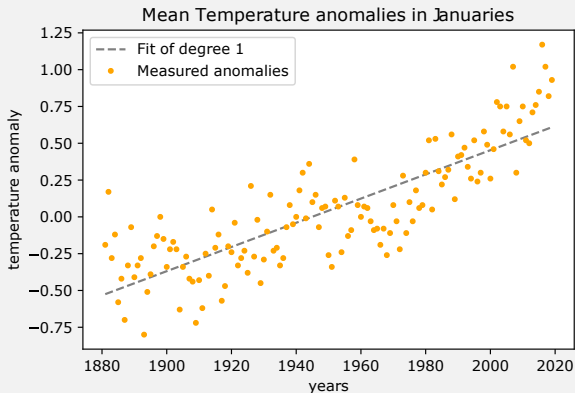
**Data  
Visualization**

**Numerical  
Data Analysis  
with NumPy**

**Modeling  
Experimental  
Data**

# Numerical analysis

- Numerical data: anything measurable
- Methods:
  - Interpolation and regression
  - Solving differential equations
  - Optimization
  - ...



# N-dimensional array: `numpy.ndarray`

## Array data structure

- ❖ immutable
- ❖ n-dimensional
- ❖ very storage efficient
- ❖ can store only data of same type

# Quiz

- Given the following variables: `a = np.array((1, 2, 3))`,  
`b = np.array(((4, 5, 6), (7, 8, 9)))`, `c = 10`, specify the following  
statements use *vectorized operation* or *broadcasting*:
  - `a * b`
  - `a ** a`
  - `b * c`
  - `a / b[1]`
- Determine the data type of the following NumPy arrays:
  - `np.array((1, ))`
  - `np.array((1.0, ))`
  - `np.array((1, 1.0, (1, 1.0)))`
  - `np.array(((1, 1.0), (1, 1.0)))`

# Quiz

- Given the following variables: `a = np.array((1, 2, 3))`,  
`b = np.array(((4, 5, 6), (7, 8, 9)))`, `c = 10`, specify the following  
statements use *vectorized operation* or *broadcasting*:
- `a * b` broadcasting
  - `a ** a` vectorized op.
  - `b * c` broadcasting
  - `a / b[1]` vectorized op.
- Determine the data type of the following NumPy arrays:
- `np.array((1, ))` int64
  - `np.array((1.0, ))` float64
  - `np.array((1, 1.0, (1, 1.0)))` object
  - `np.array(((1, 1.0), (1, 1.0)))` float64



**Data  
Visualization**

**Numerical  
Data Analysis  
with NumPy**

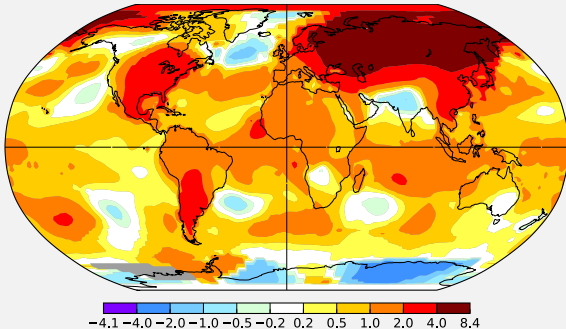
**Modeling  
Experimental  
Data**

# NASA's GISS Surface Temperature Analysis

March 2020

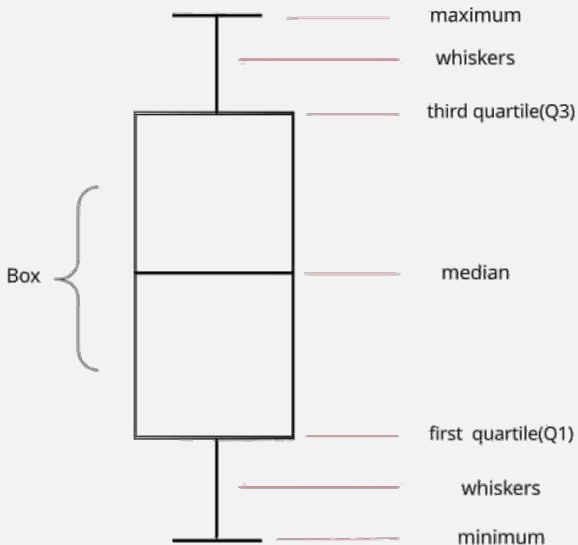
L-OTI(°C) Anomaly vs 1951-1980

1.18



- ❖ <https://data.giss.nasa.gov/gistemp>
- ❖ Collection of temperature data from thousands of meteorological stations
- ❖ Data represents *anomalies*, i.e., deviations from mean temperature measured in 1951-1980

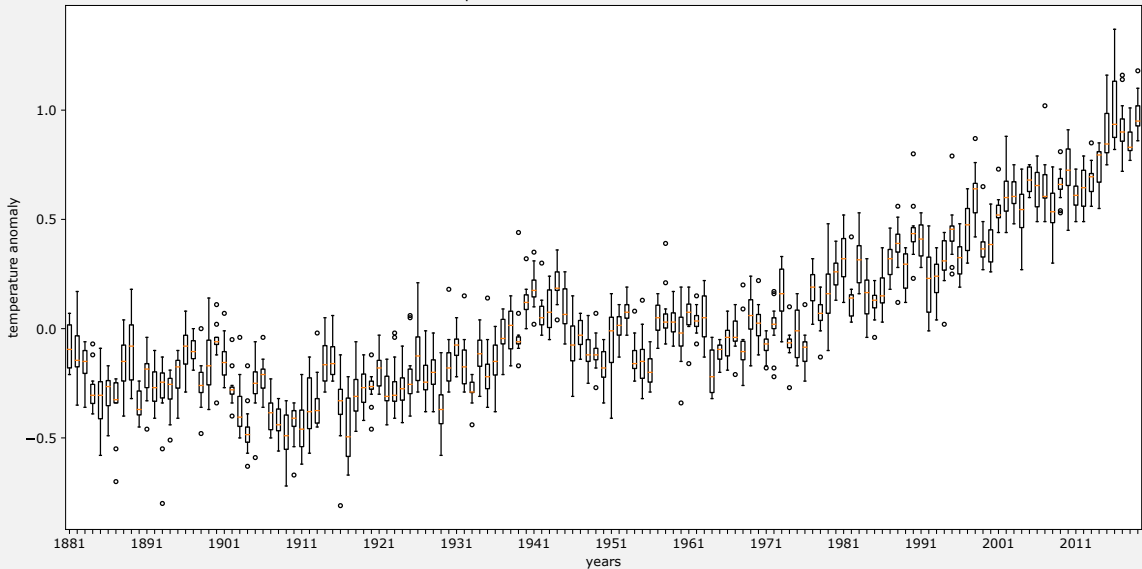
# Whisker plot



source: <https://aiaspirant.com/box-plot/>

# Whisker plot of GISS data

Temperature anomalies between 1881-2019



# Further classes on methods in Data Science

## Classes at Bielefeld University:

- ❖ 311850, Multivariate Methods / Regression Analysis (winter semester)
- ❖ 392140, Introduction to Machine Learning (winter semester)
- ❖ 392219 Applied Optimization (winter semester)

## MIT Course 6.0002, Lectures on Understanding Experimental Data:

- ❖ <https://www.youtube.com/v/vIFKGF11Cn8>
- ❖ <https://www.youtube.com/v/fQvg-hh9dUw>

# Linear regression

Estimate a *linear number* of parameters from given input, e.g.:

$$X = \begin{pmatrix} X_0 \\ X_1 \\ \vdots \\ X_{N-1} \end{pmatrix}, Y = \begin{pmatrix} Y_0 \\ Y_1 \\ \vdots \\ Y_{N-1} \end{pmatrix} \rightarrow \alpha = \begin{pmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_{N-1} \end{pmatrix}$$

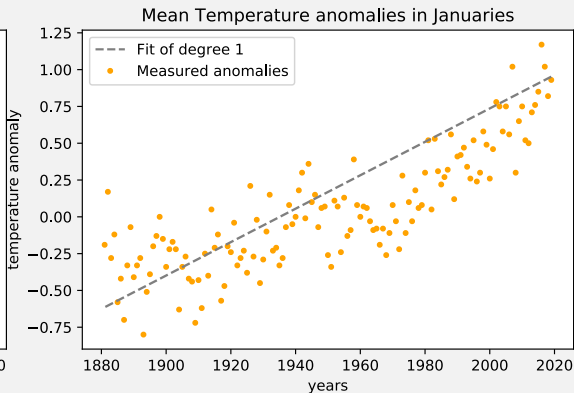
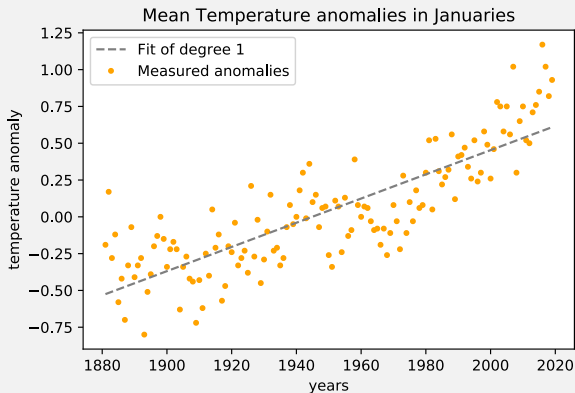
Estimator:

$$\hat{Y} = \alpha_0 + \alpha_1 X + \alpha_2 X^2 + \dots + \alpha_{N-1} X^{N-1}$$

**Simple linear regression:** Estimate line, i.e, estimate  $\alpha_0, \alpha_1$  and set

$$\alpha_2 = \dots = \alpha_{N-1} = 0$$

# What criterion to optimize?

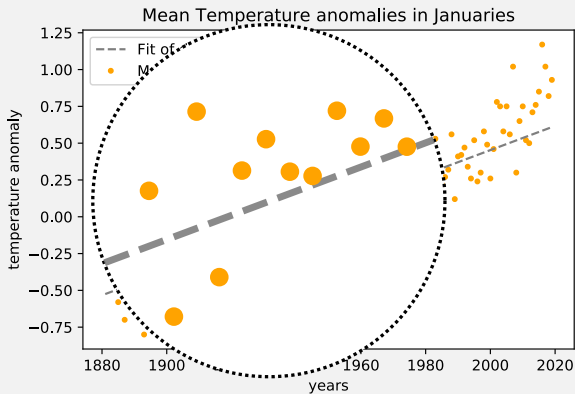


# Optimization criteria

- ❖ Residual: difference predicted/observed  $|Y_i - \hat{Y}_i|$
- ❖ Possible minimization criteria:
  - ❖ Sum of residuals
  - ❖ Maximum
  - ❖ Variance of residuals

$$\text{Var}_{res} := \frac{1}{N} \sum_i (Y_i - \hat{Y}_i)^2 = E[(Y - \hat{Y})^2]$$

- ❖ Minimize  $\text{Var}_{res}$  = ordinary least squares optimization



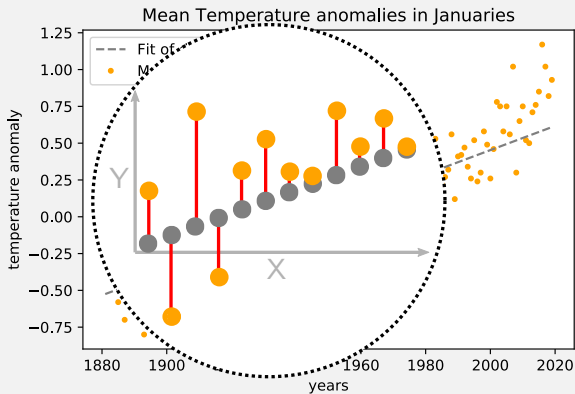


# Optimization criteria

- ❖ Residual: difference predicted/observed  $|Y_i - \hat{Y}_i|$
- ❖ Possible minimization criteria:
  - ❖ Sum of residuals
  - ❖ Maximum
  - ❖ Variance of residuals

$$\text{Var}_{res} := \frac{1}{N} \sum_i (Y_i - \hat{Y}_i)^2 = E[(Y - \hat{Y})^2]$$

- ❖ Minimize  $\text{Var}_{res}$  = ordinary least squares optimization



# Coefficient of determination $R^2$

How to measure quality of fit?

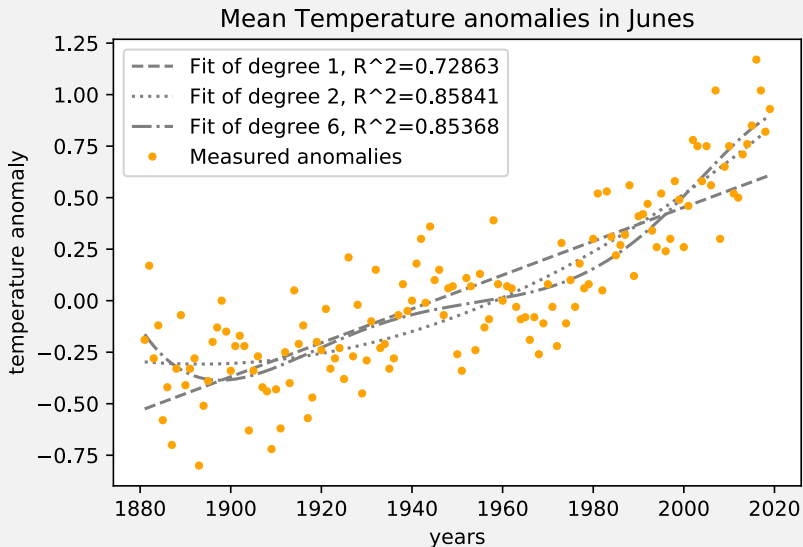
Recall: Ordinary Least squares optimization minimizes  $Var_{res}$

$R^2$  is a normalized measure thereof:

$$R^2 := 1 - \frac{Var_{res}}{Var(Y)}$$

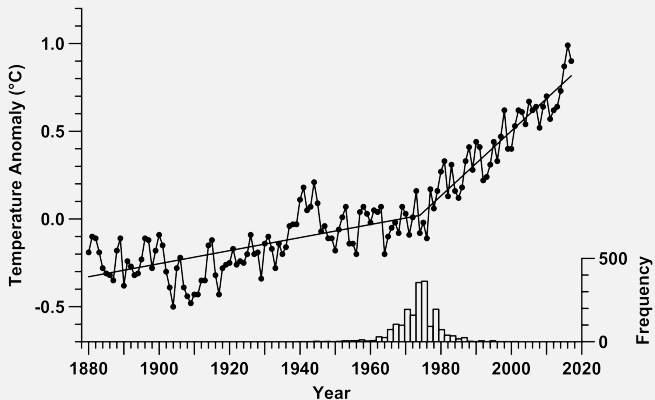
i.e,  $R^2 \rightarrow 0$  bad fit,  $R^2 \rightarrow 1$  good fit

# Separated training from testing



# The “break” model

- ❖ Combination of two linear functions
- ❖ Estimated break around year  $1974 \pm 5.9$



source: Mudelsee, M. (2019). Trend analysis of climate time series: A review of methods. *Earth Science Reviews*, 190, 310–322.

# Quiz

## ❖ True or false?

- ❖ The residual is the distance between an observed and its predicted data point
- ❖ Linear regression always minimizes the variance of residuals
- ❖ Linear regression is the task of fitting a line to a set of data points
- ❖ Ordinary least squares always minimizes the variance of residuals

## ❖ How does linear regression measure the distance between an observed and its predicted data point?



(a)



(b)

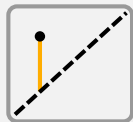


(c)

# Quiz

## True or false?

- ❖ The residual is the distance between an observed and its predicted data point true
  - ❖ Linear regression always minimizes the variance of residuals false
  - ❖ Linear regression is the task of fitting a line to a set of data points false
  - ❖ Ordinary least squares always minimizes the variance of residuals true
- ❖ How does linear regression measure the distance between an observed and its predicted data point? (a)



(a)



(b)



(c)

# Recap

# Summary

- ❖ Plots with `matplotlib`:
  - ❖ Line- and scatter plot
  - ❖ Histogram
  - ❖ Whisker (box) plot
- ❖ Numpy
  - ❖ `ndarray` data type
  - ❖ Vectorized operations, broadcasting
  - ❖ Curve fitting: `polyfit()`
- ❖ Realistic data analysis: climate trends