Example sheet 1

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

\[ \text{temp} \approx \alpha + \beta_1 \sin(2\pi t) + \beta_2 \cos(2\pi t) + \gamma t \]

in which the \( +\gamma t \) 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

\[ u = \text{'decade_'} + \text{str(math.floor(t/10))} + \text{'0s'} \]

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

\[ \text{temp} \approx \alpha + \beta_1 \sin(2\pi t) + \beta_2 \cos(2\pi t) + \gamma u. \]

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

QUESTION. This code doesn’t pass the Moodle tester. What’s the bug?
§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.
Try the practical exercises, test your answers on Moodle, discuss with your supervisor. For questions, use the Moodle Q&A forum.

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.

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

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

\[ x = \alpha_A 1_{\text{cond}=A} + \alpha_B 1_{\text{cond}=B} \]

Or

\[ x = \alpha + \beta \]

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

$\beta$ measures the difference 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:

\[
y \approx \alpha_{1g=A} + \beta_{1g=B} \quad (M1)
\]
\[
y \approx \alpha' + \beta'_{1g=B} \quad (M2)
\]
\[
y \approx \alpha'' + \beta''_{1g=A} + \gamma''_{1g=B} \quad (M3)
\]

<table>
<thead>
<tr>
<th>$g$</th>
<th>$y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.5</td>
</tr>
<tr>
<td>A</td>
<td>1.9</td>
</tr>
<tr>
<td>B</td>
<td>3.5</td>
</tr>
<tr>
<td>B</td>
<td>1.1</td>
</tr>
<tr>
<td>B</td>
<td>2.3</td>
</tr>
</tbody>
</table>

What predictions do these models make?

- $\alpha_{g=A}$
- $\alpha'_{g=B}$
- $\alpha''_{g=A}$
- $\beta'_{g=B}$
- $\gamma''_{g=B}$

**Remark about notation.**

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

### M1 picks out the predicted responses in each group

- $\alpha_{g=A}$
- $\beta'_{g=B}$

### M2 picks out the difference between the two groups

- $\alpha''_{g=A}$
- $\beta''_{g=B}$
- $\gamma''_{g=B}$

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

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

So the parameters are not identifiable.

**Example:**

\[\bar{y} \approx 1.2\mathbf{1}_{g=A} + 2.3\mathbf{1}_{g=B}\]
\[\approx \mathbf{1} + 0.2\mathbf{1}_{g=A} + 1.3\mathbf{1}_{g=B}\]
\[\approx 2.3\mathbf{1} - 1.1\mathbf{1}_{g=A}\]
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

\[ y_i \approx \alpha + \beta_{\text{eth}_i} \]

where \( \text{eth}_i \) 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) 

\[ y \approx \alpha 1 + \beta_{\text{As}} e_{\text{As}} + \beta_{\text{B}} e_{\text{B}} + \beta_{\text{Mi}} e_{\text{Mi}} + \beta_{\text{M}} e_{\text{M}} + \beta_{\text{Wh}} e_{\text{Wh}} \quad \text{where} \quad e_k = 1 \quad \text{if eth}_i = k \]

(b) These are linearly dependent: 

\[ 1 = e_{\text{As}} + e_{\text{B}} + e_{\text{Mi}} + e_{\text{M}} + e_{\text{Wh}} \]

So the parameters are not identifiable, i.e., we're likely to get silly answers out of linear model fitting.
The non-identifiable model that was proposed by the question:

\[ y \approx \alpha 1 + \beta_{As} e_{As} + \beta_{Bl} e_{Bl} + \beta_{Mi} e_{Mi} + \beta_{Oth} e_{Oth} + \beta_{Wh} e_{Wh} \]

(b) Rewrite it to have identifiable parameters.

\[ \tilde{y} \approx \alpha' \tilde{1} + \beta'_{Bl} \tilde{e}_{Bl} + \beta'_{Mi} \tilde{e}_{Mi} + \beta'_{Oth} \tilde{e}_{Oth} + \beta'_{Wh} \tilde{e}_{Wh} \]

These 5 features are linearly independent.

(c) Interpret the parameters.

For a person with \( \text{eth} = \text{As} \) predicted \( y = \alpha' \)

\[ \text{eth} = \text{Bl} \quad = \alpha' + \beta'_{Bl} \]
\[ \text{eth} = \text{Mi} \quad = \alpha' + \beta'_{Mi} \]
\[ \text{eth} = \text{Oth} \quad = \alpha' + \beta'_{Oth} \]
\[ \text{eth} = \text{Wh} \quad = \alpha' + \beta'_{Wh} \]

These \( \beta' \)s measure differences with respect to the baseline of people with \( \text{eth} = \text{Asian} \).

E.g. if \( \beta'_{Bl} > 0 \), then the response for people with \( \text{eth} = \text{Bl} \) is higher than that for people with \( \text{eth} = \text{As} \).
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} \quad \beta = ', end='')
for k,β in zip(some_levels, βs):
    print(f'\beta[\{k\}] = {β}', end=' ')
Output from the non-identifiable model

\[ y \approx \alpha + \beta_{\text{As}} 1_{\text{eth=As}} + \beta_{\text{Bl}} 1_{\text{eth=Bl}} + \beta_{\text{Mi}} 1_{\text{eth=Mi}} + \beta_{\text{Oth}} 1_{\text{eth=Oth}} + \beta_{\text{Wh}} 1_{\text{eth=Wh}} \]

Asian  Black  Mixed  Other  White

\[
\begin{align*}
\alpha &= -34037792910.00365 \\
\beta[\text{Asian}] &= 34037792910.26522 \\
\beta[\text{Black}] &= 34037792910.265717 \\
\beta[\text{Mixed}] &= 34037792910.2939 \\
\beta[\text{Other}] &= 34037792910.26049 \\
\beta[\text{White}] &= 34037792910.261383
\end{align*}
\]
§2.4 Least squares estimation & probability
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 \( \beta_1, \ldots, \beta_K \) so as to minimize the mean square error

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

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

\[ PL \approx \alpha + \beta SL + \gamma SL^2 \]

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

\[ \log \Pr(y_1, \ldots, y_n) = \sum_{i=1}^{n} \log \Pr_Y(y_i; \cdots) \]

Example

Let’s fit the probability model

\[ PL_i \sim \alpha + \beta SL_i + \gamma SL_i^2 + \text{Normal}(0, \sigma^2) \]
Model for a single observation: \( P_{L_i} \sim \alpha + \beta S_{L_i} + \gamma S_{L_i}^2 + N(0, \sigma^2) \)

Rewrite it as

\[ Y_i \sim \alpha + \beta e_i + \gamma e_i^2 + N(0, \sigma^2) \]

\[ \sim N(\alpha + \beta e_i + \gamma e_i^2, \sigma^2) \]

Likelihood of a single observation:

\[
P_{Y_i}(y; \alpha, \beta, \gamma, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2} [y - (\alpha + \beta e_i + \gamma e_i^2)]^2}
\]

Log likelihood of the dataset:

\[
\log P_r (y_1, \ldots, y_n ; \alpha, \beta, \gamma, \sigma) = -\frac{n}{2} \log (2\pi\sigma^2) - \frac{1}{2\sigma^2} \sum_{i=1}^{n} [y_i - (\alpha + \beta e_i + \gamma e_i^2)]^2
\]

We want to maximize this over \( \alpha, \beta, \gamma, \sigma \).
Maximize over the unknown parameters, $\alpha, \beta, \gamma$, and $\sigma$:

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

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

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

$$
= \max_{\sigma} \left[ -\frac{n}{2} \log(2\pi \sigma^2) - \frac{1}{2\sigma^2} \sum_{i} (Y_i - \hat{y}_i)^2 \right] \quad \text{where} \quad \hat{y}_i = \hat{\alpha} + \hat{\beta} \epsilon_i + \hat{\gamma} f_i
$$

$$
\Rightarrow \hat{\sigma} = \sqrt{\frac{1}{n} \sum_{i} (Y_i - \hat{y}_i)^2}
$$

obtained by least squares estimation.
Maximize over the unknown parameters, $\alpha, \beta, \gamma$, and $\sigma$:

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

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

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

$$= \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2} \frac{(y_i - \mu)^2}{\sigma^2}}$$

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

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

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

Fit $\alpha$ and $\beta_{B1}, \beta_{Mi}, \ldots$ using least squares estimation

or, equivalently, fit using maximum likelihood estimation

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

Fit the parameters using maximum likelihood estimation

There's a more advanced version called Logistic Regression, for Bin$(1, \theta_i)$ where $\theta_i$ 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:
\[ \text{Temp}(t) \sim a + b \sin(2\pi(t + \phi)) + N(0, \sigma^2) \]

Temperatures are increasing linearly:
\[ \text{Temp}(t) \sim \ldots + \gamma t \]

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

And if so, when is the tipping point?