Lecture 7: Linear Regression

Using Global Warming as a Case Study

SOES3042/6025 Part II, Lecturer: Duo Chan

Road Map of the Statistics Part

	Lecture 5	Lecture 6
Quantification Technique	Mean, variance, skewness, & kurtosis	Pearson's Correlation (Linear relationship)
Uncertainty & Significance	Gaussian distribution Chi-2 distribution	r, p = scipy.stats.pearsonr(x, y)
Assumptions	Data is Gaussian or follows specific types of distribution Independent Sampling	Data is Gaussian Independent Sampling
Test assumptions	K-S test	
Treatment		Bootstrapping

Data is Gaussian → K-S test → Bootstrapping

What will be covered in this lecture?

1. Regression to find the trend

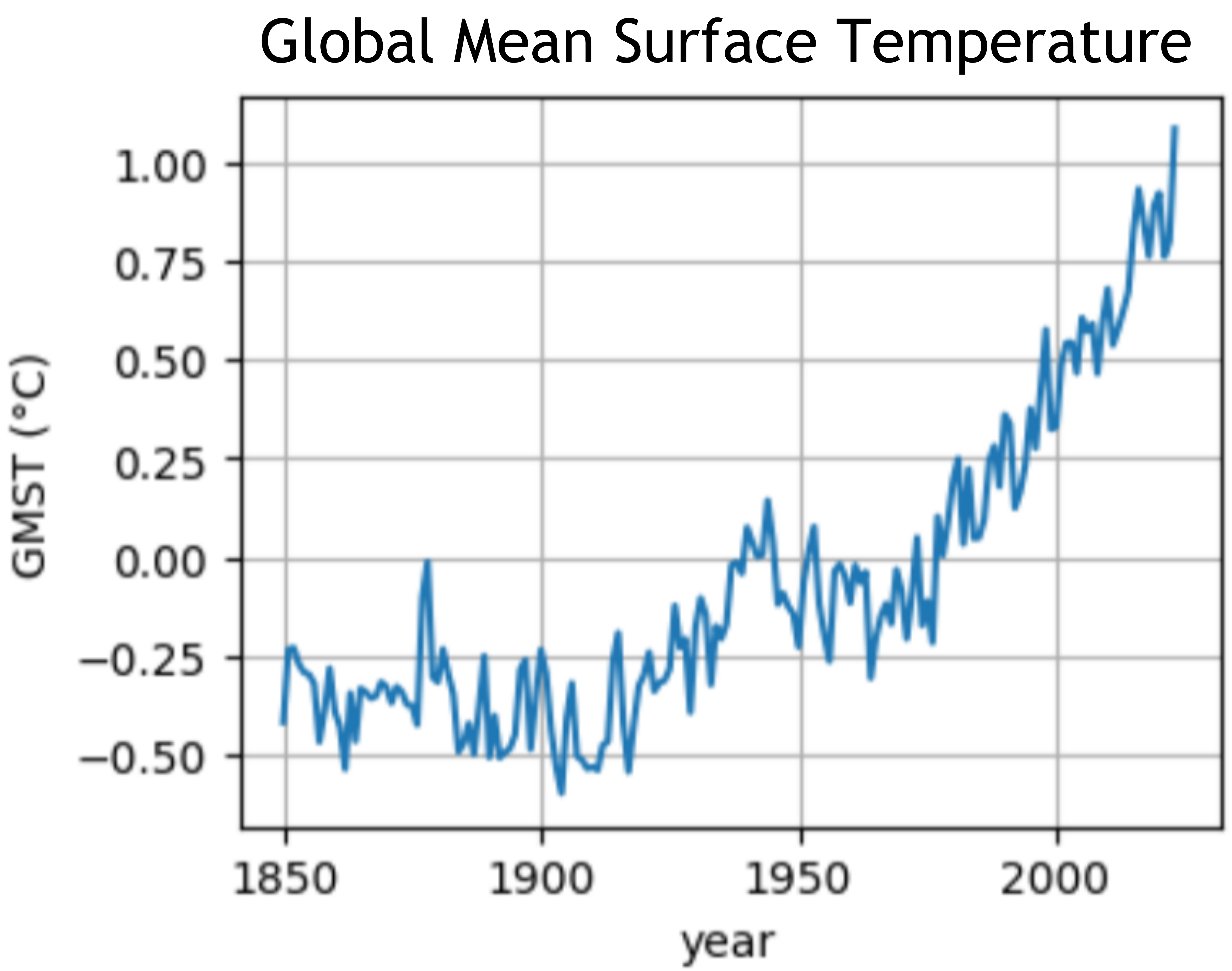
2. Uncertainty of regression analysis

3. Minimizing Square Loss and Gaussian Likelihood

4. Assumptions of Ordinary Least Squares and their validation

    -> Auto-correlation / Effective Sample Size / Block Bootstrapping

Earth's Surface Temperature has been Increasing



How fast is the Global Mean Surface Temperature (GMST) changing with time?

Regression to Find the Trend

Find the **function of predictors** that **fit the predicted variable the most**.

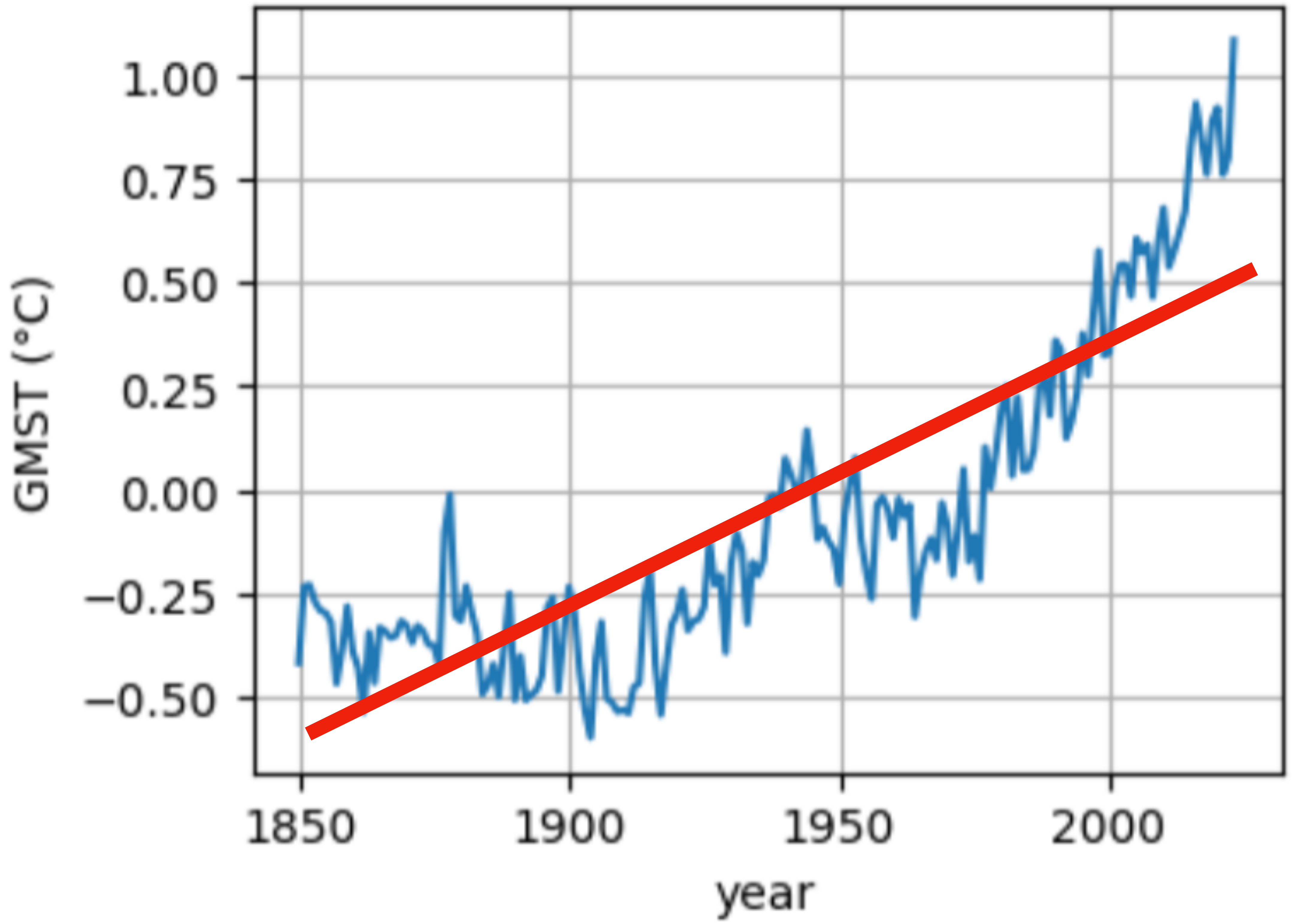
$$y = F(x) + \varepsilon$$

1 Choose a functional form → **2** Define what it means by fit → **3** Find that "most"

1. Choose a Functional Form

The simplest is to find a linear-relationship

$$T = \alpha t + \mu$$



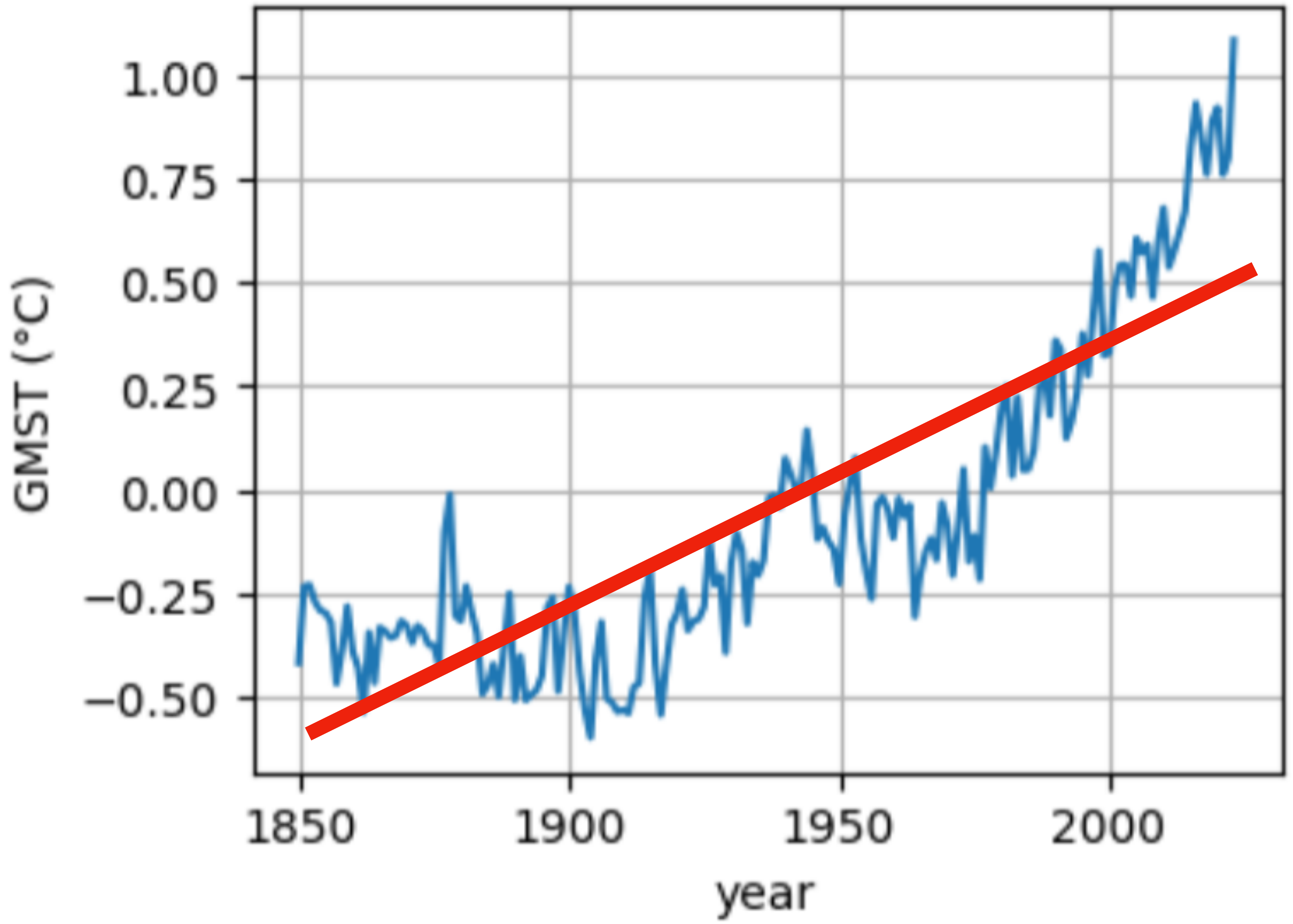
Linear regression vs. **Non-linear regression**

Find **only** scaling factors	Find **more than** scaling factors
$y = ae^{2x}$	$y = ae^{bx}$

Fitting a line is linear regression, but linear regression is more than fitting a line.

2. Defining Loss

Quantify the closeness of alignment between data and the regression line.



Squared Loss: $L(\alpha, \mu) = \sum_{i=1}^{n} (T_i - \hat{T}_i)^2.$

Variance: $s^2 = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{X})^2$

Variance of data relative to the regression line,

also called the **Mean Squared Error (MSE).**

3. Optimization and find the solution

When we are fitting a line:

$$\alpha = \frac{\mathrm{Cov}(x, y)}{\mathrm{Var}(x)}$$

$$\mu = \bar{y} - \alpha\bar{x}$$

Ordinary Least Square (OLS) Solution

```
import statsmodels.api as sm
years_matrix = sm.add_constant(years)
model = sm.OLS(GMST, years_matrix)
results = model.fit()
GMST_hat = results.fittedvalues
```

3. Optimization and find the solution

When we are fitting a line:

$$\alpha = \frac{\mathrm{Cov}(x, y)}{\mathrm{Var}(x)}$$

$$\mu = \bar{y} - \alpha\bar{x}$$

Ordinary Least Square (OLS) Solution

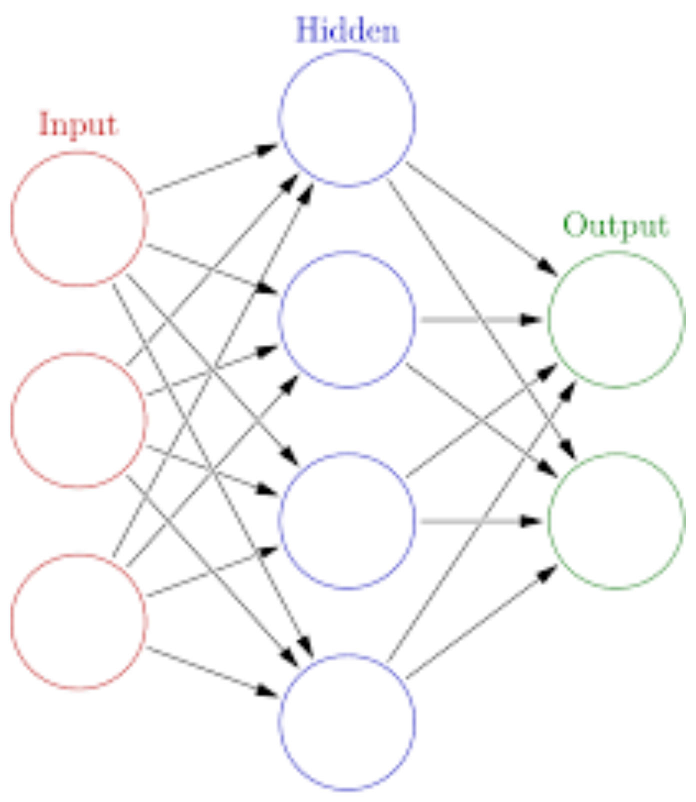
1 Choose functional form → 2 Define what it means by fit → 3 Find that "most"

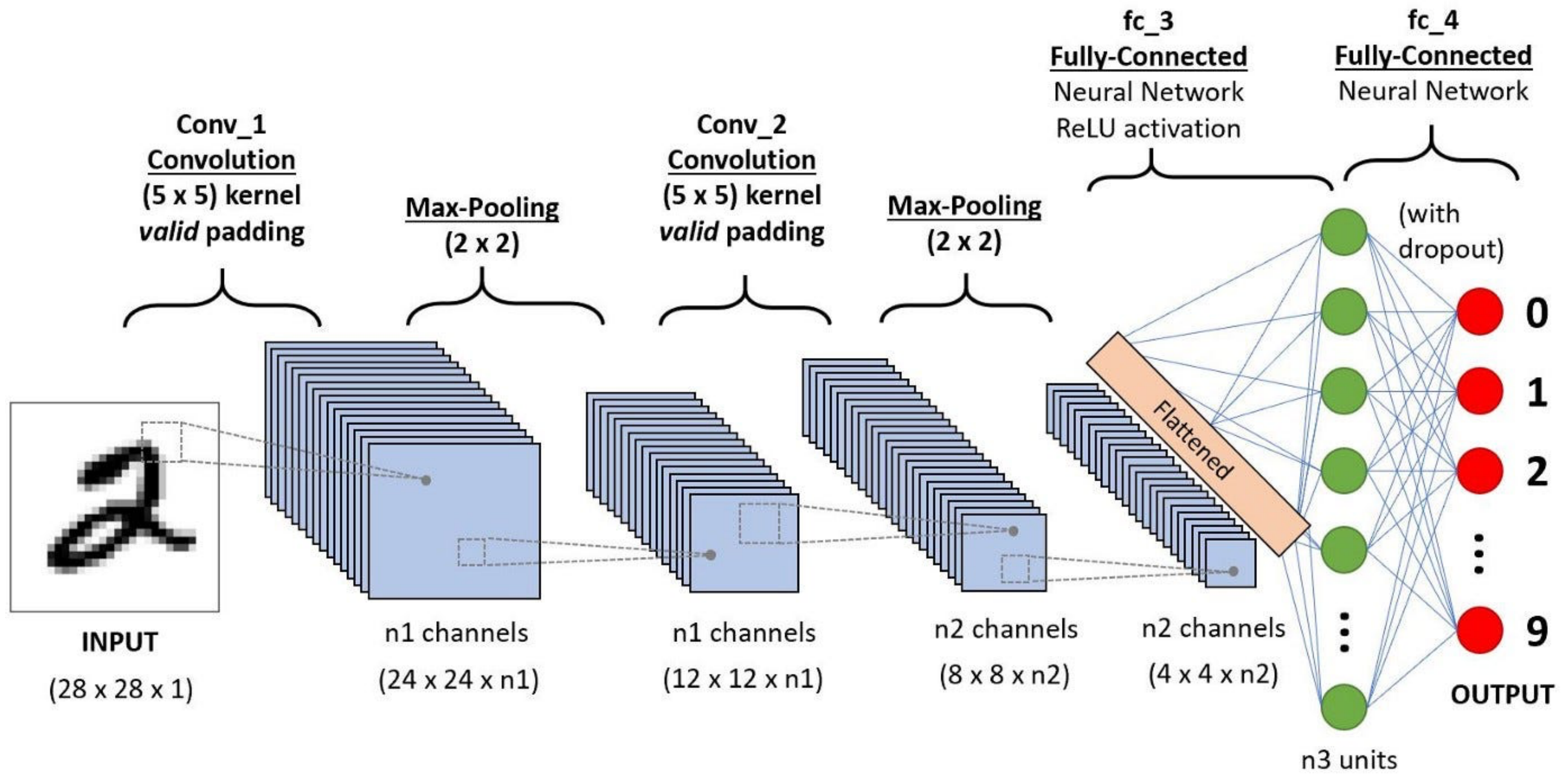
```
import statsmodels.api as sm
years_matrix = sm.add_constant(years)
model = sm.OLS(GMST, years_matrix)
results = model.fit()
GMST_hat = results.fittedvalues
```

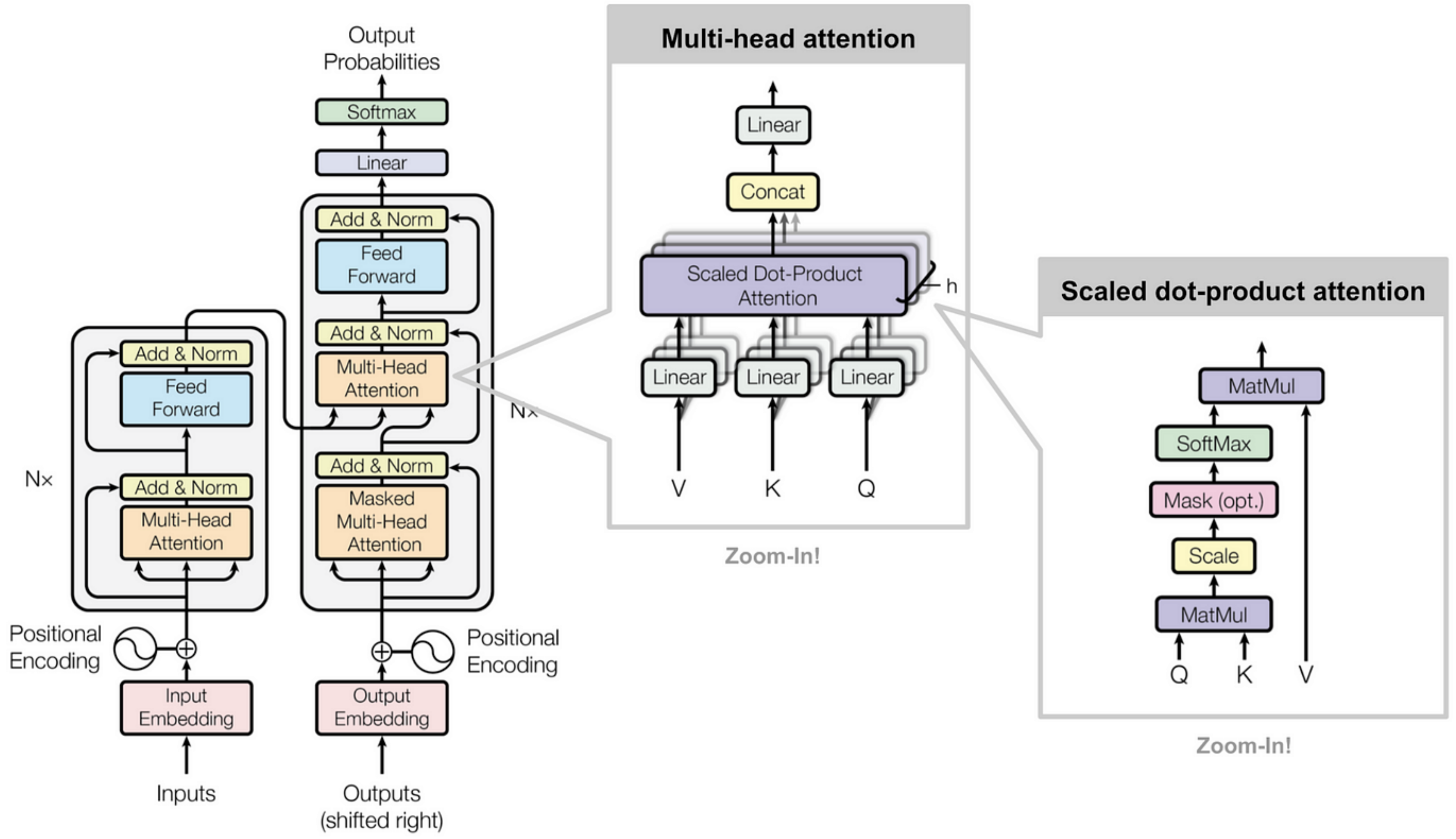
The three steps are the key philosophy of data science and machine learning

$$y = F(x) + \varepsilon$$

1 Choose a functional form → **2** Define what it means by fit → **3** Find that "most"







Understanding the Summary of the Ordinary Least Square Estimate

```
print(results.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.714
Model:                            OLS   Adj. R-squared:                  0.712
Method:                 Least Squares   F-statistic:                     428.7
Date:                Mon, 05 Feb 2024   Prob (F-statistic):           1.40e-48
Time:                        11:34:14   Log-Likelihood:                 31.252
No. Observations:                 174   AIC:                            -58.50
Df Residuals:                     172   BIC:                            -52.19
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         -0.0722      0.015     -4.686      0.000      -0.103      -0.042
x1             0.0064      0.000     20.706      0.000       0.006       0.007
==============================================================================
Omnibus:                        4.837   Durbin-Watson:                   0.335
Prob(Omnibus):                  0.089   Jarque-Bera (JB):                4.856
Skew:                           0.376   Prob(JB):                       0.0882
Kurtosis:                       2.679   Cond. No.                         50.2
==============================================================================
```

Overall significance of the entire model

Significance of individual parameters

Assumptions behind Ordinary Least Squares (OLS)

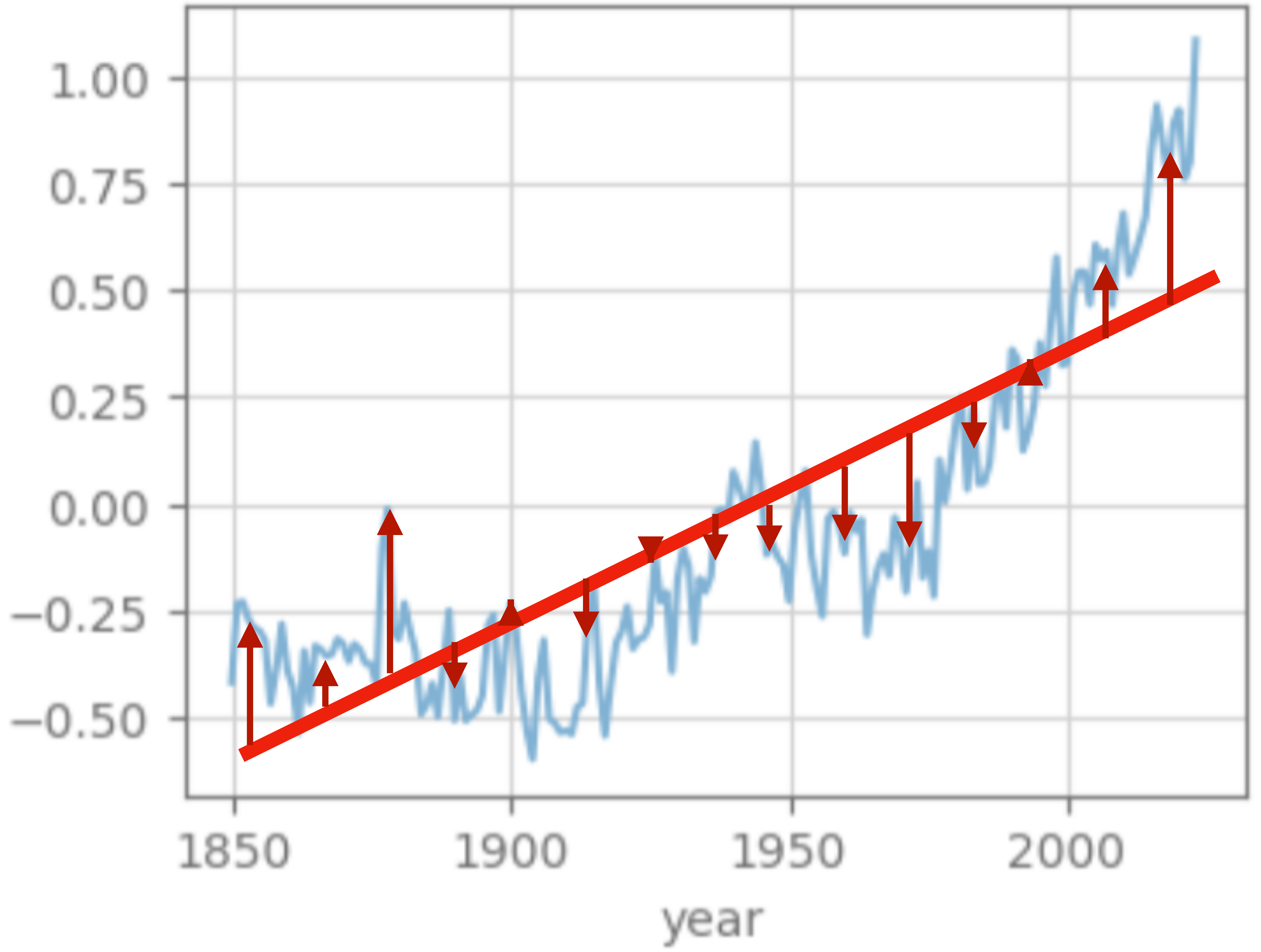
(1) X is noise free

(2) Errors follow Gaussian distribution

(3) Errors are independent with each other

(4) Errors have the same variance

Data is Gaussian → K-S test → Bootstrapping



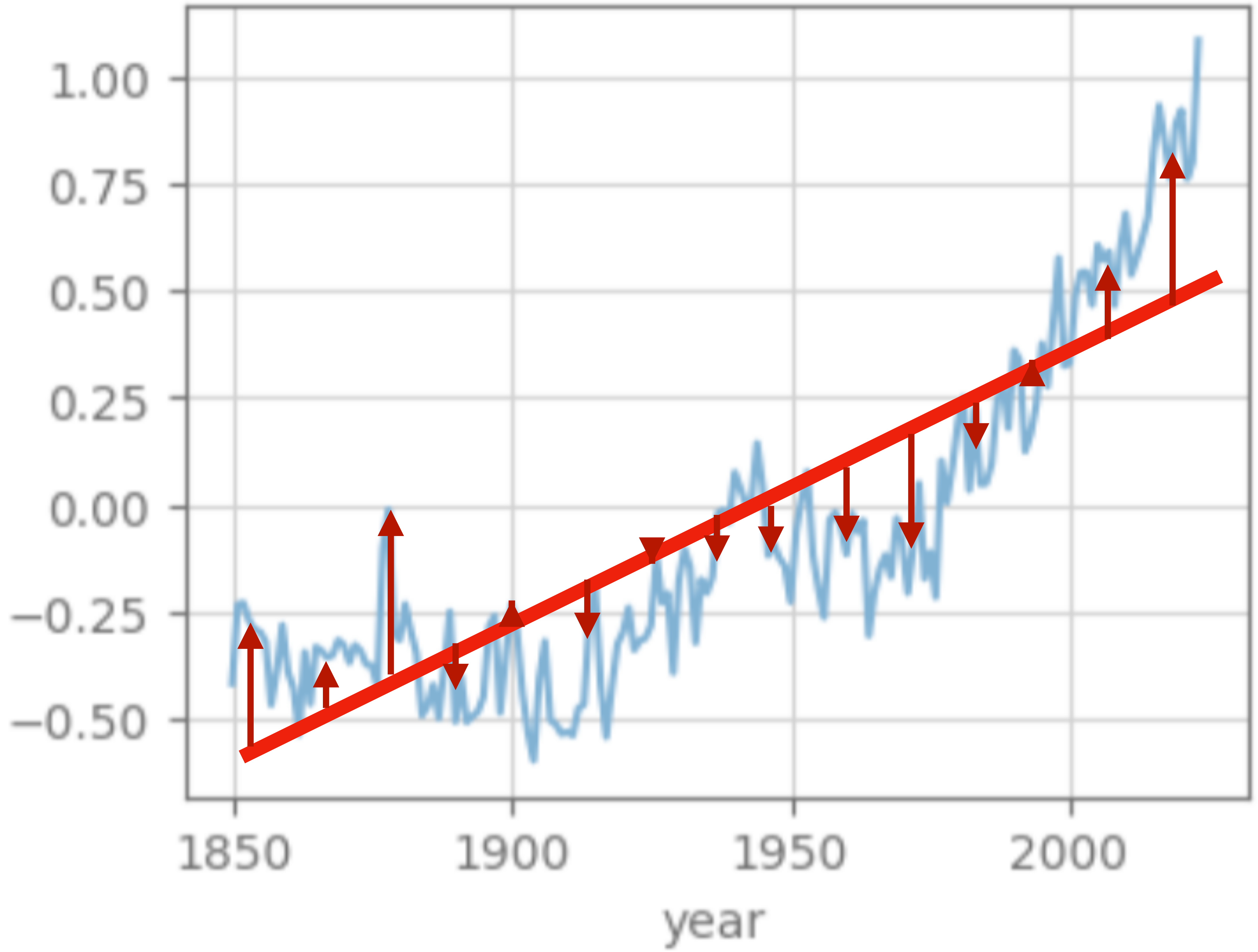
Assumptions behind Ordinary Least Squares (OLS)

(1) X is noise free

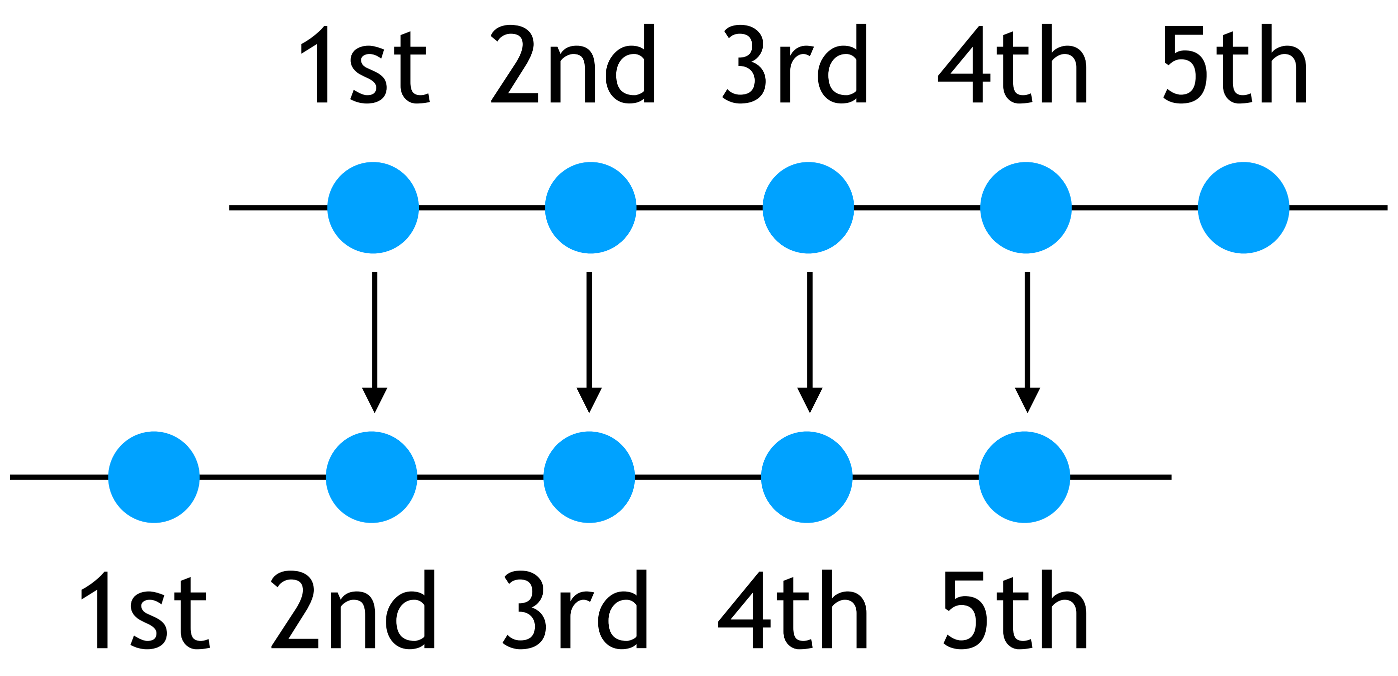
(2) Errors follow Gaussian distribution

(3) Errors are independent with each other

(4) Errors have the same variance

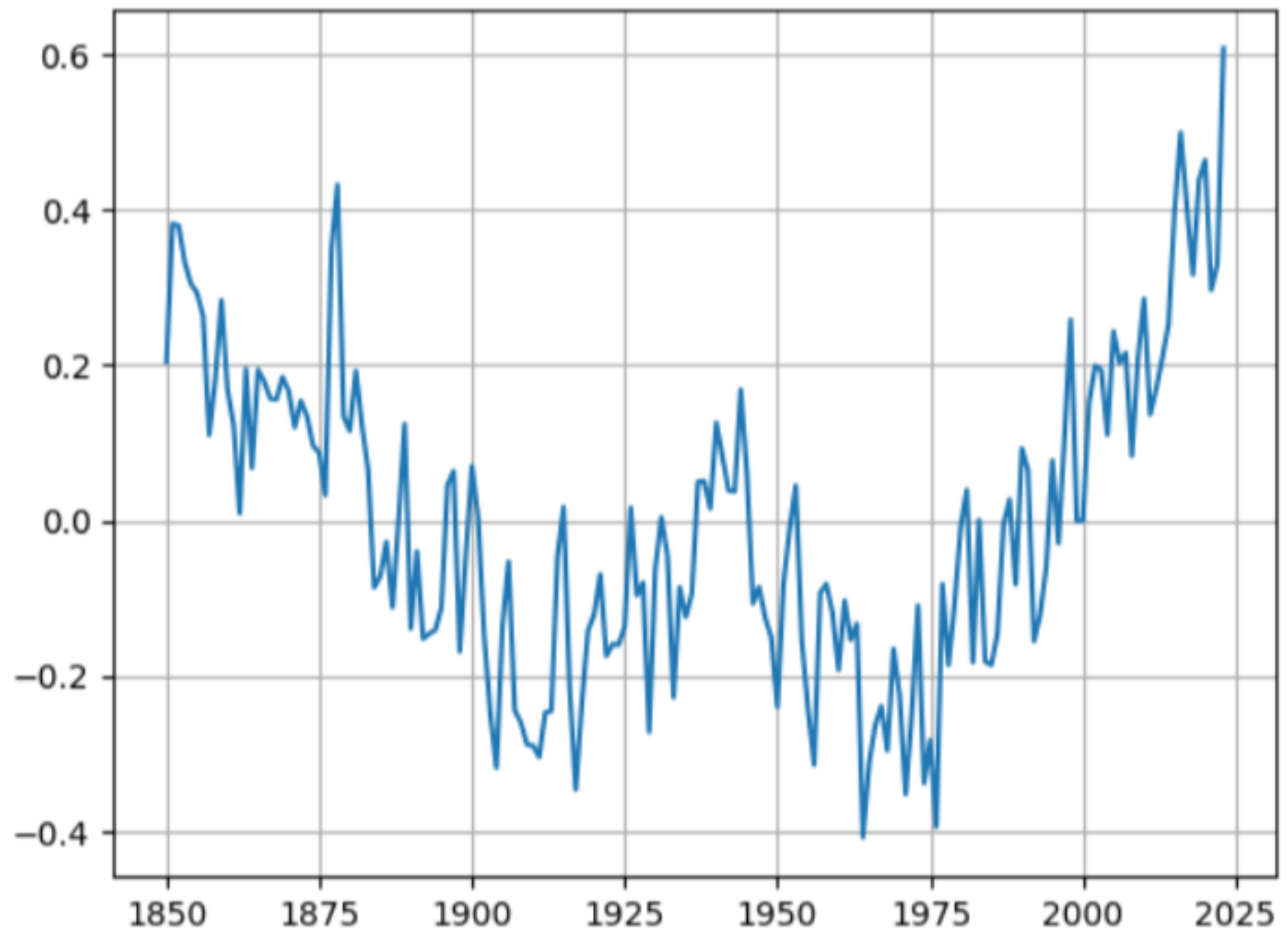


Whether errors are independent with each other?

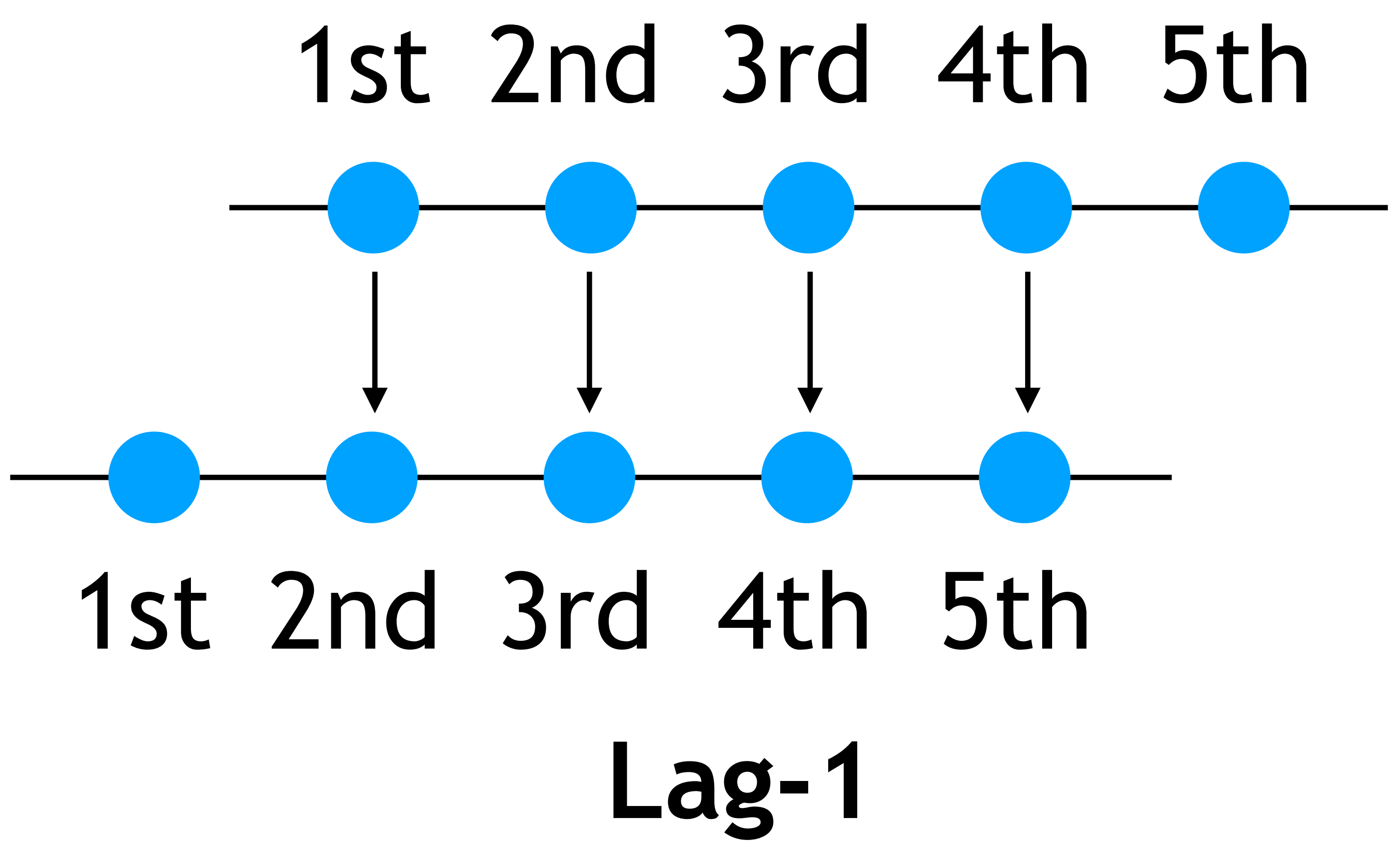


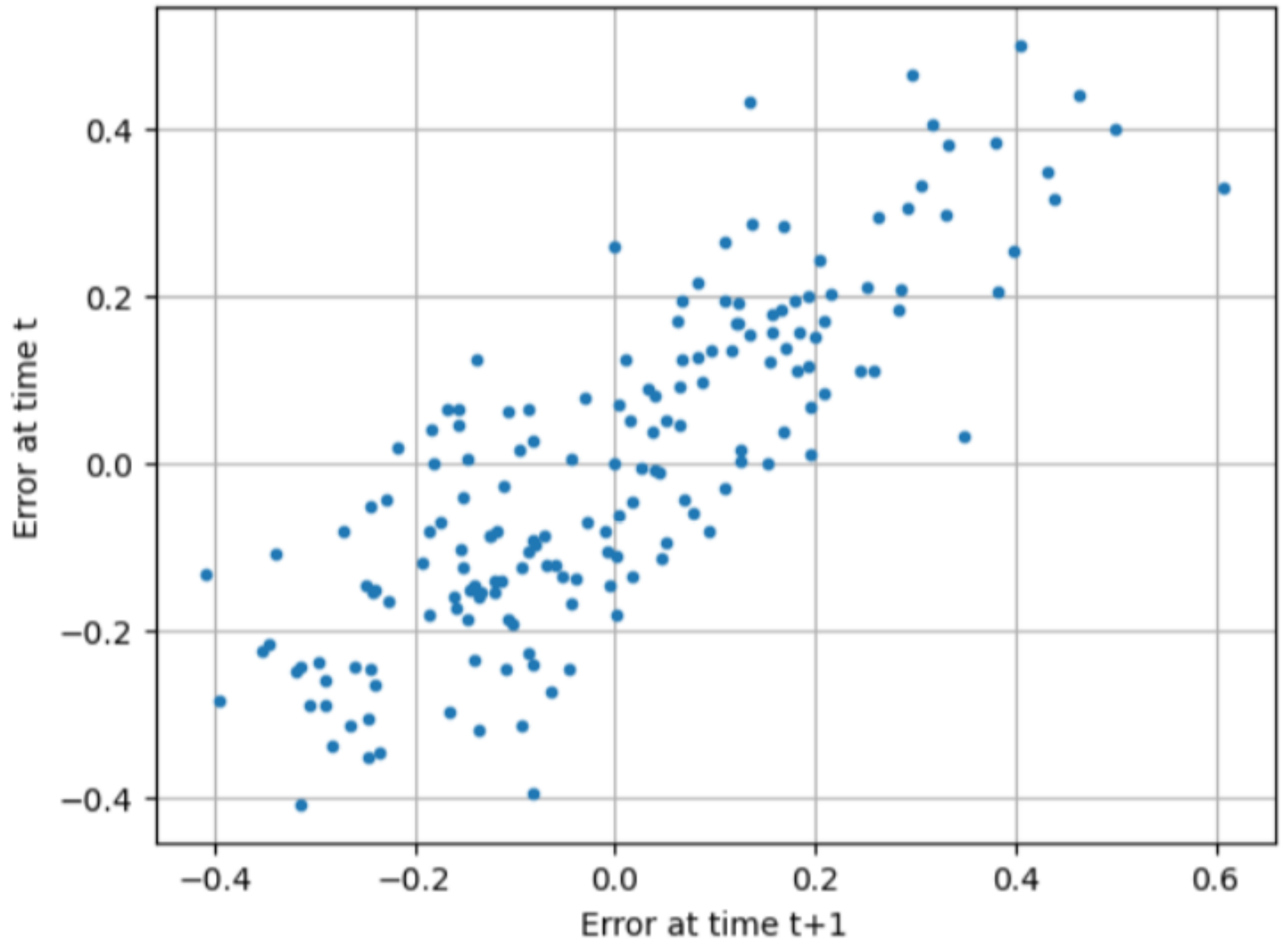
Auto-correlation: Testing independence of error across time

Statistics	Definition	Meaning	Functions to use
Auto-Correlation	$E[X_{std}X_{std}^{+\tau}]$	Measures how a variable's current value is related to its past value at a time lag of τ .	pearsonr(data1[0:-τ], data1[τ:])
Correlation	$E[X_{std}Y_{std}]$	the degree to which two variables are linearly associated	pearsonr(data1, data2) **or** dataframe[select_columns].corr()

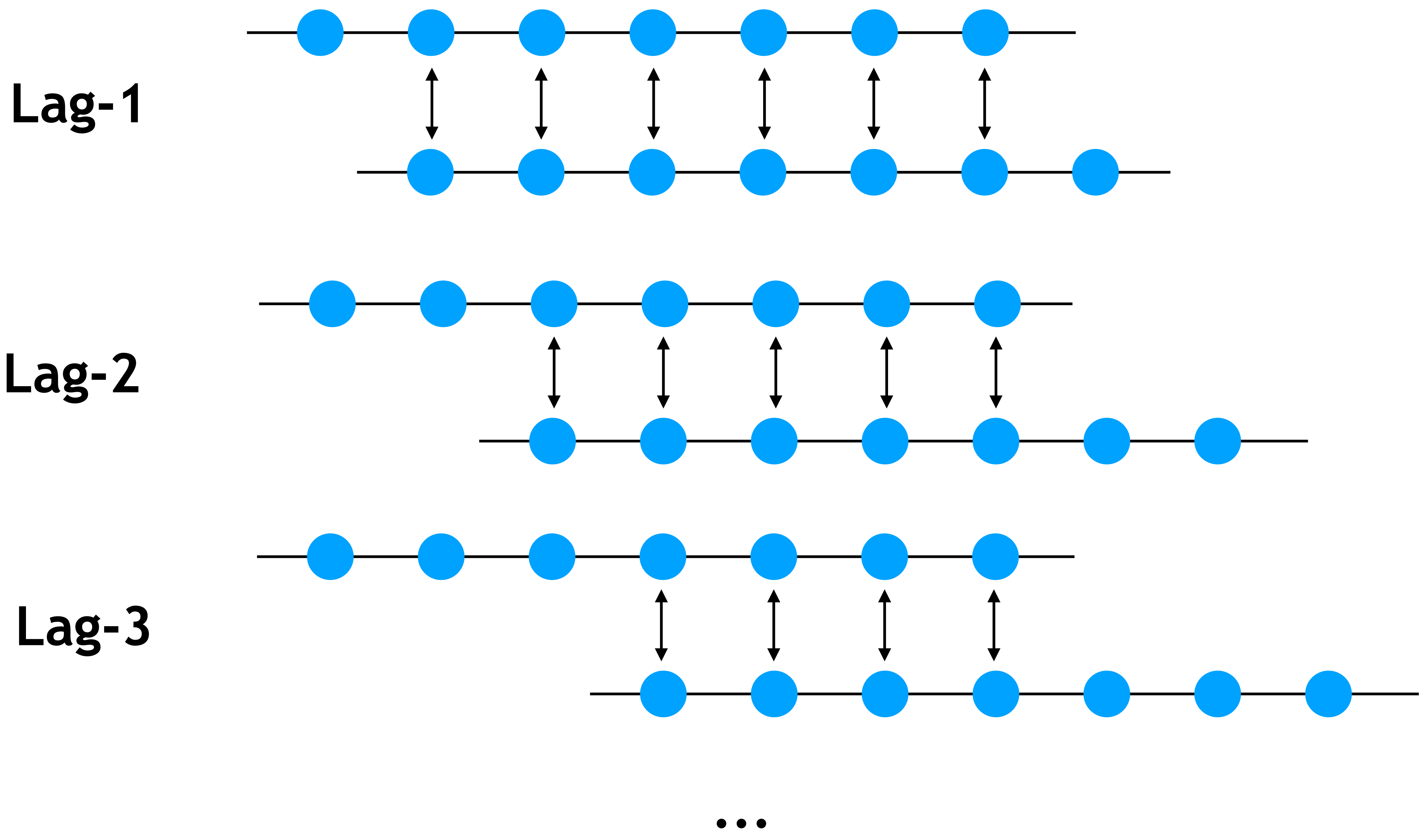


Auto-correlation is the correlation with itself but with a time lag in between

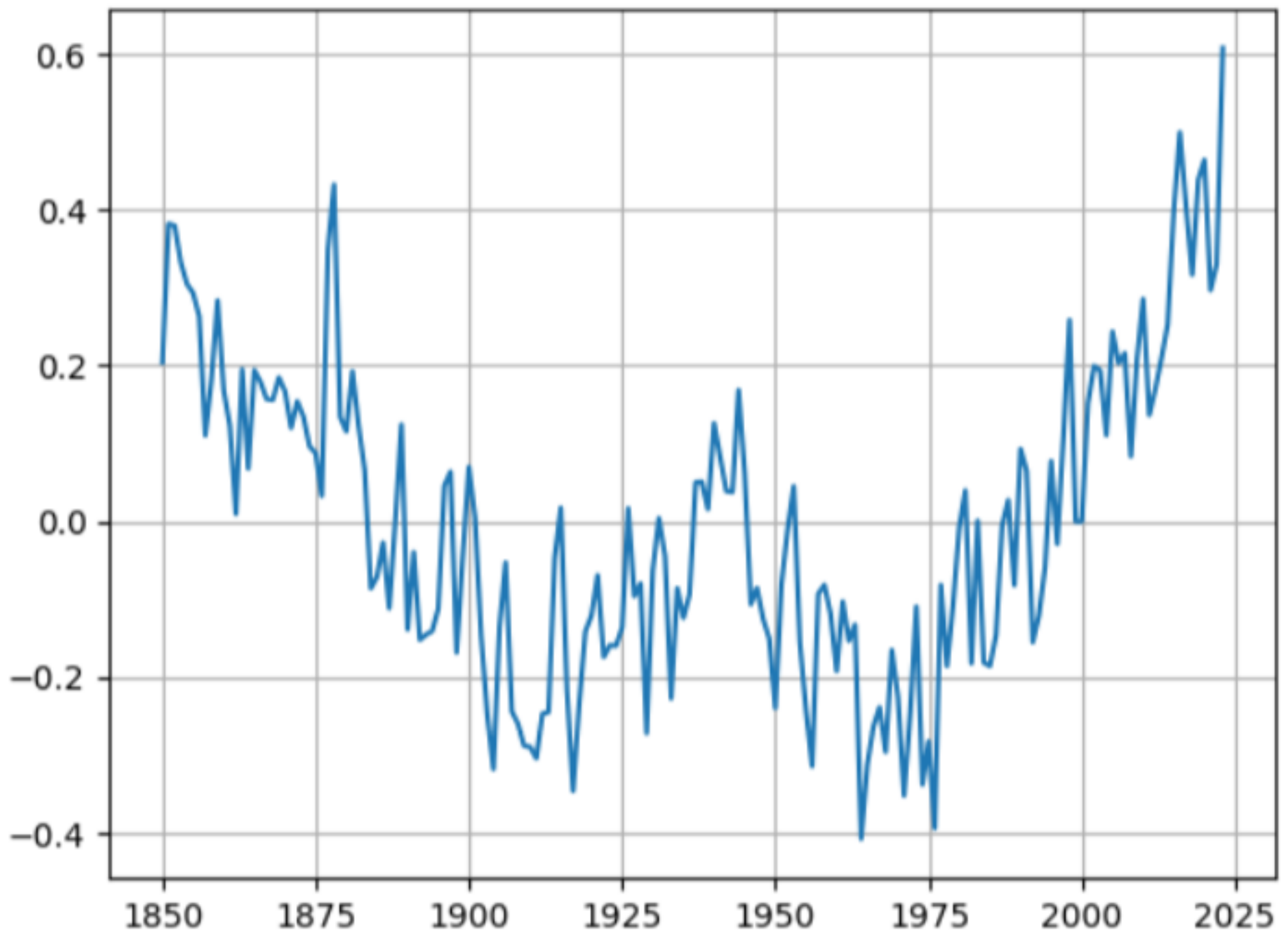


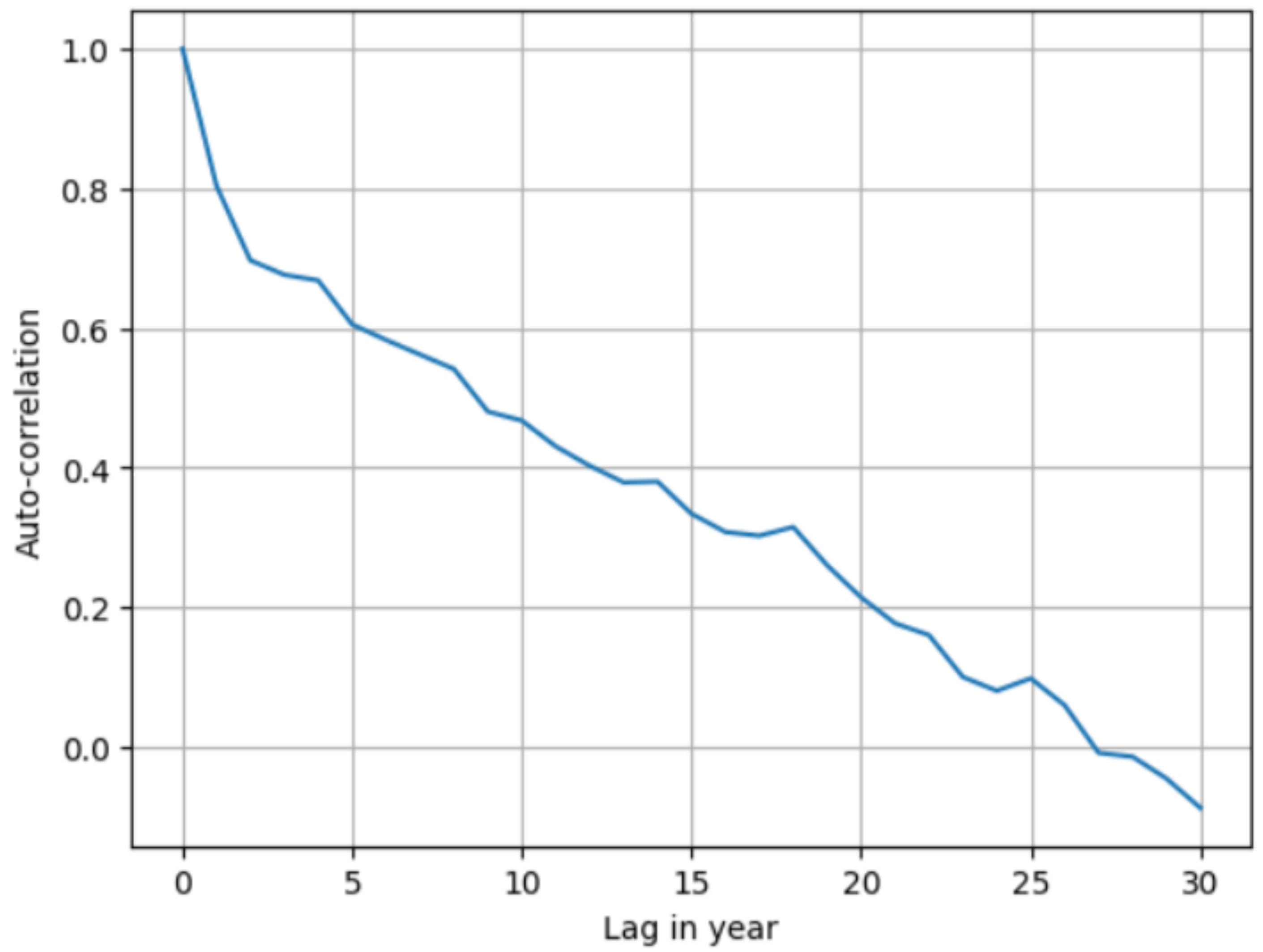


Auto-correlation Function: auto-correlation as a Function of Lag



statsmodels.api.tsa.acf(x, nlags)





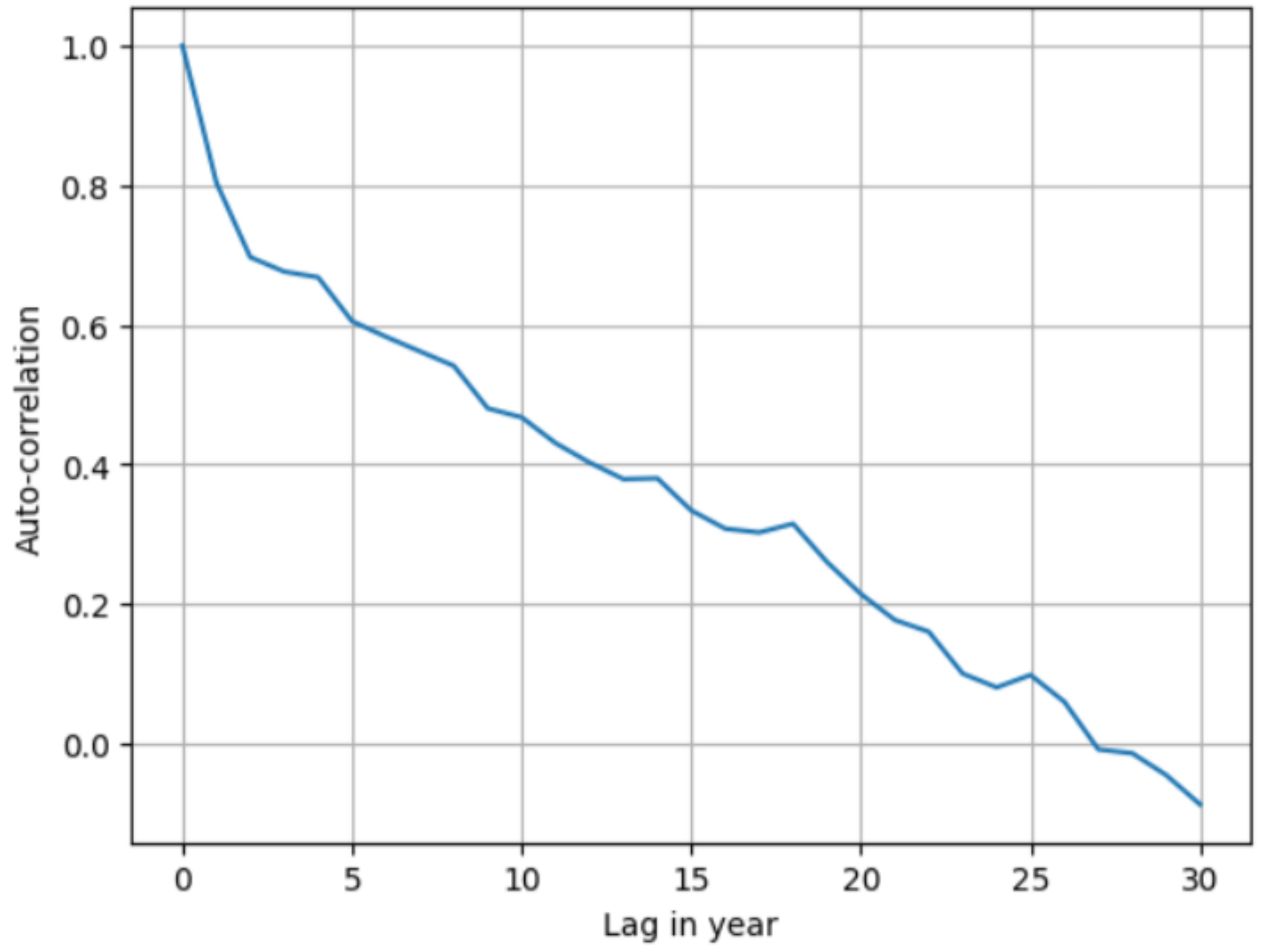
Effective Sample Size (ESS)

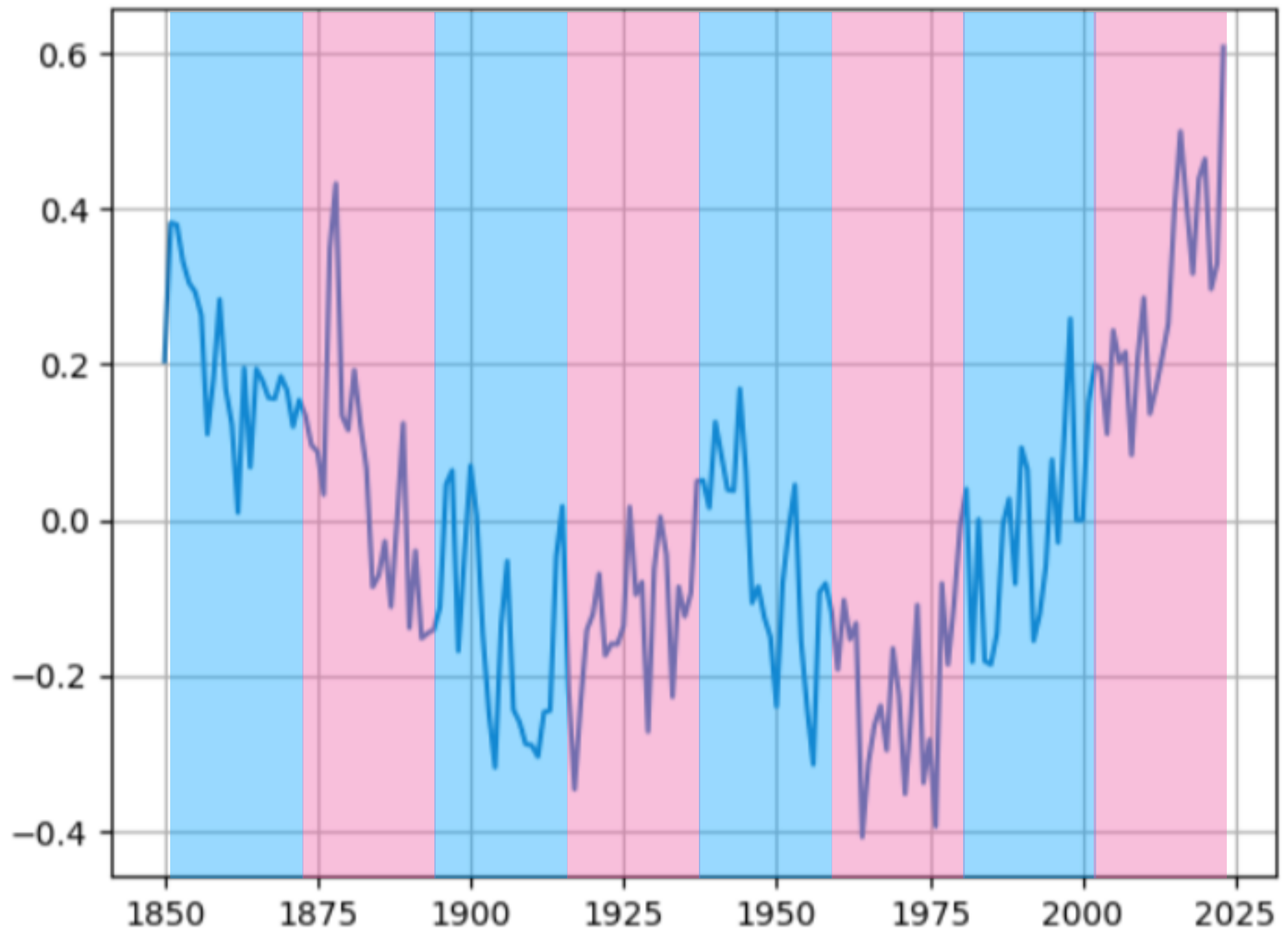
If errors have significant auto-correlation, the effective number of independent sample will be fewer than the number of time step.

This effective sample size can be estimated by:

$$\mathrm{ESS} = \frac{N}{1 + 2\sum_{k=1}^{\infty} \rho_k}$$

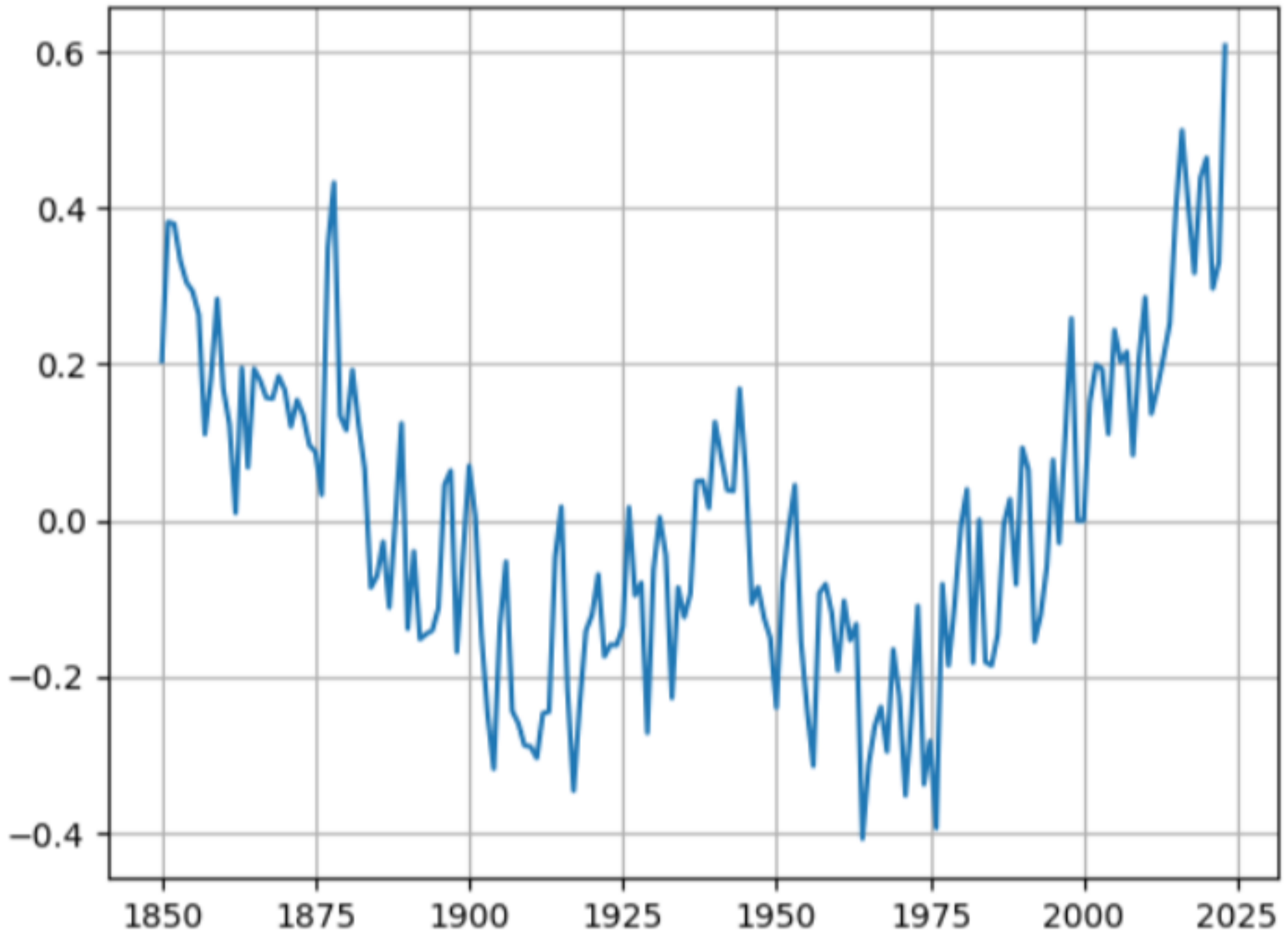
Plugging in estimated autocorrelation for the GMST problem, we get an ESS of 8, suggesting the dataset can be broken into 8 independent chunks of data.





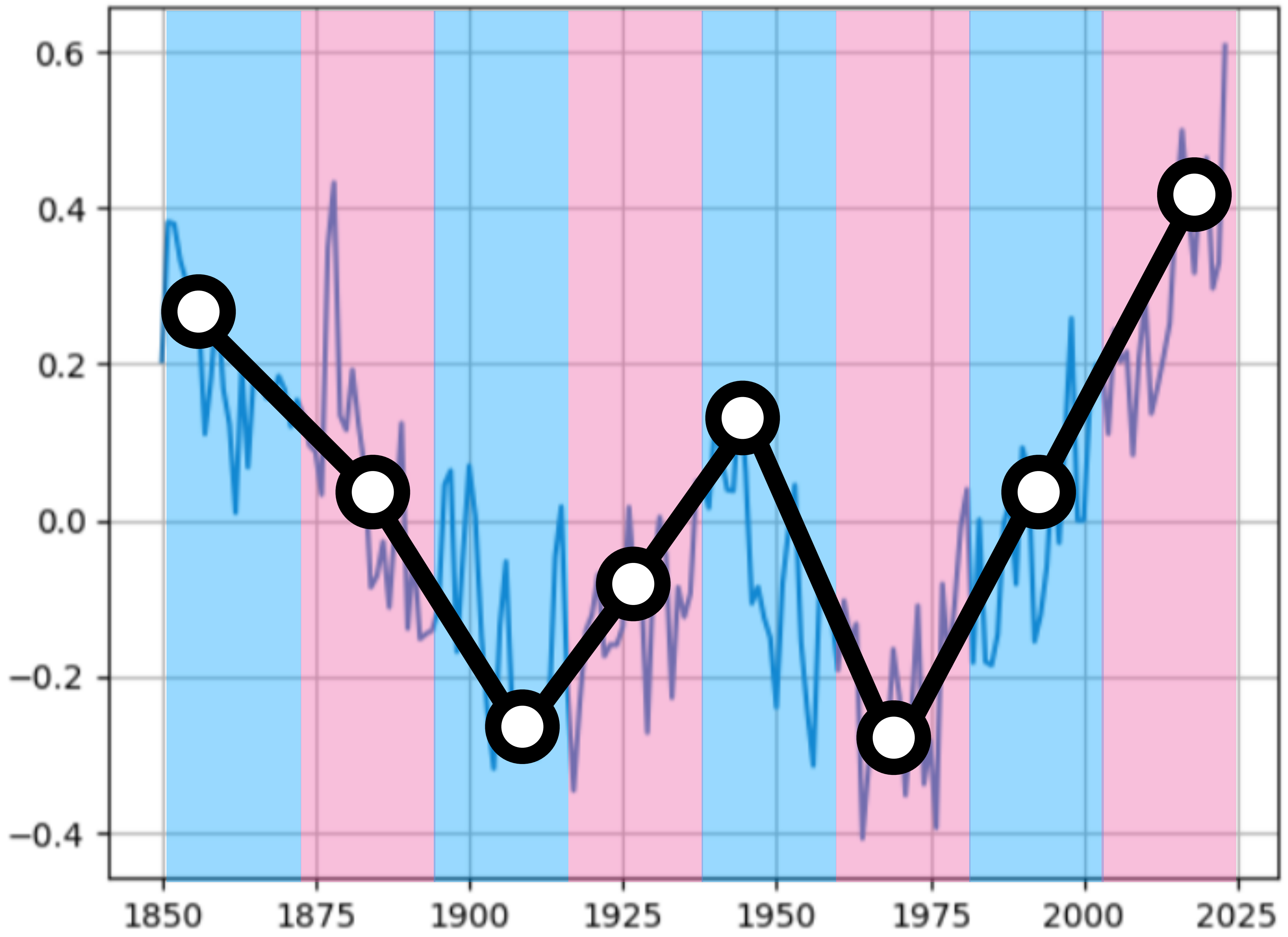
Understanding the Effective sample size

Large auto-correlation

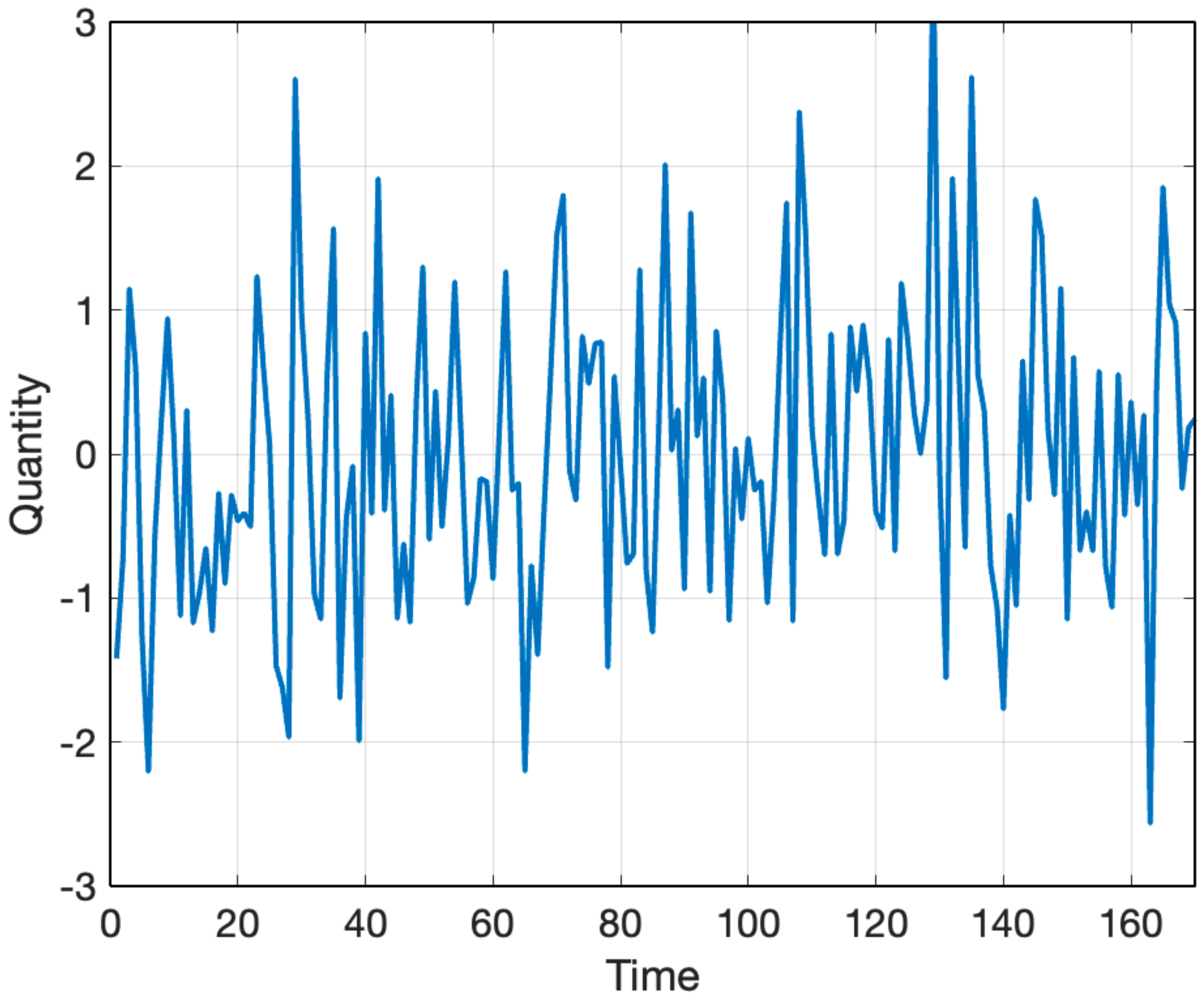


Understanding the Effective sample size

Large auto-correlation



Small auto-correlation



Block Bootstrapping: Further Accounting for Auto-correlation structures

(0) Estimate block size using ESS

(1) Resampling blocks of data with replacement

(2) Calculate the target statistics on resampled data

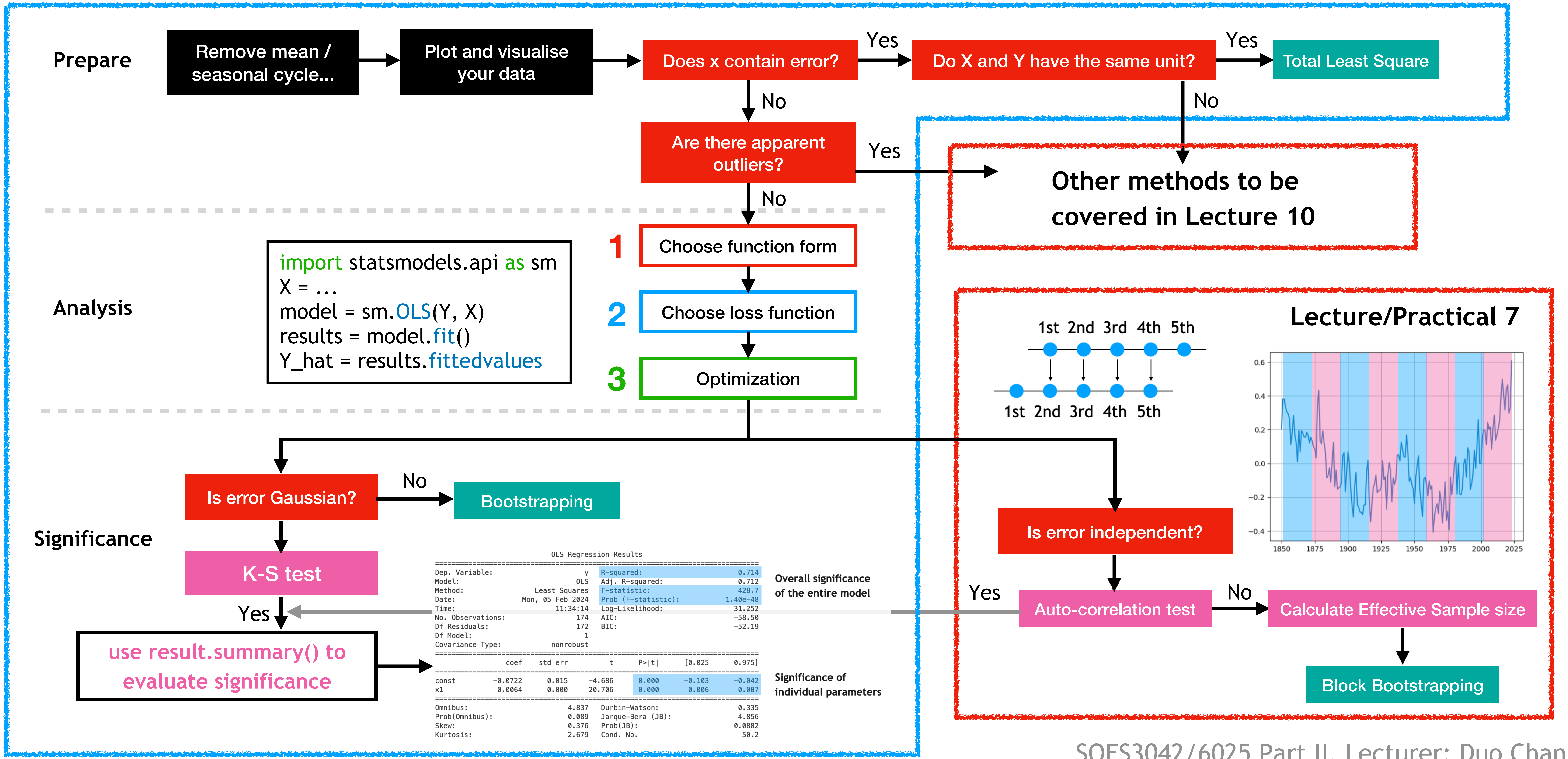
(3) Repeat to generate a distribution

Block bootstrapping often leads to wider uncertainty estimates.

```
ESS = ...

N_blocks = int(N / ESS)

for ct in np.arange(N_boot):

    Resample data blocks with replacement

    Calculate statistics using resampled data

    Save calculated statistics in an array

Evaluate the confidence interval of statistics
```

np.random.randint()

Road Map of the Statistics Part



Road Map of the Statistics Part

	Lecture 13	Lecture 14	Lecture 15
Quantification Technique	Mean, variance, skewness, & kurtosis	Pearson's Correlation (Linear relationship)	Linear regression (OLS)
Uncertainty & Significance	Gaussian distribution Chi-2 distribution	r, p = scipy.stats.pearsonr(x, y)	results.summary( )
Assumptions	Data is Gaussian or follows specific types of distribution Independent Sampling	Data is Gaussian Independent Sampling	x is noise free Error is Gaussian Independent Sampling Equal err variance
Test assumptions	K-S test		Auto-correlation (Effective Sample Size)
Treatment		Bootstrapping	Block Bootstrapping

Data is Gaussian → K-S test → Bootstrapping

Independent Sampling → Auto-correlation (Effective Sample Size) → Block Bootstrapping