This week we're building up our skills at inventing useful probability models.

MONDAY

❖ linear models & feature design

(to quickly turn our ideas into easy-to-fit models)

WEDNESDAY

❖ "debugging" models

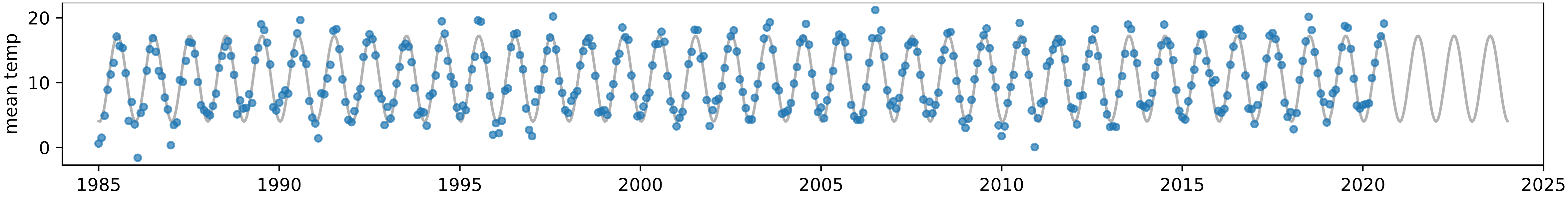
FRIDAY

❖ identifiability of parameters

❖ the link between least squares and likelihood

With our periodic model ...

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



... how do we *discover* we should add a secular term, such as $+\gamma\texttt{t}$?

Machine learning models don’t fail with nice friendly exceptions. They fail by giving us fishy answers.

So what does “debugging” look like?

§2.3. Diagnosing a linear model

After fitting a model

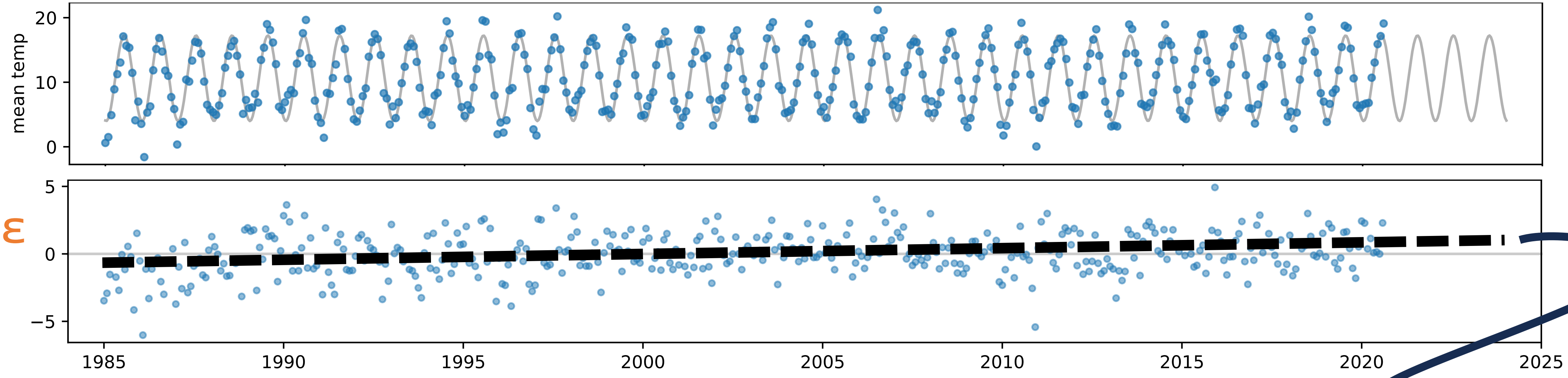
```
model.fit(..., y)
```

1. Compute the prediction errors a.k.a. the residuals

   ```
   ε = y – model.predict()
   ```

2. Plot ε against anything we can think of. It tells us where our model is poor.

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



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

This suggests a revised model …

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

α'

Example sheet 1

Question 8. For the climate data from section 2.2.5 of lecture notes, we proposed the model

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

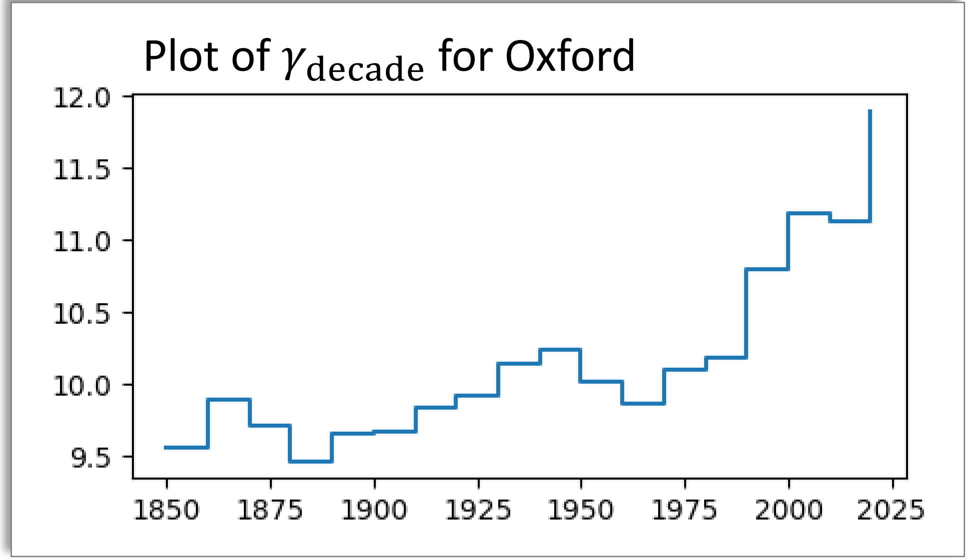
in which the $+\gamma\texttt{t}$ term asserts that temperatures are increasing at a constant rate. We might suspect though that temperatures are increasing non-linearly. To test this, we can create a non-numerical feature out of `t` by

```
u = 'decade_' + str(math.floor(t/10)) + '0s'
```

(which gives us values like `'decade_1980s'`, `'decade_1990s'`, etc.) and fit the model

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

Write this as a linear model, and give code to fit it. *[Note. You should explain what your feature vectors are, then give a one-line command to estimate the parameters.]*



QUESTION. This code doesn't pass the Moodle tester. What's the bug?

should be

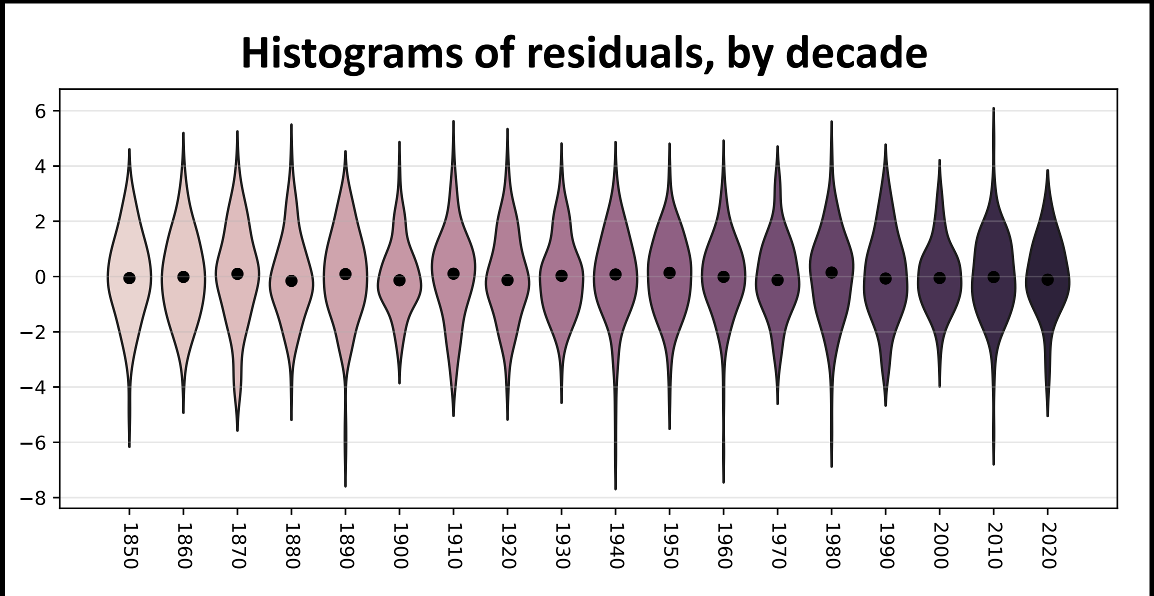
$2\pi t$

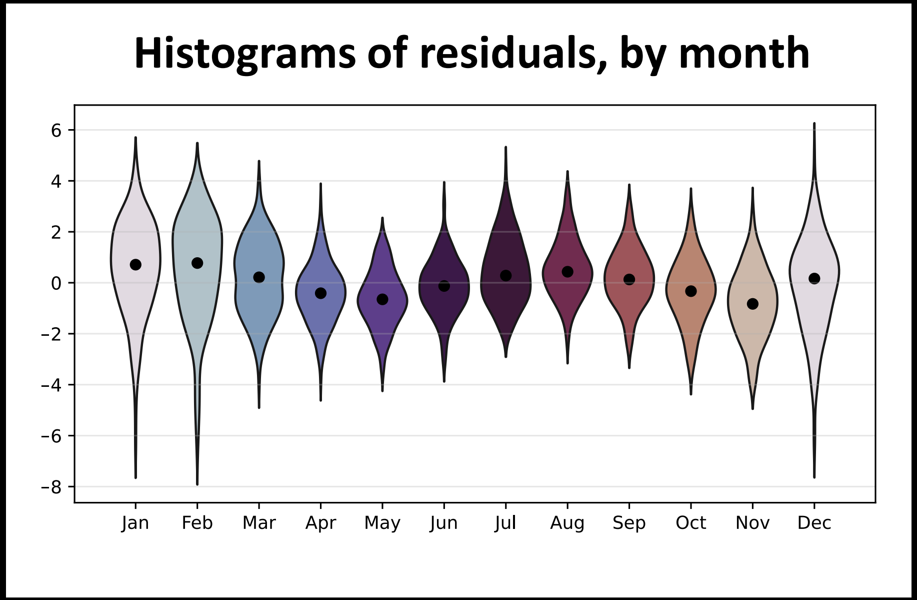
```
class StepPeriodicModel():
    def __init__(self):
        self.mindec = np.nan
        self.maxdec = np.nan

    def fit(self, t, temp):
        self.mindec = np.floor(min(t)/10)*10
        self.maxdec = np.floor(max(t) / 10) * 10
        indicators = [np.where(np.floor(t/10)*10 == year*10 + self.mindec, 1, 0)
                      for year in range(int((self.maxdec - self.mindec)/10) + 1)]
        X = np.column_stack([np.sin(2 * np.pi * np.mod(t,1)), np.cos(2 * np.mod(t,1)), *indicators])
        model = sklearn.linear_model.LinearRegression(fit_intercept=False)
        model.fit(X, temp)
        (_,_,*γ) = model.coef_
        self.γ = np.append(γ, np.nan)

    def predict_step(self, t):
        t = np.array(t).astype(float)
        ℓ = ((np.floor(t/10)*10-self.mindec)/10).astype(int)
        replace_mask = np.where((ℓ<0) | (ℓ>=len(self.γ)-1))
        ℓ[replace_mask] = len(self.γ) - 1
        return np.take(self.γ, ℓ)
```

I computed the residuals.
I plotted them against the variables in the question, looking for any systematic pattern.





This tells me there's something dodgy with how the model captures month-by-month variation, i.e. with the features for the sinusoid shape.

When we propose a model, write down the predicted response at some “useful” datapoints.

This should help us interpret what the parameters mean.

After fitting a model

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

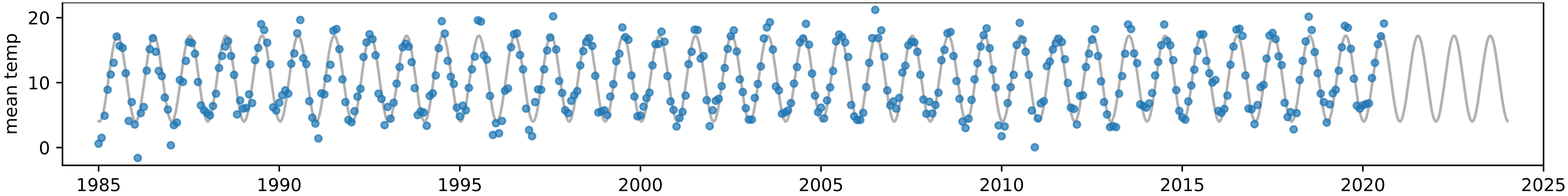
1. Compute the prediction errors a.k.a. the residuals

```
ε = y - model.predict()
```

2. Plot ε against anything we can think of. It tells us where our model is poor.

When we fit the model with a constant-rate climate change term ...

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



the fitted parameters are surprising:

α	β_1	β_2	γ
-67.5	-1.1	-6.6	0.039

Why do these three models lead to such different estimates for α?

Model 0: $\texttt{temp} \approx \alpha\, 1 + \beta_1 \sin(2\pi\, \texttt{t}) + \beta_2 \cos(2\pi\, \texttt{t})$ $\Rightarrow$ $\hat{\alpha} = 10.6\ °\mathrm{C}$

Model A: $\texttt{temp} \approx \alpha\, 1 + \beta_1 \sin(2\pi\, \texttt{t}) + \beta_2 \cos(2\pi\, \texttt{t}) + \gamma\, \texttt{t}$ $\Rightarrow$ $\hat{\alpha} = -67.5\ °\mathrm{C}$

Model B: $\texttt{temp} \approx \underline{\alpha}\, 1 + \beta_1 \sin(2\pi\, \texttt{t}) + \underline{\beta_2} \cos(2\pi\, \texttt{t}) + \gamma\, (\texttt{t}-2000)$ $\Rightarrow$ $\hat{\alpha} = 10.5\ °\mathrm{C}$

EXERCISE. Interpret α, by writing out the predicted response `temp` at several different timepoints `t`.

Model 0: at $t=0$, pred. temp $= \alpha + \beta_2$ $\longrightarrow$ α is av. temp. over a single sinusoid.

Model A: at $t=0$, pred. temp $= \alpha + \beta_2$
at $t=2000$, $= \alpha + \beta_2 + 2000\gamma$
$\longrightarrow$ α is avg. temp. over a single sinusoid at ~~0 AD~~ 1 BC.

Model B: at $t=0$, pred. temp $= \alpha + \beta_2 - 2000\gamma$
at $t=2000$, —— $\alpha + \beta_2$
$\longrightarrow$ α is avg temp over a single sinusoid at 2000 AD

Model A asks: what was av. temp in 1 BC?
B —————— 2000 AD?

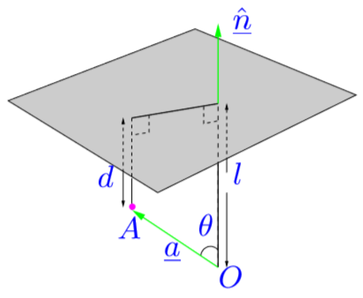
§2.5 The geometry of linear models

To really understand what's going on in a linear model, we need to understand the linear algebra behind them.

NST Maths B, Michaelmas

Example: Distance of point from plane

- What is distance of point A with position vector $\underline{a}$ from plane $\underline{r} \cdot \hat{\underline{n}} = l$?



- Line containing A and point of closest approach of plane to A must be $\parallel \hat{\underline{n}}$; has equation

$$\underline{r} = \underline{a} + \lambda \hat{\underline{n}}$$

- Line meets plane where $\underline{r} \cdot \hat{\underline{n}} = l$, i.e. where

$$l = \underline{a} \cdot \hat{\underline{n}} + \lambda$$

- λ is distance along line from $\underline{a}$ so required distance is $|\underline{a} \cdot \hat{\underline{n}} - l|$

NST Maths B, Easter

Definition. V is called a **vector space over** K, and the elements of V are called **vectors**, if the following **axioms** hold:

A1 For any vectors $\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w} \in V$, $(\boldsymbol{u}+\boldsymbol{v})+\boldsymbol{w} = \boldsymbol{u}+(\boldsymbol{v}+\boldsymbol{w})$. (**Associativity.**)

A2 For any vectors $\boldsymbol{u}, \boldsymbol{v} \in V$, $\boldsymbol{u}+\boldsymbol{v} = \boldsymbol{v}+\boldsymbol{u}$. (**Commutativity.**)

A3 There is a vector in V denoted 0, called the **zero vector** for which $\boldsymbol{u}+0 = \boldsymbol{u}$ $\forall\, \boldsymbol{u} \in V$.

A4 For each vector $\boldsymbol{u} \in V$ there is a vector in V denoted $-\boldsymbol{u}$ for which $\boldsymbol{u}+(-\boldsymbol{u}) = 0$. (**Inverse.**)

A5 For any $a \in K$ and any $\boldsymbol{u}, \boldsymbol{v} \in V$, $a(\boldsymbol{u}+\boldsymbol{v}) = a\boldsymbol{u}+a\boldsymbol{v}$.

A6 For any $a, b \in K$ and any $\boldsymbol{u} \in V$, $(a+b)\boldsymbol{u} = a\boldsymbol{u}+b\boldsymbol{u}$.

A7 For any $a, b \in K$ and any $\boldsymbol{u} \in V$, $(ab)\boldsymbol{u} = a(b\boldsymbol{u})$.

A8 For the unit scalar $1 \in K$ and any $\boldsymbol{u} \in V$, $1\boldsymbol{u} = \boldsymbol{u}$.

The *subspace spanned* by a collection of vectors $\{e_1, \dots, e_K\}$ is the set of all linear combinations

$$\mathcal{S} = \{\lambda_1 e_1 + \cdots + \lambda_K e_K : \lambda_k \in \mathbb{R} \text{ for all } k\}$$

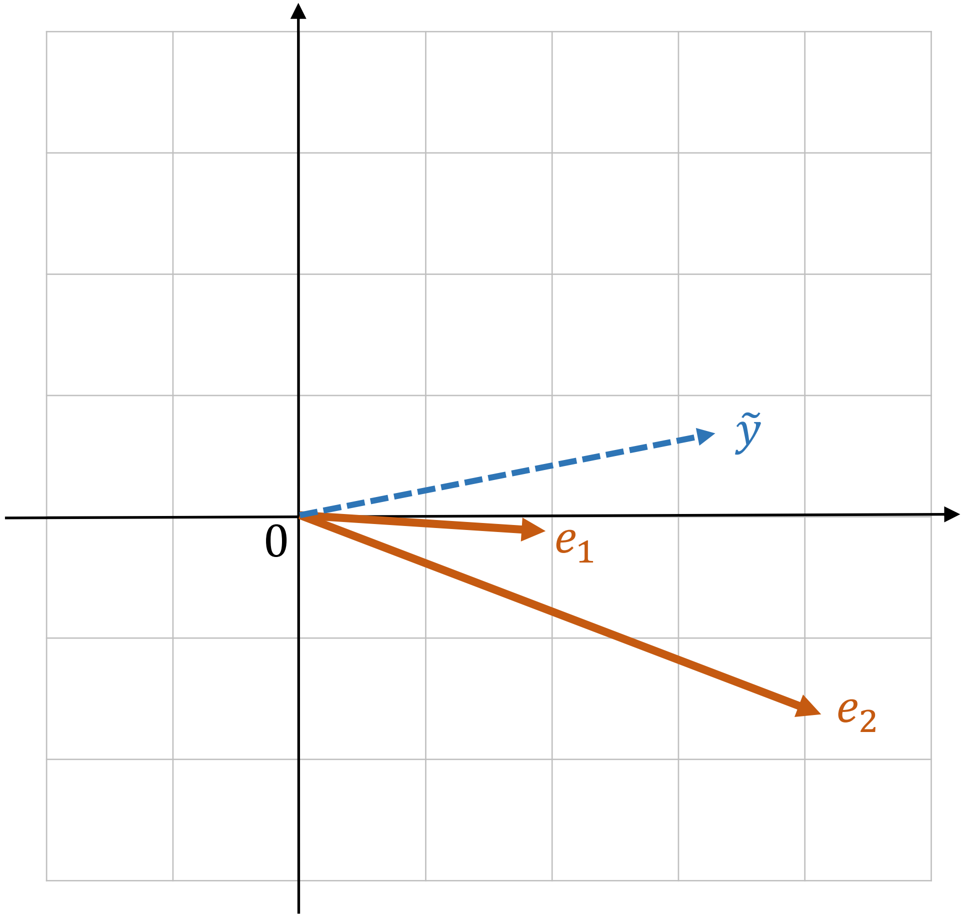
The vectors are *linearly dependent* if at least one of the e_k can be written as a linear combination of the others, i.e. there is some set of real numbers $(\lambda_1, \dots, \lambda_K)$ not all equal to zero such that

$$\lambda_1 e_1 + \cdots + \lambda_K e_K = 0$$

If not, they are *linearly independent*, and

$$\lambda_1 e_1 + \cdots + \lambda_K e_K = 0 \quad \Rightarrow \quad \lambda_1 = \cdots = \lambda_K = 0$$

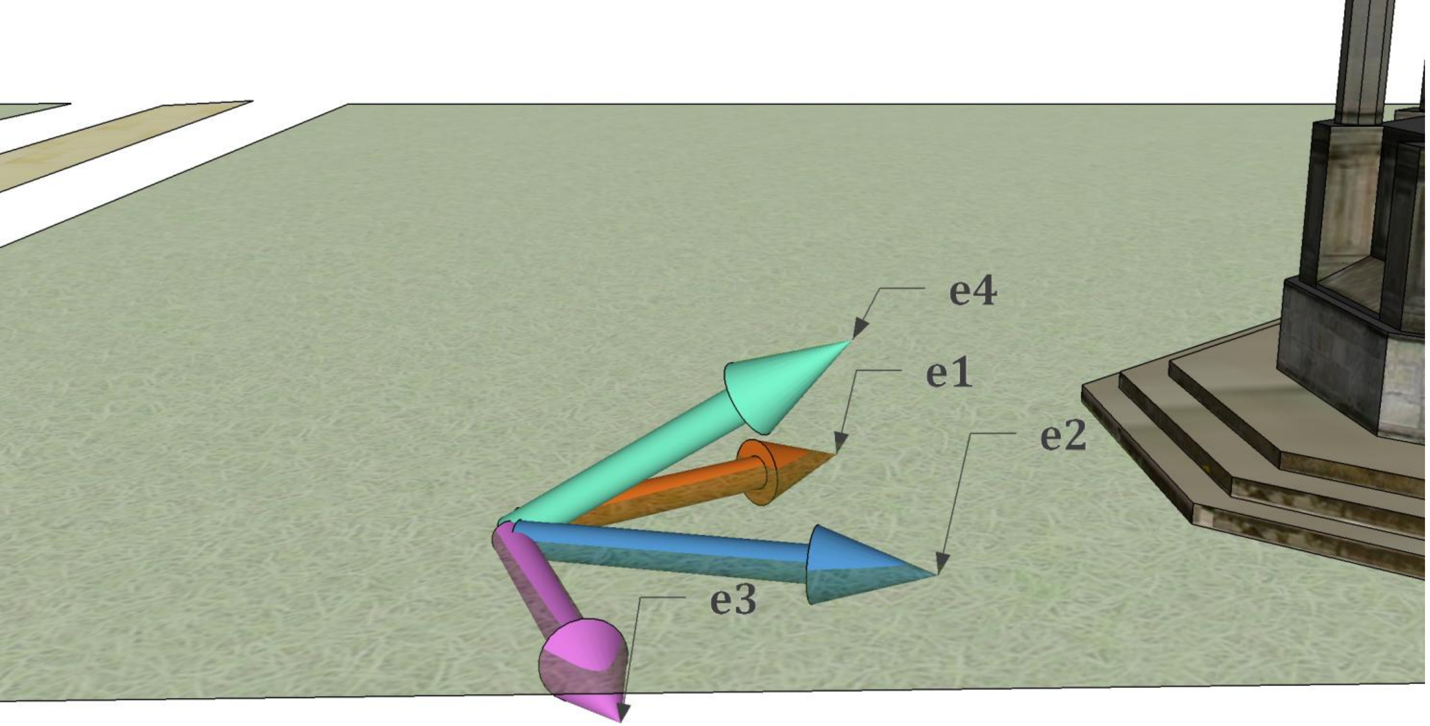
`np.linalg.matrix_rank(np.column_stack([`$e_1, \dots, e_K$`]))` is $< K$ if linearly dependent
$= K$ if linearly independent



The subspace spanned by $\{e_1, e_2\}$ is $\mathbb{R}^2$

Any $\tilde{y} \in \mathbb{R}^2$ can be written as a linear combination of e_1 and e_2

- by eye, $\tilde{y} = 2.5e_1 - 0.3e_2$



The subspace spanned by $\{e_1, e_2, e_3, e_4\}$ is $\mathbb{R}^3$

Are $\{e_1, e_2, e_3, e_4\}$ linearly independent? No.

If we discarded e_2 ...
Are $\{e_1, e_3, e_4\}$ linearly independent? What's the span? They are linearly ind, span is $\mathbb{R}^3$

If we discarded e_1 ...
Are $\{e_2, e_3, e_4\}$ linearly independent? What's the span? They are linearly ind, span is $\mathbb{R}^3$.

Exercise 2.5.2

Are the following five vectors linearly independent? If not, find a subset that is.

$$e_1 = [1,1,1,1]$$
$$e_2 = [0,1,1,0]$$
$$e_3 = [1,0,0,1]$$
$$e_4 = [1,1,1,0]$$
$$e_5 = [0,0,0,1]$$

Are they linearly independent?

$$e_1 = e_4 + e_5$$
$$e_1 = e_2 + e_3$$

So $\operatorname{span}(\{e_1, e_2, e_3, e_4, e_5\})$

$= \operatorname{span}(\{e_1, e_2, e_3, e_4\})$ since $e_5 = e_1 - e_4$

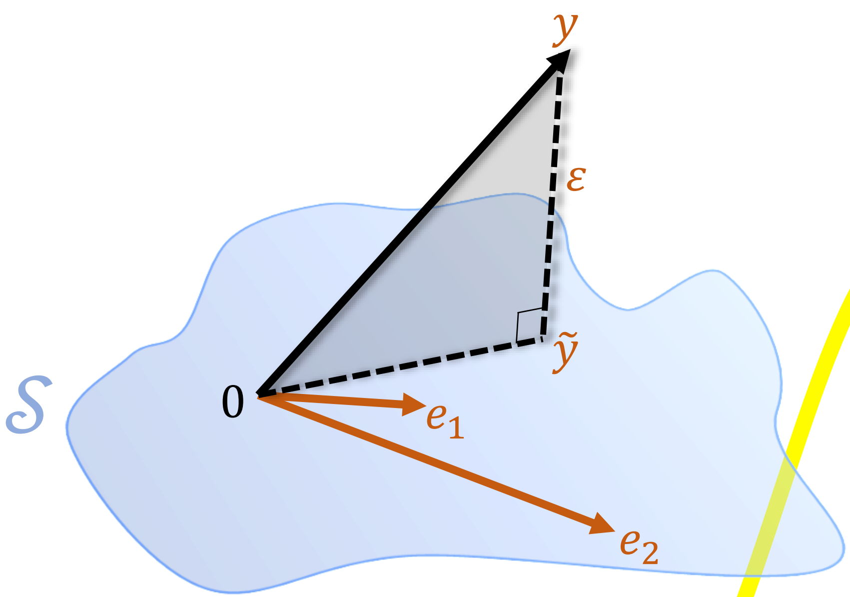
$= \operatorname{span}(\{e_1, e_2, e_4\})$ since $e_2 = e_1 - e_3$

Are $\{e_1, e_2, e_4\}$ linearly independent?

Suppose $\lambda_1 e_1 + \lambda_2 e_2 + \lambda_4 e_4 = 0$

$$\Rightarrow \lambda_1 \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} + \lambda_2 \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} + \lambda_4 \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\Rightarrow \left.\begin{aligned} \lambda_1 + \lambda_4 &= 0 \\ \lambda_1 + \lambda_2 + \lambda_4 &= 0 \\ \lambda_1 + \lambda_2 + \lambda_4 &= 0 \\ \lambda_1 &= 0 \end{aligned}\right\} \Rightarrow \left.\begin{aligned} \lambda_1 &= 0 \\ \lambda_4 &= 0 \\ \lambda_2 &= 0 \end{aligned}\right\} \Rightarrow \text{they are linearly independent.}$$



What does "closest" even mean?
It means: find $\tilde{y} \in \mathcal{S}$ to minimize $\|y - \tilde{y}\|$
i.e. to minimize $\sqrt{\sum_i \varepsilon_i^2}$ where $\varepsilon = y - \tilde{y}$.
The minimization is over all $\tilde{y} \in \mathcal{S}$,
i.e. over all linear combinations of $\{e_1, \dots, e_K\}$.

GEOMETRY OF PROJECTIONS

Let $\mathcal{S}$ be the span of $\{e_1, \dots, e_K\}$.

What's the closest we can get to y, while staying in $\mathcal{S}$?

LECTURE CUT SHORT BY A FIRE ALARM