RICE CRUMB #2

$$\lim_{N \to \infty} P\left(\| \underline{\hat{\vartheta}}_{ML} - \underline{\vartheta}_0 \| < \varepsilon \right) =$$

$$\lim_{N \to \infty} \mathbb{E}\left[\| \underline{\hat{\vartheta}}_{ML} - \underline{\vartheta}_0 \|^2 \right] =$$

RICE CRUMB #2

$$\lim_{N\to\infty} p\left(\|\hat{\underline{\vartheta}}_{ML} - \underline{\vartheta}_0\| < \varepsilon\right) = 1 \quad \forall \varepsilon$$

CONVERGENCE IN PROBABILITY

$$\lim_{N\to\infty} \mathbb{E}\left[\|\hat{\underline{\vartheta}}_{ML} - \underline{\vartheta}_0\|^2\right] = 0$$

CONVERGENCE IN MEAN SQUARE

(ASYMPTOTICALLY CONSISTENT)

§2.1
Fitting a linear model

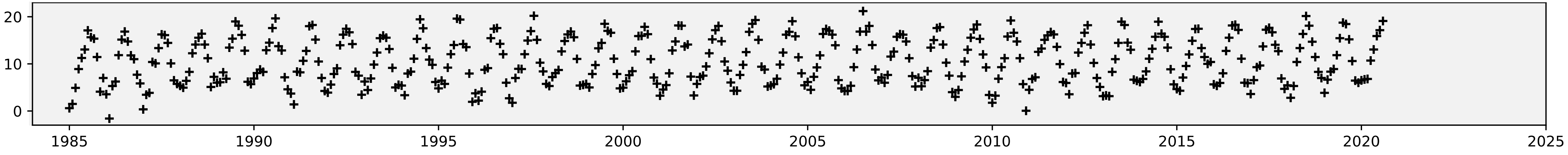
You’ve got to have models in your head. And you’ve got to array your experience – both vicarious and direct – on this latticework of models.

You may have noticed students who just try to remember and pound back what is remembered. Well, they fail in school and in life. You’ve got to hang experience on a latticework of models in your head.

Charlie Munger (business partner of Warren Buffet),
A lesson on elementary, worldly wisdom as it relates to investment management & business.

Monthly average temperatures in Cambridge, UK

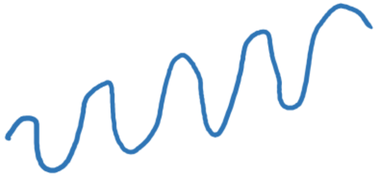
What's a good model for this dataset?



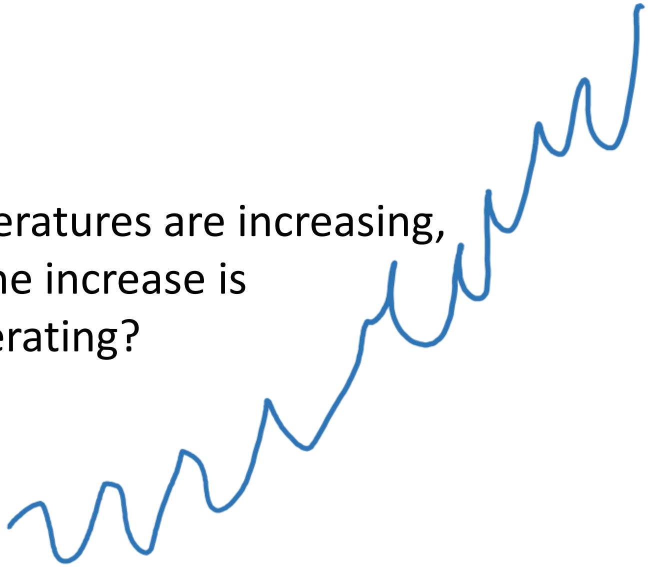
Climate is stable?

$$\text{Temp}(t) \sim a + b\sin\big(2\pi(t+\phi)\big) + N(0,\sigma^2)$$

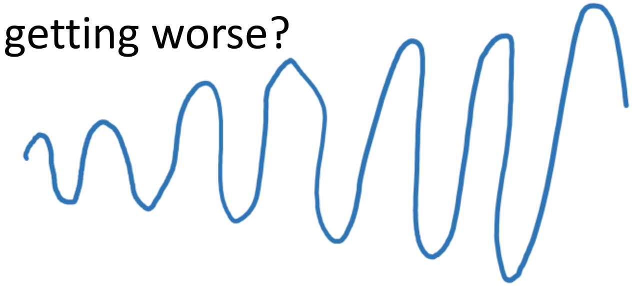
Temperatures are increasing?



Temperatures are increasing, and the increase is accelerating?



The extremes are getting worse?



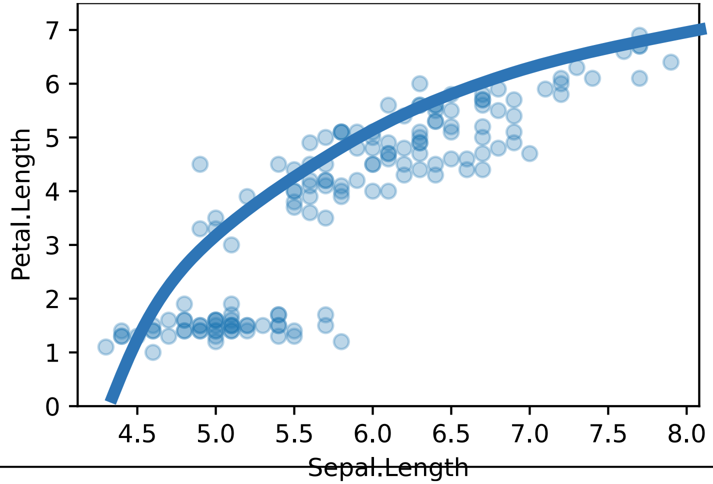
There are so many possible models. We want to make it easy to invent and fit new models, so we have time to explore all the possibilities.

Example 2.1.1

The Iris dataset has 50 records of iris measurements, from three species.

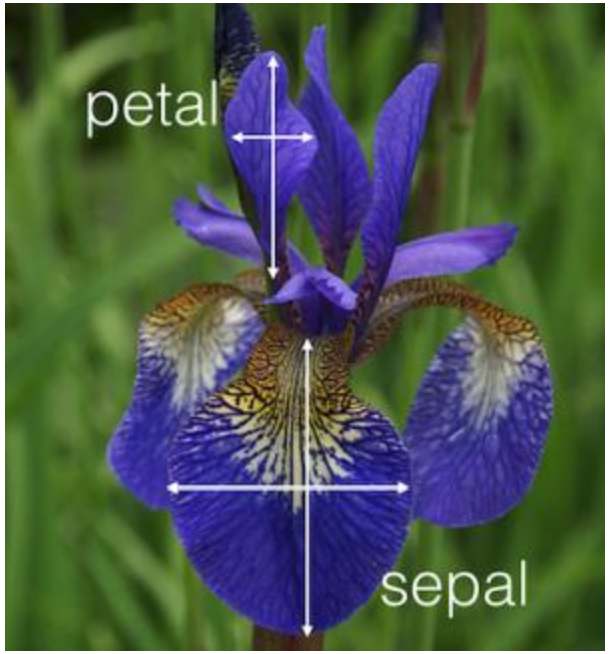
Petal. Length	Petal. Width	Sepal. Length	Sepal. Width	Species
1.0	0.2	4.6	3.6	setosa
5.0	1.9	6.3	2.5	virginica
5.8	1.6	7.2	3.0	virginica
4.2	1.2	5.7	3.0	versicolor
…				

How does `Petal.Length` depend on `Sepal.Length`?



Let's guess that for parameters $\alpha, \beta, \gamma, \sigma$ (to be estimated),

$$\texttt{Petal.Length} \approx \alpha + \beta\ \texttt{Sepal.Length} + \gamma(\texttt{Sepal.Length})^2$$



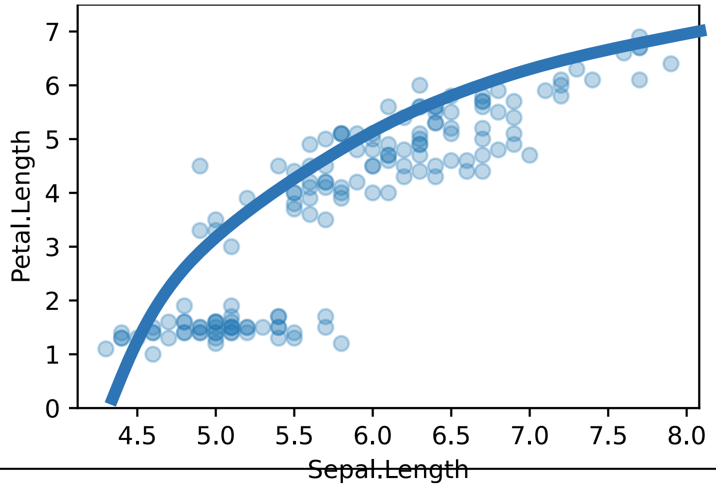
Dataset collected by Edgar Anderson and popularized by Ronald Fisher in 1936

Example 2.1.1

The Iris dataset has 50 records of iris measurements, from three species.

Petal. Length	Petal. Width	Sepal. Length	Sepal. Width	Species
1.0	0.2	4.6	3.6	setosa
5.0	1.9	6.3	2.5	virginica
5.8	1.6	7.2	3.0	virginica
4.2	1.2	5.7	3.0	versicolor
…				

How does `Petal.Length` depend on `Sepal.Length`?



Let's guess that for parameters $\alpha, \beta, \gamma, \sigma$ (to be estimated),

$$\texttt{Petal.Length} \approx \alpha + \beta\ \texttt{Sepal.Length} + \gamma(\texttt{Sepal.Length})^2$$

Linear Model

unknown parameters to be estimated

$$\begin{bmatrix} PL_1 \\ PL_2 \\ \vdots \\ PL_n \end{bmatrix} \approx \alpha \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} + \beta \begin{bmatrix} SL_1 \\ SL_2 \\ \vdots \\ SL_n \end{bmatrix} + \gamma \begin{bmatrix} (SL_1)^2 \\ (SL_2)^2 \\ \vdots \\ (SL_n)^2 \end{bmatrix}$$

response vector

feature vectors

Not a linear model

$$\text{Temp} \approx \alpha + \beta \sin(2\pi(t+\phi)) + \gamma t$$

Dataset collected by Edgar Anderson and popularized by Ronald Fisher in 1936

Let's guess that for parameters $\alpha, \beta, \gamma, \sigma$ (to be estimated),

$$\texttt{Petal.Length} \approx \alpha + \beta\ \texttt{Sepal.Length} + \gamma(\texttt{Sepal.Length})^2$$

Supervised learning

Response & features are numeric

Response is predicted by a linear combination of (known) feature vectors, weighted by unknown parameters.

$$\begin{bmatrix} \mathrm{PL}_1 \\ \mathrm{PL}_2 \\ \vdots \\ \mathrm{PL}_n \end{bmatrix} \approx \alpha \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} + \beta \begin{bmatrix} \mathrm{SL}_1 \\ \mathrm{SL}_2 \\ \vdots \\ \mathrm{SL}_n \end{bmatrix} + \gamma \begin{bmatrix} (\mathrm{SL}_1)^2 \\ (\mathrm{SL}_2)^2 \\ \vdots \\ (\mathrm{SL}_n)^2 \end{bmatrix}$$

Models of this form are called *linear models* (because they're based on linear algebra).

They are flexible, and very fast to optimize.

Least squares estimation

Consider a linear model

$$y \approx \beta_1 e_1 + \cdots + \beta_K e_K$$

“All models are wrong.”

The vector of prediction errors is called the *residual vector*,

$$\varepsilon = y - (\beta_1 e_1 + \cdots + \beta_K e_K)$$

We can fit the model using *least squares estimation.* This means finding parameters $\beta_1, \ldots, \beta_K$ to minimize the mean square error

$$\mathrm{mse} = \frac{1}{n} \sum_{i=1}^{n} \varepsilon_i^2$$

```
iris = pandas.read_csv(...)
```

$$\texttt{Petal.Length} \approx \alpha + \beta\ \texttt{Sepal.Length} + \gamma(\texttt{Sepal.Length})^2$$

$$\begin{bmatrix} \mathrm{PL}_1 \\ \mathrm{PL}_2 \\ \vdots \\ \mathrm{PL}_n \end{bmatrix} \approx \alpha \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} + \beta \begin{bmatrix} \mathrm{SL}_1 \\ \mathrm{SL}_2 \\ \vdots \\ \mathrm{SL}_n \end{bmatrix} + \gamma \begin{bmatrix} (\mathrm{SL}_1)^2 \\ (\mathrm{SL}_2)^2 \\ \vdots \\ (\mathrm{SL}_n)^2 \end{bmatrix}$$

Fitting the model

```
one, SL, PL = np.ones(len(iris)), iris['Sepal.Length'], iris['Petal.Length']
model = sklearn.linear_model.LinearRegression(fit_intercept=False)
model.fit(np.column_stack([one, SL, SL**2]), PL)
(α,β,γ) = model.coef_
```

Making predictions / getting fitted values from the model

```
newSL = np.linspace(4.2, 8.2, 20)
predPL = α + β*newSL + γ*(newSL**2)
```

```
iris = pandas.read_csv(...)
```

$$\texttt{Petal.Length} \approx \alpha + \beta\ \texttt{Sepal.Length} + \gamma(\texttt{Sepal.Length})^2$$

$$\begin{bmatrix} \mathrm{PL}_1 \\ \mathrm{PL}_2 \\ \vdots \\ \mathrm{PL}_n \end{bmatrix} \approx \alpha \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} + \beta \begin{bmatrix} \mathrm{SL}_1 \\ \mathrm{SL}_2 \\ \vdots \\ \mathrm{SL}_n \end{bmatrix} + \gamma \begin{bmatrix} (\mathrm{SL}_1)^2 \\ (\mathrm{SL}_2)^2 \\ \vdots \\ (\mathrm{SL}_n)^2 \end{bmatrix}$$

sklearn.linear_model puts in the $\begin{bmatrix} \vdots \end{bmatrix}$ feature for us by default. If we don't want it we have to specify fit_intercept=False.

Fitting the model (cleaner code)

```
SL, PL = iris['Sepal.Length'], iris['Petal.Length']
model = sklearn.linear_model.LinearRegression()
model.fit(np.column_stack([SL, SL**2]), PL)
α,(β,γ) = model2.intercept_, model2.coef_
```

This saves us from having to explicitly code up the prediction formula — so there's less chance we introduce bugs.

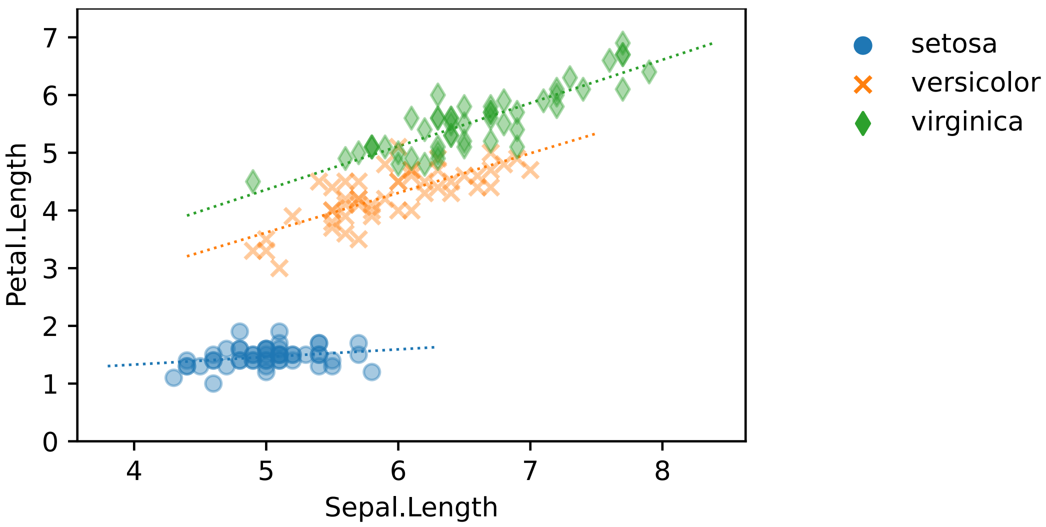
Making predictions / getting fitted values from the model (cleaner code)

```
newSL = np.linspace(4.2, 8.2, 20)
predPL = model.predict(np.column_stack([newSL, newSL**2]))
```

§2.2 Feature design

How do we design features, so that linear models answer the questions we want answered?

ONE-HOT CODING



$PL \approx \alpha_{species} + \beta_{species}\, SL$

This model has 6 unknown parameters:

$\alpha_{set} \quad \alpha_{vers} \quad \alpha_{virg}$

$\beta_{set} \quad \beta_{vers} \quad \beta_{virg}.$

So my model has 6 feature vectors.

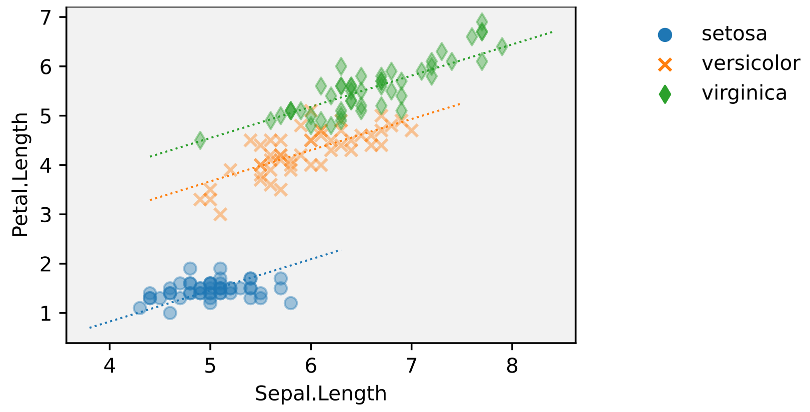
Row 1: setosa: $PL \approx \alpha_{setosa} + \beta_{setosa}\, SL.$

$$\begin{array}{r} * \text{ seto} \\ \text{virg} \\ \text{virg} \\ \text{seto} \\ \text{vers} \\ \\ \end{array} \begin{bmatrix} PL_1 \\ PL_2 \\ PL_3 \\ PL_4 \\ PL_5 \\ \vdots \end{bmatrix} \approx \alpha_{\text{seto}} \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \end{bmatrix} + \alpha_{\text{virg}} \begin{bmatrix} 0 \\ 1 \\ 1 \\ \vdots \end{bmatrix} + \alpha_{\text{vers}} \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \end{bmatrix} + \beta_{\text{seto}} \begin{bmatrix} SL_1 \\ 0 \\ 0 \\ \vdots \end{bmatrix} + \beta_{\text{virg}} \begin{bmatrix} 0 \\ SL_2 \\ SL_3 \\ \vdots \end{bmatrix} + \beta_{\text{vers}} \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \end{bmatrix}$$

$$\vec{PL} \approx \alpha_{seto}\, 1_{\vec{spec}=setosa} + \alpha_{virg}\, 1_{\vec{spec}=virg} + \alpha_{vers}\, 1_{\vec{spec}=vers} + \beta_{seto}\,(\vec{SL} \times 1_{\vec{spec}=seto}) + \cdots$$

EXERCISE

Fit the model with three parallel straight lines.

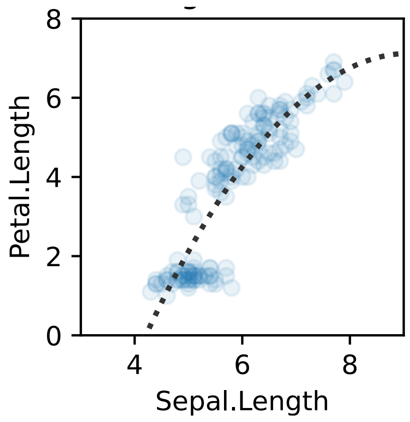


$$PL \approx \alpha_{species} + \beta\, SL$$

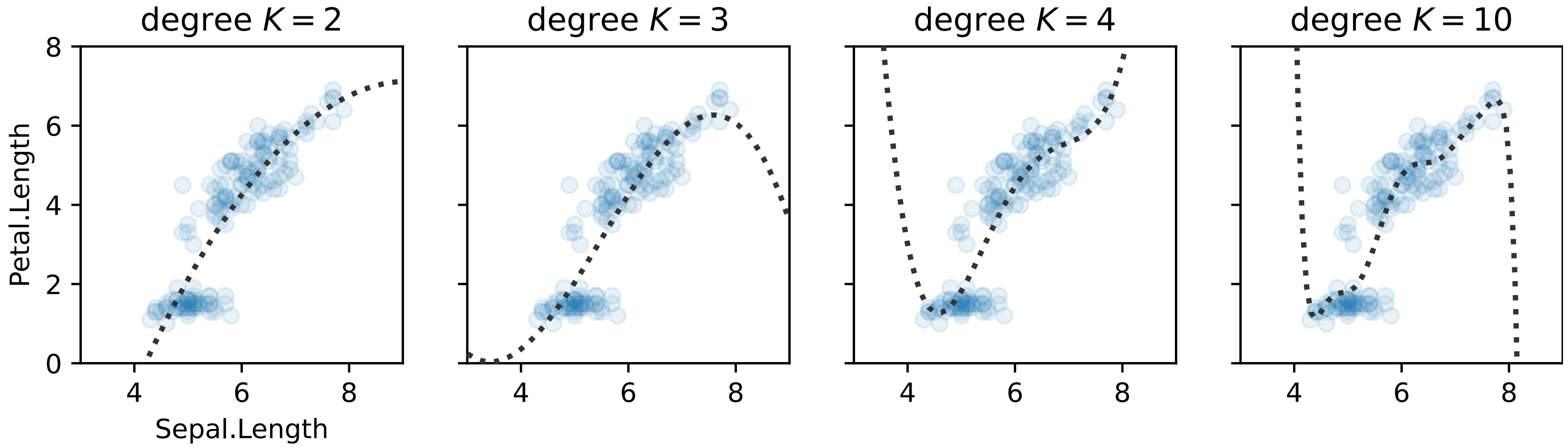
$$\approx \alpha_{set}\, 1_{\vec{spec} = set} + \cdots$$

$$+ \beta\, \vec{SL}$$

NON-LINEAR RESPONSE



$\texttt{Petal.Length} \approx$
$$\alpha + \beta\ \texttt{Sepal.Length} + \gamma(\texttt{Sepal.Length})^2$$



$\texttt{Petal.Length} \approx$
$$\beta_0 + \sum_{k=1}^{K} \beta_k (\texttt{Sepal.Length})^k$$

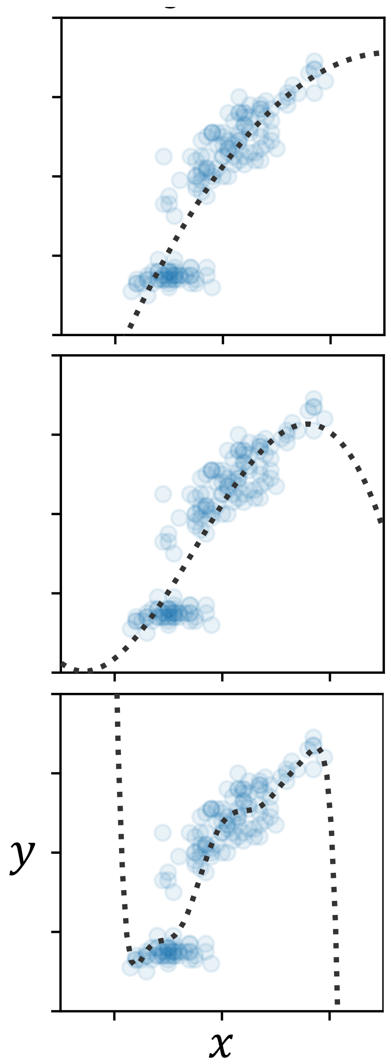
Q. Should we just keep adding more and more features to our model?

(seeing as the more features we add, the better we can fit the dataset)

A. No. If we did, we'd *overfit*.

Only add in features that you (as a scientist) believe are relevant.

Or do model testing, e.g. evaluation on a holdout set.
[§5–7]

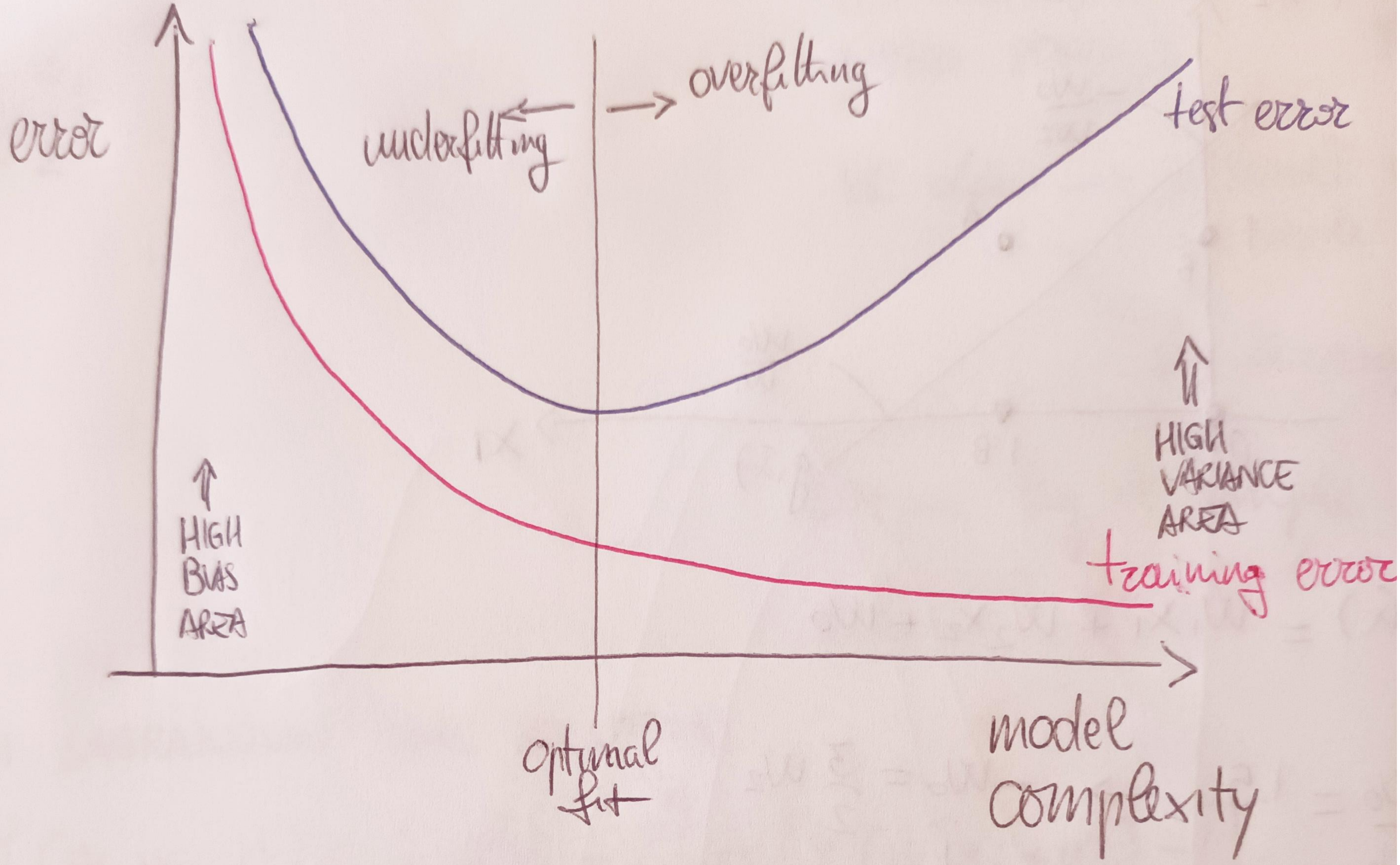


quadratic

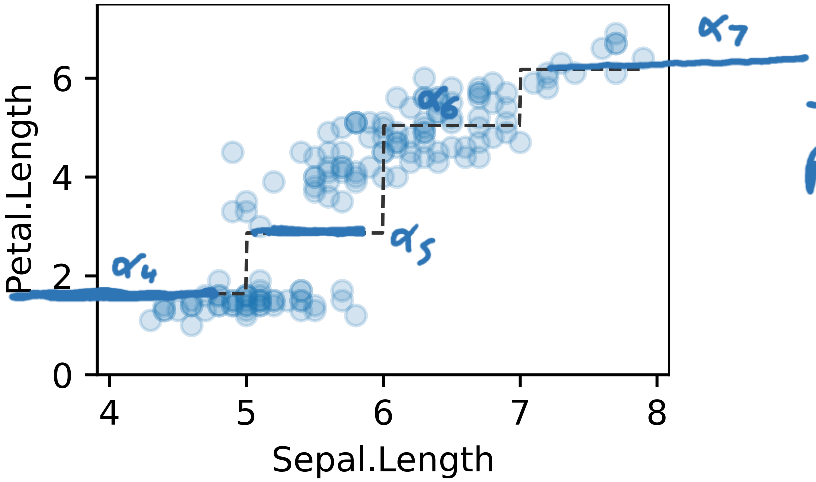
cubic

polynomial
degree 10

$y \approx \beta_0 + \beta_1 x + \cdots + \beta_{10} x^{10}$


error
underfitting
overfitting
test error
HIGH VARIANCE AREA
training error
HIGH BIAS AREA
optimal fit
model complexity

NON-LINEAR RESPONSE via one-hot coding



$$\vec{PL} \approx \alpha_4 \mathbb{1}_{\vec{SL}<5} + \alpha_5 \mathbb{1}_{\lfloor \vec{SL} \rfloor = 5} + \alpha_6 \mathbb{1}_{\lfloor \vec{SL} \rfloor = 6} + \alpha_7 \mathbb{1}_{\vec{SL} \geq 7}$$

e.g. for an observation with $SL = 5.3$ we predict α_5

$SL = 2.7$ α_4

Memory lane

Credits: Smithsonian Institution

Listening modes:

- Rock
- Jazz
- Classical

How were listening modes calculated?

- Take a bunch of songs N_S of three different music genres
- In a linear model
  - the amplitudes for each frequency would be the feature vectors;
  - the amplitude of the bass at the speaker would be the response vector
  - (dimensionality = N_S)
- Calculate for each frequency the corresponding parameter

Scope of listening modes: improving audio quality

- By applying a frequency filter in the shape of the estimated distribution
  - Fostering some frequencies, reducing other frequencies

(

+

)

X

Accurate listening mode

=

(

+

)

X

Wrong listening mode

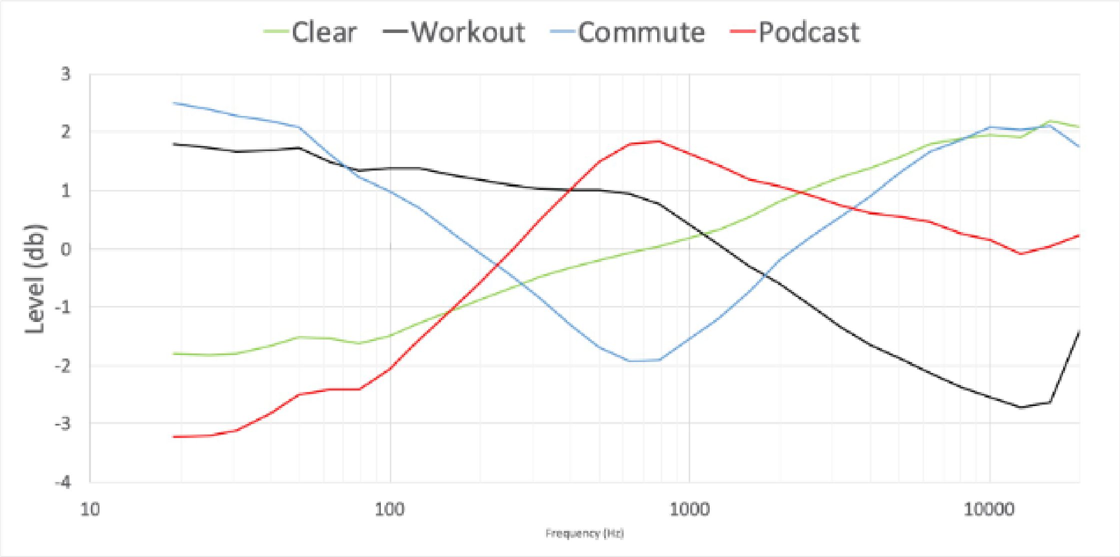
=

Listening modes, today

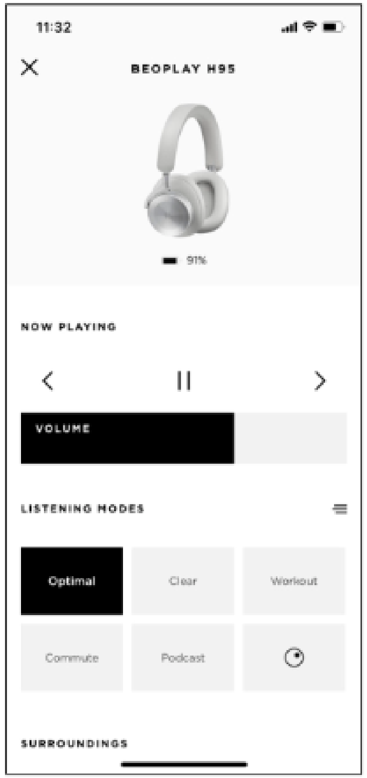
Listening modes, today

Listening Modes

- These are the differences in frequency response relative to **Optimal Mode** (shown previous slide)
- **Clear Mode** is a upward sloped filter centered at 1kHz boosting treble above and cutting below
- **Workout Mode** does a similar EQ but sloped downwards
- **Commute Mode** boosts bass and treble
- **Podcast Mode** cuts bass below 250 Hz and boosts above that

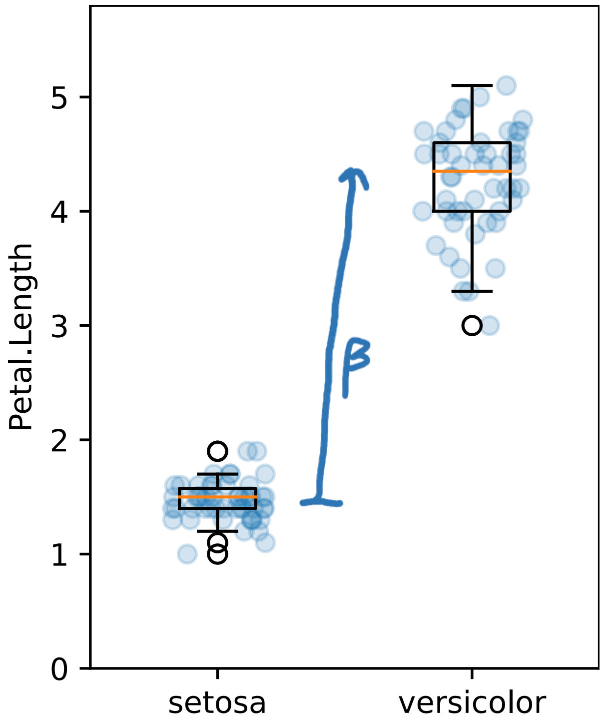


Note: these curves are based on acoustical measurements of the left channel



Credits: Sean Olive

COMPARING GROUPS



Measurements for condition A: $a = [a_1, a_2, \ldots, a_m]$
Measurements for condition B: $b = [b_1, b_2, \ldots, b_n]$

Can we use a linear model to compare A and B?

cond	x
A	a_1
A	⋮
⋮	⋮
A	a_m
B	b_1
B	b_2
⋮	⋮
B	b_n

$$\vec{x} \approx \alpha_A 1_{\vec{cond}=A} + \alpha_B 1_{\vec{cond}=B}$$

Or

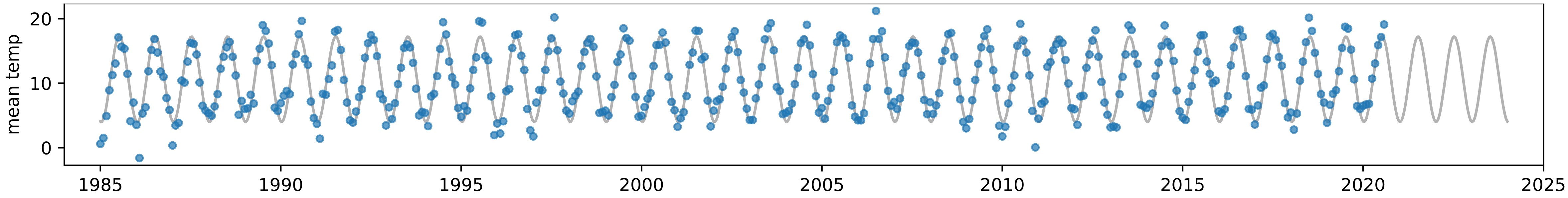
$$\vec{x} = \alpha + \beta 1_{\vec{cond}=B}.$$

For a person of type A, $x \approx \alpha$

B, $x \approx \alpha + \beta$

β measures the difference between the two groups.

PERIODIC PATTERN



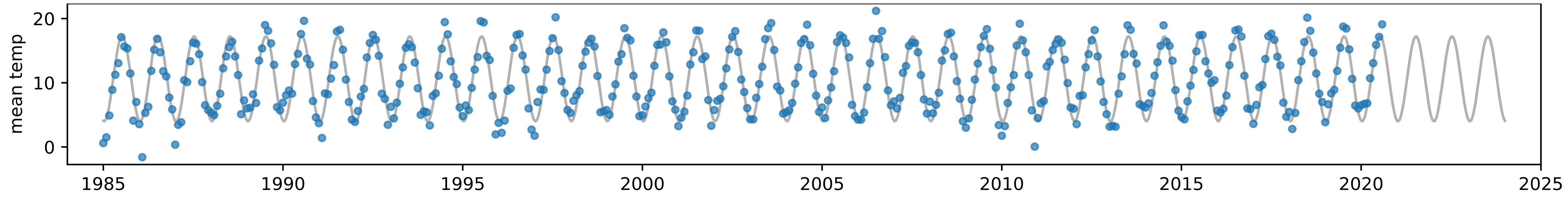
We'd like to fit the model:

$$\texttt{temp} \approx \alpha + \beta \sin(2\pi \texttt{t} + \varphi)$$

It looks like we can't use `sklearn.LinearRegression`. That's only for linear models, e.g.

$$\texttt{temp} \approx \alpha + \beta \texttt{e} + \gamma \texttt{f}$$

PERIODIC PATTERN



We'd like to fit the model:

$$\text{temp} \approx \alpha + \beta \sin(2\pi t + \varphi)$$

$$\approx \alpha + \beta\,\{\sin(2\pi t)\cos\phi \;+\; \cos(2\pi t)\sin\phi\}$$

$$= \alpha + (\beta\cos\phi)\sin(2\pi t) + (\beta\sin\phi)\cos(2\pi t)$$

$$= \alpha + \beta_1 \sin(2\pi t) + \beta_2 \cos(2\pi t)$$

$$= \alpha + \beta_1 e + \beta_2 f$$

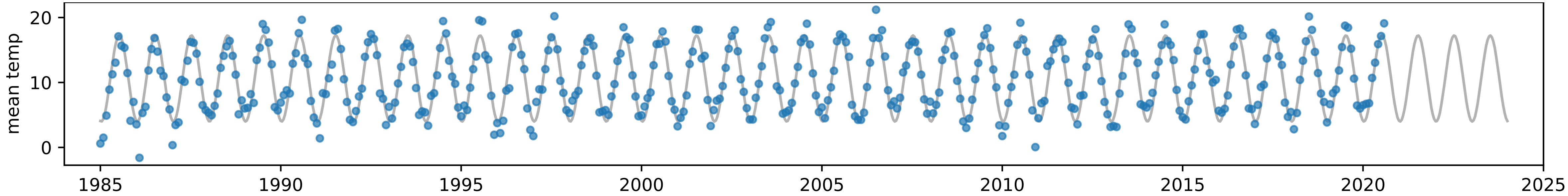
From school trigonometry,

$$\sin(A+B) = \sin(A)\cos(B) + \cos(A)\sin(B)$$

a linear model with feature vectors 1, $\sin(2\pi t)$, $\cos(2\pi t)$

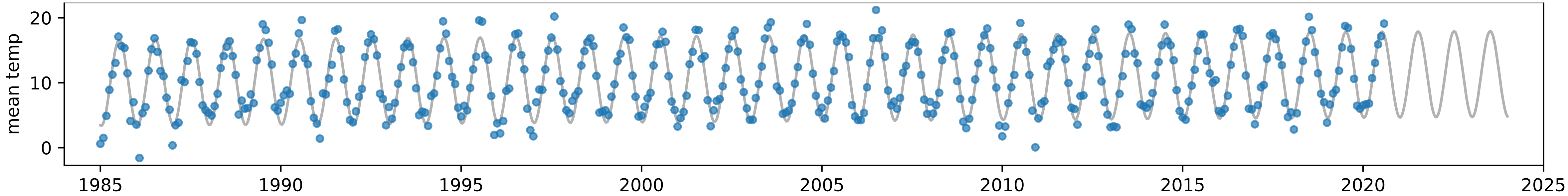
PERIODIC PATTERN

$$\texttt{temp} \approx \alpha + \beta_1 \sin(2\pi \texttt{t}) + \beta_2 \cos(2\pi \texttt{t})$$



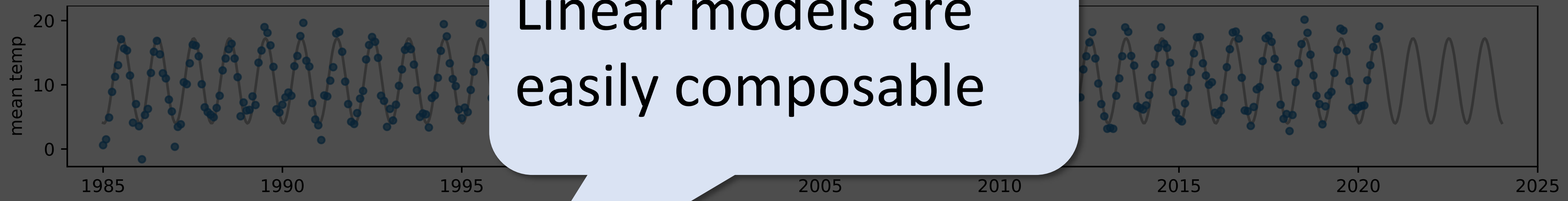
PERIODIC PATTERN + SECULAR TREND

$$\texttt{temp} \approx \alpha + \beta_1 \sin(2\pi \texttt{t}) + \beta_2 \cos(2\pi \texttt{t}) + \gamma \texttt{t}$$



PERIODIC PATTERN

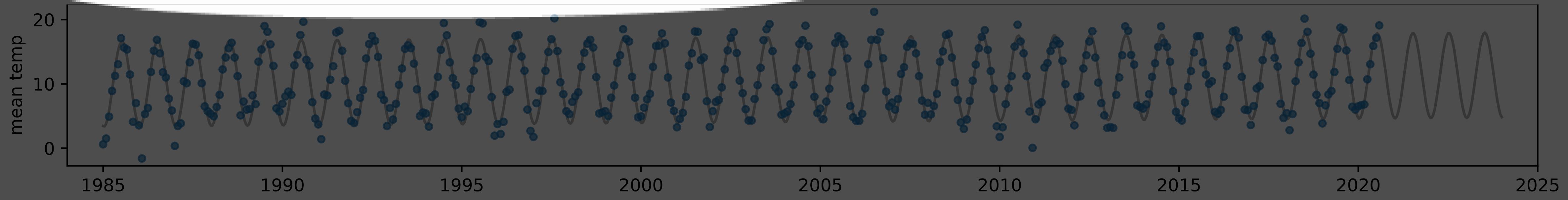
$\texttt{temp} \approx \alpha + \beta_1 \sin(2\pi \texttt{t}) + \beta$



Linear models are easily composable

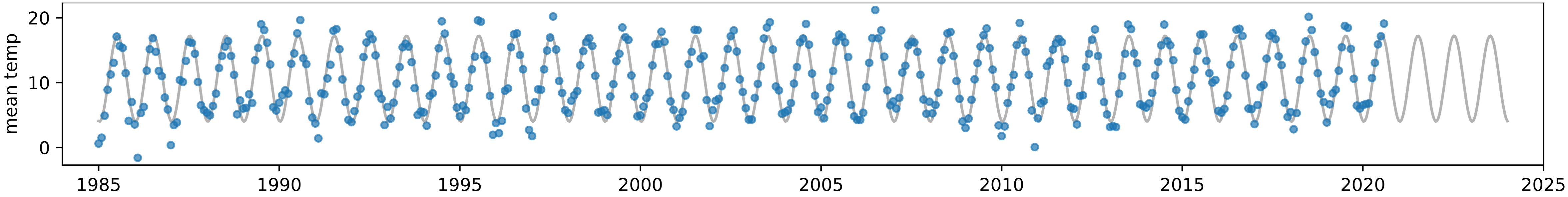
PERIODIC PATTERN + SECULAR TREND

$\texttt{temp} \approx \alpha + \beta_1 \sin(2\pi \texttt{t}) + \beta_2 \cos(2\pi \texttt{t}) + \gamma \texttt{t}$



With our periodic model …

$$\texttt{temp} \approx \alpha + \beta_1 \sin(2\pi \texttt{t}) + \beta_2 \cos(2\pi \texttt{t})$$

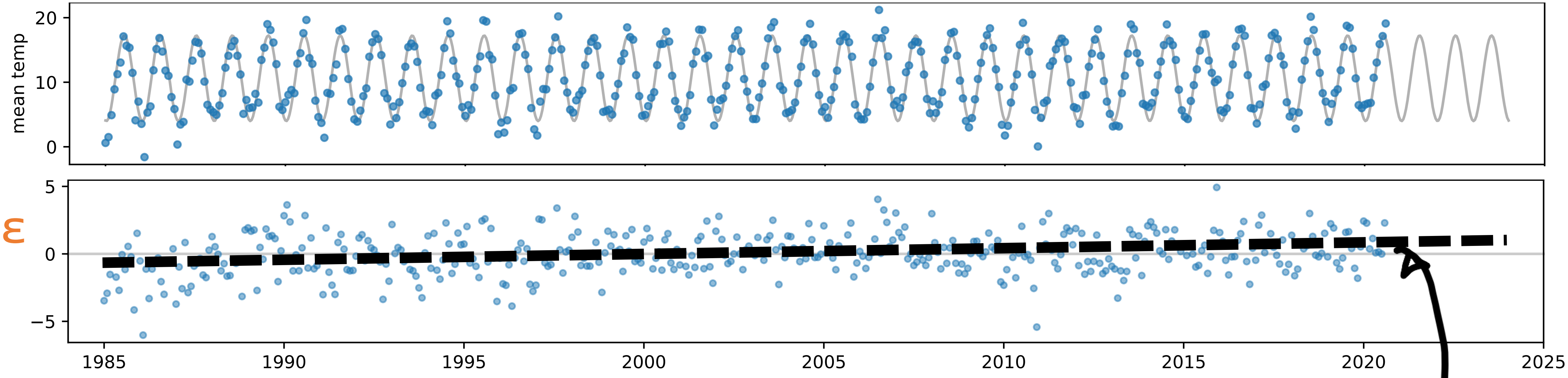


… how do we *discover* we should add a secular term $+\gamma\texttt{t}$?

If we hadn't thought to include climate change in our temperature model ...

$$\texttt{temp} \approx \alpha + \beta \sin\big(2\pi(\texttt{t} + \phi)\big)$$

$$\texttt{temp} = \alpha + \beta \sin\big(2\pi(\texttt{t} + \phi)\big) + \varepsilon$$



$\varepsilon \approx \delta + \gamma t$

This suggests a revised model ...

$$\texttt{temp} = \alpha' + \beta' \sin\big(2\pi(\texttt{t} + \phi)\big) + \gamma \texttt{t} + \varepsilon$$

Diagnosing a linear model

After fitting a model

```
model.fit(..., y)
```

1. Compute the residuals

```
ε = y - model.predict()
```

2. Plot ε every way we can think of. If there's a systematic pattern, add a feature that describes it.