Example sheet 1

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

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

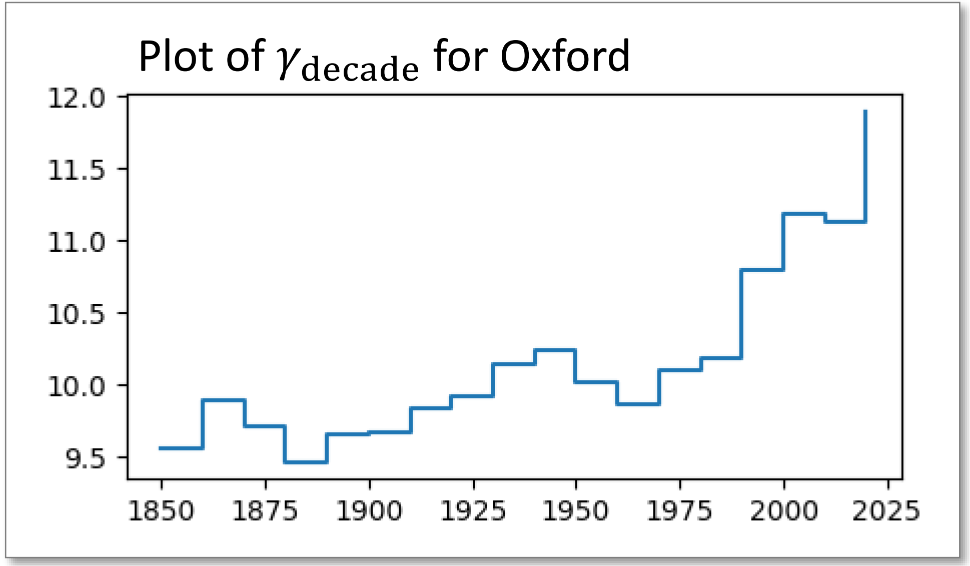
in which the $+\gamma\texttt{t}$ term asserts that temperatures are increasing at a constant rate. We might suspect though that temperatures are increasing non-linearly. To test this, we can create a non-numerical feature out of `t` by

```
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.]*



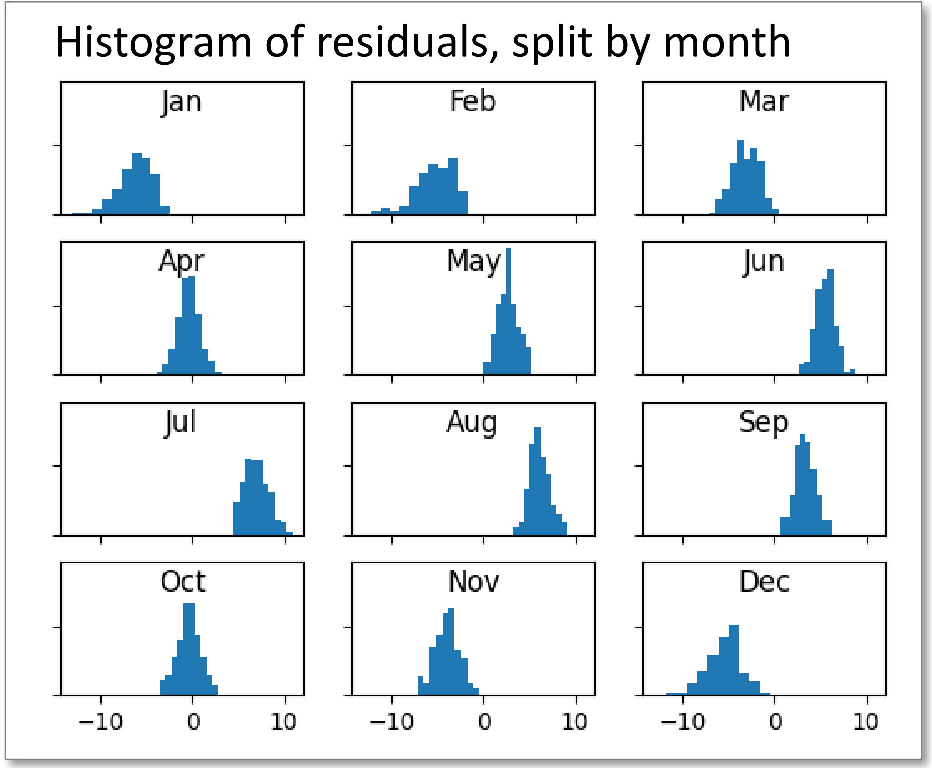
QUESTION. This code doesn't pass the Moodle tester. What's the bug?

sin(2πt) cos(2t)

```
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)
```



§2.3. Diagnosing a model

After fitting a model,

1. Compute the prediction errors a.k.a. the residuals
2. Plot them every way we can think of. They're telling us where our model is poor.

Machine learning models don't fail with nice simple exceptions or incorrect answers. They fail by giving us fishy answers.

The only way to debug them is through data science investigation.

§1–§4. Learning with probability models

Lecture 1	1. Learning with probability models (4:08)
[slides]	1.1 Specifying probability models (15:20)
Lecture 2	1.2 Standard random variables (3:21)
[slides]	1.3 Maximum likelihood estimation (17:35)
	1.4 Numerical optimization (8:01)
Lecture 3	1.5 Likelihood notation (10:00)
[slides]	1.6 Generative models (8:14)
	1.7 Supervised learning (14:18)
	3.1, 3.2 Prediction accuracy *versus* probability modelling (* non-examinable)
Lecture 4	**Mock exam question 1** and walkthrough (23:36)
[slides]	3.3 Neural networks (* non-examinable)
Lecture 5	2.1 Linear modelling (13:27)
[slides]	2.2 Feature design (19:39)
	2.3 Diagnosing a linear model (5:29)
Lecture 6	2.5 The geometry of linear models (12:07)
[slides]	2.6 Interpreting parameters (20:03)
Lecture 7	2.4 Probabilistic linear modelling (9:45)
	4.1 Measuring model fit (* non-examinable)

Example sheet 1

OPTIONAL ex1 practical exercises [ex1.ipynb]

OPTIONAL PyTorch introduction and challenge

OPTIONAL climate dataset challenge [climate.ipynb]

Code snippets: [fitting.ipynb], [lm.ipynb]

Datasets investigated: [climate.ipynb], [stop-and-search.ipynb]

- ex1
  Try the practical exercises, test your answers on Moodle, discuss with your supervisor. For questions, use the Moodle Q&A forum.
- pytorch
  For your own fun, good if you want to do more ML. Submit your answer on Moodle, and I'll share a leaderboard at the end of term.
- climate
  Useful practice if you want to do real data science. Submit your answer on Moodle, and we'll discuss in lectures next week.

TODAY'S AGENDA

§2.3 Model diagnostics ✓

§2.6 Interpreting parameters

§2.4 Least squares estimation & probability

§4 Measuring model fit (* non-examinable)

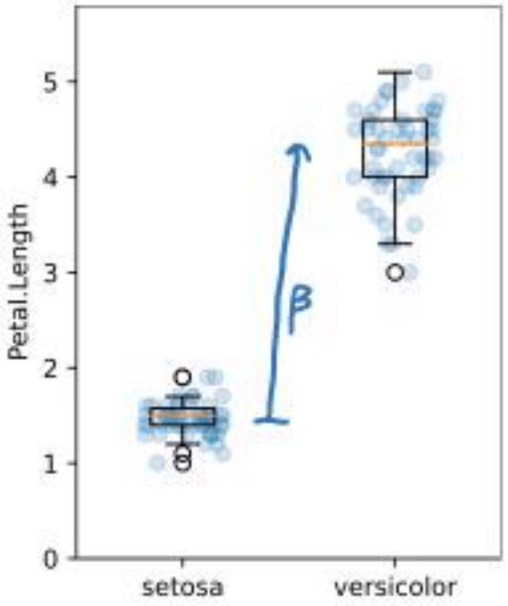
§2.6 Interpreting parameters

- Write out the predicted response for a few typical / representative datapoints.
  This helps see what the parameters mean.

- Write out the features.
  If two models have different features but the same feature space, then (once fitted) they make the same predictions on the dataset.

- Check if the features are linearly dependent.
  If so, the parameters have no intrinsic meaning.
  We say the features are *confounded*, and the parameters are *non-identifiable.*

COMPARING GROUPS



Measurements for condition A: $\mathrm{a} = [\mathrm{a}_1, \mathrm{a}_2, \dots, \mathrm{a}_m]$
Measurements for condition B: $\mathrm{b} = [\mathrm{b}_1, \mathrm{b}_2, \dots, \mathrm{b}_n]$

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

cond	x
A	a_1
A	$\vdots$
$\vdots$	$\vdots$
A	a_m
B	b_1
B	b_2
$\vdots$	$\vdots$
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, $x \approx \alpha$

For a person of type B, $x \approx \alpha + \beta$

β measures the <u>difference</u> between the two groups.

Exercise 2.6.2 (Contrasts)

In the dataset below, of measurements from two groups A and B, interpret the parameters from these models:

$$y \approx \alpha 1_{g=A} + \beta 1_{g=B} \qquad \text{(M1)}$$
$$y \approx \alpha' + \beta' 1_{g=B} \qquad \text{(M2)}$$
$$y \approx \alpha'' + \beta'' 1_{g=A} + \gamma'' 1_{g=B} \qquad \text{(M3)}$$

g	y
A	0.5
A	1.9
B	3.5
B	1.1
B	2.3

Remark about notation.

- $\vec{1}$ means the constant vector [1,1,1,1,1]
- $\vec{g}$ is a vector from the dataset, [A,A,B,B,B]
- $f(\vec{g})$ means "apply the function to each element of $\vec{g}$"
- $1_{\vec{g}=A}$ means "apply the indicator to each element of $\vec{g}$"

What predictions do these models make?

	M1	M2
person from group A:	α	α'
person from group B:	β	$\alpha' + \beta'$

M1 picks out the predicted responses in each group

M2 picks out the difference between the two groups.

M3: features are $\vec{1},\ 1_{\vec{g}=A},\ 1_{\vec{g}=B}$.

These are linearly dependent: $1_{\vec{g}=A} + 1_{\vec{g}=B} = \vec{1}$

So the parameters are not identifiable

e.g.
$$\begin{aligned}\vec{y} &\approx 1.2\, 1_{\vec{g}=A} + 2.3\, 1_{\vec{g}=B}\\ &\approx \vec{1} + 0.2\, 1_{\vec{g}=A} + 1.3\, 1_{\vec{g}=B}\\ &\approx 2.3\, \vec{1} - 1.1\, 1_{\vec{g}=A}\end{aligned}$$

Sign in

The Guardian

News website of the year

Stop and search

This article is more than **3 years old**

Met police 'disproportionately' use stop and search powers on black people

London's minority black population targeted more than white population in 2018 – official figures

Can I set up a model with a parameter that *measures* the quantity I'm interested in?

Example 2.6.4

The UK Home Office makes available a dataset of police stop-and-search incidents. We wish to investigate whether there is racial bias in police decisions to stop-and-search. Consider the linear model

$$y_i \approx \alpha + \beta_{\mathtt{eth}_i}$$

where $\mathtt{eth}_i$ is the officer-defined ethnicity for record i, and y_i records the outcome: $y_i = 1$ if the police found something, 0 otherwise.

a) Write this as a linear equation using one-hot coding.

b) Are the parameters identifiable? If not, rewrite the model so that they are.

c) Does the model suggest there is racial bias in policing actions?

(a)

$$y \approx \alpha 1 + \beta_{As} e_{As} + \beta_{Bl} e_{Bl} + \beta_{Mi} e_{Mi} + \beta_{Oth} e_{Oth} + \beta_{Wh} e_{Wh} \quad \text{where} \quad e_k = 1_{eth=k}$$

ethnic groups: Asian, Black, Mixed, Other, White

(b) These are linearly dependent: $1 = e_{As} + e_{Bl} + e_{Mi} + e_{Oth} + e_{Wh}$

So the parameters are not identifiable, ie we're likely to get silly answers out of linear_model fitting.

The non-identifiable model that was proposed by the question:

$$y \approx \alpha\, 1 + \beta_{\mathrm{As}} e_{\mathrm{As}} + \beta_{\mathrm{Bl}}\, e_{\mathrm{Bl}} + \beta_{\mathrm{Mi}}\, e_{\mathrm{Mi}} + \beta_{\mathrm{Oth}}\, e_{\mathrm{Oth}} + \beta_{\mathrm{Wh}}\, e_{\mathrm{Wh}}$$

(b) Rewrite it to have identifiable parameters.

$$\vec{y} \approx \alpha' \vec{1} + \beta'_{Bl} \vec{e}_{Bl} + \beta'_{Mi} \vec{e}_{Mi} + \beta'_{oth} \vec{e}_{oth} + \beta'_{wh} \vec{e}_{wh}$$

These 5 features are linearly independent.

(c) Interpret the parameters.

For a person with eth = As predicted $y = \alpha'$

eth = Bl $= \alpha' + \beta'_{Bl}$

eth = Mi $= \alpha' + \beta'_{Mi}$

eth = Oth $= \alpha' + \beta'_{oth}$

eth = Wh $= \alpha' + \beta'_{wh}$.

These β'_{eth} measure differences with respect to the baseline of people with eth = Asian.

e.g. if $\beta'_{Bl} > 0$, then the avg. response for people with eth = Bl is higher than that for people with eth = As.

Output from the identifiable model

$$y \approx \alpha' \, \mathbf{1} + \beta'_{\mathrm{Bl}}\, e_{\mathrm{Bl}} + \beta'_{\mathrm{Mi}}\, e_{\mathrm{Mi}} + \beta'_{\mathrm{Oth}}\, e_{\mathrm{Oth}} + \beta'_{\mathrm{Wh}}\, e_{\mathrm{Wh}}$$

Q. Is this meaningful — is it a sign of police bias, or is it just noise?

See next two weeks!

```
some_levels = [k for k in ethnicity_levels if k != 'Asian']
eth_onehot = [np.where(eth==k,1,0) for k in some_levels]

model = sklearn.linear_model.LinearRegression()
model.fit(np.column_stack(eth_onehot), y)
α,βs = model.intercept_, model.coef_

print(f'α = {α}')
for k,β in zip(some_levels, βs):
    print(f'β[{k}] = {β}')
```

```
α’ = 0.261423626892284
β’[Black] = 0.0008154705731562644
β’[Mixed] = 0.028715617211154926
β’[Other] = -0.004471057366589165
β’[White] = -0.003720378247083333
```

Output from the non-identifiable model

$$y \approx \alpha + \underbrace{\beta_{\text{As}} 1_{\text{eth}=\text{As}}}_{\text{Asian}} + \underbrace{\beta_{\text{Bl}} 1_{\text{eth}=\text{Bl}}}_{\text{Black}} + \underbrace{\beta_{\text{Mi}} 1_{\text{eth}=\text{Mi}}}_{\text{Mixed}} + \underbrace{\beta_{\text{Oth}} 1_{\text{eth}=\text{Oth}}}_{\text{Other}} + \underbrace{\beta_{\text{Wh}} 1_{\text{eth}=\text{Wh}}}_{\text{White}}$$

```
ethnicity_levels = np.unique(eth)
eth_onehot = [np.where(eth==k,1,0) for k in ethnicity_levels]

model = sklearn.linear_model.LinearRegression()
model.fit(np.column_stack(eth_onehot), y)
α,βs = model.intercept_, model.coef_

print(f'α = {α}')
for k,β in zip(ethnicity_levels, βs):
    print(f'β[{k}] = {β}')
```

```
α = -34037792910.00365
β[Asian] = 34037792910.26522
β[Black] = 34037792910.265717
β[Mixed] = 34037792910.2939
β[Other] = 34037792910.2604
β[White] = 34037792910.261
```

§2.4 Least squares estimation & probability

Carl Friedrich Gauss
1777–1855

Least squares estimation

Fit the linear model

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

i.e.

$$y_i = \beta_1 e_{1,i} + \cdots + \beta_K e_{K,i} + \varepsilon_i$$

by choosing the parameters $\beta_1, \dots, \beta_K$ so as to minimize the mean square error

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

Maximum likelihood estimation

Fit the probability model

$$Y_i \sim \cdots$$

by choosing the model parameters so as to maximize the log likelihood of the observed data

$$\log \Pr(y_1, \dots, y_n) = \sum_{i=1}^{n} \log \Pr_Y(y_i\,;\dots)$$

Example 2.1.1

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

How does `Petal.Length (PL)` depend on `Sepal.Length (SL)`?

We fitted the linear model

$$\text{PL} \approx \alpha + \beta\ \text{SL} + \gamma\ \text{SL}^2$$

Example

Let's fit the probability model

$$\text{PL}_i \sim \alpha + \beta\ \text{SL}_i + \gamma\ \text{SL}_i^2 + \text{Normal}(0, \sigma^2)$$

Model for a single observation: $\mathrm{PL}_i \sim \alpha + \beta\, \mathrm{SL}_i + \gamma\, \mathrm{SL}_i^2 + N(0, \sigma^2)$

Rewrite it as $Y_i \sim \alpha + \beta e_i + \gamma f_i + N(0, \sigma^2)$

$\sim N(\alpha + \beta e_i + \gamma f_i, \sigma^2)$

Likelihood of a single observation: $\Pr_{Y_i}(y; \alpha, \beta, \gamma, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}[y - (\alpha + \beta e_i + \gamma f_i)]^2}$

Log likelihood of the dataset: $\log \Pr(y_1 \cdots y_n; \alpha, \beta, \gamma, \sigma) = -\frac{n}{2} \log(2\pi\sigma^2) - \frac{1}{2\sigma^2} \sum_{i=1}^{n} [y_i - (\alpha + \beta e_i + \gamma f_i)]^2$

We want to maximize this over $\alpha, \beta, \gamma, \sigma$

Maximize over the unknown parameters,
α, β, γ, and σ:

$$\max_{\alpha,\beta,\gamma,\sigma} \left\{ -\frac{n}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}\left(y_i - (\alpha + \beta e_i + \gamma f_i)\right)^2 \right\}$$

$$= \max_{\sigma}\left[\max_{\alpha,\beta,\gamma} \left\{ -\frac{n}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{i}\left(y_i - (\alpha + \beta e_i + \gamma f_i)\right)^2 \right\} \right]$$

$$= \max_{\sigma}\left[-\frac{n}{2}\log(2\pi\sigma^2) + \max_{\alpha,\beta,\gamma} \left\{ -\frac{1}{2\sigma^2}\sum_{i}\left(y_i - (\alpha + \beta e_i + \gamma f_i)\right)^2 \right\} \right]$$

$$= \max_{\sigma}\left[-\frac{n}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2} \left\{ \min_{\alpha,\beta,\gamma} \sum_{i}\left(y_i - (\alpha + \beta e_i + \gamma f_i)\right)^2 \right\} \right]$$

$$= \max_{\sigma}\left[-\frac{n}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{i}(y_i - \hat{y}_i)^2 \right] \quad \text{where } \hat{y}_i = \underbrace{\hat{\alpha} + \hat{\beta} e_i + \hat{\gamma} f_i}_{\text{obtained by least squares estimation}}$$

$$\Rightarrow \quad \hat{\sigma} = \sqrt{\frac{1}{n}\sum_{i}(y_i - \hat{y}_i)}$$

Maximize over the unknown parameters, α, β, γ, and σ:

$$\max_{\alpha,\beta,\gamma,\sigma} \left\{ -\frac{n}{2} \cdots \right.$$

$$= \max_{\sigma} \left[\max_{\alpha,\beta,\cdots} \cdots \right]$$

$$= \max_{\sigma} \left[-\frac{n}{2} \cdots \right]$$

$$= \max_{\sigma} \left[-\frac{n}{2} \cdots \right]$$

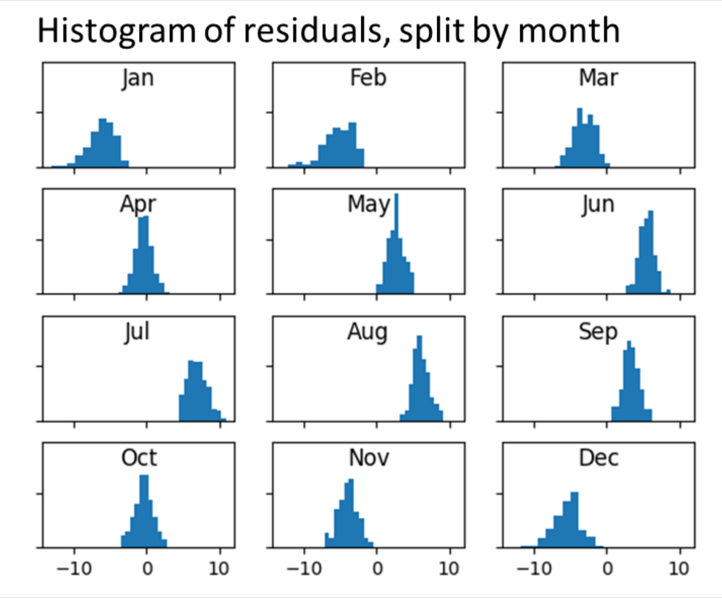
$$= \max_{\sigma} \left[-\frac{n}{2} \cdots \right]$$

$$\Rightarrow \hat{\sigma} = \sqrt{\frac{1}{n} \sum_i (y_i - \hat{y}_i)^2}$$

Least squares estimation *derives* from a Gaussian probability model.

If that model doesn't fit the data, then don't use least squares estimation!

A sensible model diagnostic is to plot a histogram of the residuals, and check they look Gaussian.



Sign in

The Guardian

News website of the year

Stop and search

This article is more than **3 years old**

Met police 'disproportionately' use stop and search powers on black people

London's minority black population targeted more than white population in 2018 - official figures

Let $y_i \in \{0,1\}$ be the outcome for stop-and-search incident i.

$y_i \approx \alpha + \beta_{\mathrm{eth}_i}$ i.e. $Y_i \sim \alpha + \beta_{\mathrm{eth}_i} + N(0, \sigma^2)$

Fit α and $\beta_{\mathrm{Bl}}, \beta_{\mathrm{Mi}}, \ldots$ using least squares estimation

or, equivalently, fit using maximum likelihood estimation

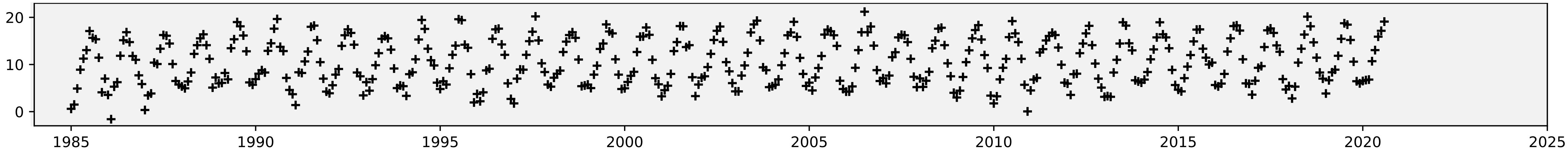
$Y_i \sim \mathrm{Bin}(1, \alpha + \beta_{\mathrm{eth}_i})$

Fit the parameters using maximum likelihood estimation

There's a more advanced version called *Logistic Regression*, for $\mathrm{Bin}(1, \theta_i)$ where θ_i depends on multiple features. It uses softmax. See the code in [`stop-and-search.ipynb`], or Part II Advanced Data Science.

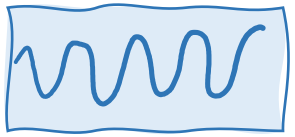
§4. How should we measure how well a model fits the data?

(* non-examinable)



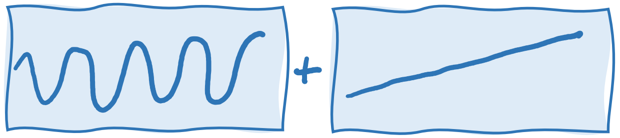
Climate is stable:

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

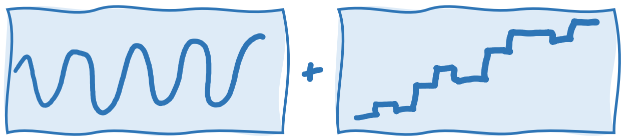


Temperatures are increasing linearly:

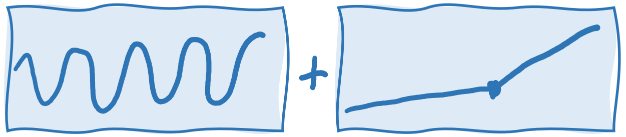
$\mathrm{Temp}(t) \sim \cdots + \gamma t$



Temperatures are increasing, and the rate is nonlinear:



Temperatures are increasing, and the rate is increasing piecewise-linearly:



And if so, when is the tipping point?