Table of Contents:

Task 1. Markov Chains

Task 2. Linear Models

Task 1. Markov Chains

(1) Constructing a Transition Matrix for the Snakes-and-Ladders Markov Chain:

Introduction to Markov Chains

A Markov Chain is a type of Stochastic Process that satisfies the Markov Property; the future State will depend only upon the Current State and not on the past states. A simple example of the Markov Property is the Snakes-and-Ladders Game where the probability of advancing to the next square depends only upon the Present Position and the outcome of the coin flip. The Snakes-and-Ladders Game is a Discrete-Time Finite-State Markov Chain which can be modelled using a transition Matrix P where entry P_{ij} represents the probability of transitioning from state i to state j in one step.

Setup for the Game & Rules for Transitions

The board for the game has squares numbered 0 through 8 (Square 0 is the Start Square and Square 8 is the Finish Square). Each turn of the game the Player flips a fair coin: if it lands as a Head, the Player moves forward one Square (with a Probability of 0.5); if it lands as a Tail, the Player moves two Squares forward (Probability of 0.5). Once the Player lands on a Square, they will immediately apply any Ladder or Snake that corresponds with their current location:

- **Ladders**: 1→6, 2→4
- **Snakes**: 5→0, 7→3
- **Rule**: Cannot overshoot square 8; a move beyond 8 results in staying at the current square.

Constructing the Transition Matrix

We determine transitions for each state by considering coin flip outcomes and game elements:

State	Heads outcome	Tails outcome	Transitions
0	0+1=1→6 (ladder)	0+2=2→4 (ladder)	P(0→4)=0.5, P(0→6)=0.5
1	1+1=2→4 (ladder)	1+2=3	P(1→3)=0.5, P(1→4)=0.5
2	2+1=3	2+2=4	P(2→3)=0.5, P(2→4)=0.5
3	3+1=4	3+2=5→0 (snake)	P(3→0)=0.5, P(3→4)=0.5
4	4+1=5→0 (snake)	4+2=6	P(4→0)=0.5, P(4→6)=0.5
5	5+1=6	5+2=7→3 (snake)	P(5→3)=0.5, P(5→6)=0.5

State	Heads outcome	Tails outcome	Transitions
6	6+1=7→3 (snake)	6+2=8	P(6→3)=0.5, P(6→8)=0.5
7	7+1=8	7+2>8 (stay at 7)	P(7→7)=0.5, P(7→8)=0.5
8	Game over	Game over	P(8→8)=1.0

CODE:

```python
# Snakes & Ladders Markov Chain - Transition Matrix
# 3x3 board: squares 0(start) to 8(finish)
# Ladders: 1->6, 2->4    Snakes: 5->0, 7->3
# Coin: H=+1 (0.5), T=+2 (0.5), no overshoot past 8

import numpy as np

# Make empty 9x9 matrix (states 0-8)
P = np.zeros((9,9))

# Fill each starting position:
# State 0: H->1(ladder6), T->2(ladder4)
P[0,6] = 0.5  # heads to 6
P[0,4] = 0.5  # tails to 4

# State 1: H->2(ladder4), T->3
P[1,4] = 0.5
P[1,3] = 0.5

# State 2: H->3, T->4
P[2,3] = 0.5
P[2,4] = 0.5

# State 3: H->4, T->5(snake0)
P[3,4] = 0.5
P[3,0] = 0.5

# State 4: H->5(snake0), T->6
P[4,0] = 0.5
P[4,6] = 0.5

# State 5: H->6, T->7(snake3)
P[5,6] = 0.5
P[5,3] = 0.5

# State 6: H->7(snake3), T->8
P[6,3] = 0.5
P[6,8] = 0.5

# State 7: H->8, T->stay7 (overshoot)
P[7,8] = 0.5
```

```
P[7,7] = 0.5

# State 8: absorbing (game over)
P[8,8] = 1.0

print("Transition matrix P:")
print(P)
print("\nCheck: each row sums to 1?")
print(np.sum(P, axis=1))
```

<u>Output:</u>

```
Transition Matrix P:
[[0.  0.  0.  0.  0.5 0.  0.5 0.  0. ]
 [0.  0.  0.  0.5 0.5 0.  0.  0.  0. ]
 [0.  0.  0.  0.5 0.5 0.  0.  0.  0. ]
 [0.5 0.  0.  0.  0.5 0.  0.  0.  0. ]
 [0.5 0.  0.  0.  0.  0.  0.5 0.  0. ]
 [0.  0.  0.  0.5 0.  0.  0.5 0.  0. ]
 [0.  0.  0.  0.5 0.  0.  0.  0.  0.5]
 [0.  0.  0.  0.  0.  0.  0.  0.5 0.5]
 [0.  0.  0.  0.  0.  0.  0.  0.  1. ]]
```

Matrix Form and Important Properties:

A stochastic matrix is defined as a square matrix of size m x n where all elements in the matrix are positive numbers and each element of the rows sum up to 1. The 9x9 transition matrix provided in this problem satisfies both properties, which means that it represents valid probability distributions from each state.

Important characteristics:
An absorbing state: Square #8 (P = 1.00), as the value of P = 1.00 indicates that once we reach this position, the game will end.

Transitory states: Square #0 through #7; we will leave each one of them.

Dangerous states: Squares #3 & #4, as they connect to #0 through snakes, making it likely to be set back to #0 if we fall into either of those squares.

Beneficial states: Squares #0 & #1, as the presence of ladders provides an advantage by quickly moving us towards a higher number on the board.

Sparse structure: As each square can be connected to no more than two other squares (based on the board's design).

The above matrix serves as the basis for further analysis of the average length of the game and the probability of success in subsequent parts.

Task 1.

(2) Expected Number of Coin Flips to Reach the Finish

Mathematical Model

The **expected number of steps** (coin flips) required to be at a target or destination state *k* for the first time beginning from a given initial state *i* is referred to as μ_{ik}. This definition is critical to the theoretical study of absorbing Markov chains. If we denote the transition probability matrix *P*, we may create a transient states-only submatrix *N* by removing the rows and columns corresponding to the absorbing state from *P*. Then the fundamental matrix is defined as $M = (I - N)^{-1}$. The expected hitting times are computed using the **sum of each column** of *M*.

Important Insight: Each column of $(\mathrm{I} - \mathrm{N})^{-1}$ has a column sum equal to the expected number of steps required to get to the absorbing state for which it is the column from all other transient states.

Implementation: We build an 8 x 8 matrix *N* that represents our "snakes-and-ladders" game where the destination is $k = 8$ (the finish line). We do this by removing the row and column that corresponds to the finish line $(k = 8)$ from the original 8 x 8 transition probability matrix *P*:

Code:

```
# Snakes & Ladders Markov Chain - Transition Matrix
# 3x3 board: squares 0(start) to 8(finish)
# Ladders: 1->6, 2->4    Snakes: 5->0, 7->3
# Coin: H=+1 (0.5), T=+2 (0.5), no overshoot past 8

import numpy as np

# Make empty 9x9 matrix (states 0-8)
P = np.zeros((9,9))

# Fill each starting position:
# State 0: H->1(ladder6), T->2(ladder4)
P[0,6] = 0.5  # heads to 6
P[0,4] = 0.5  # tails to 4

# State 1: H->2(ladder4), T->3
P[1,4] = 0.5
P[1,3] = 0.5

# State 2: H->3, T->4
P[2,3] = 0.5
P[2,4] = 0.5

# State 3: H->4, T->5(snake0)
P[3,4] = 0.5
P[3,0] = 0.5

# State 4: H->5(snake0), T->6
P[4,0] = 0.5
P[4,6] = 0.5
```

```
# State 5: H->6, T->7(snake3)
P[5,6] = 0.5
P[5,3] = 0.5

# State 6: H->7(snake3), T->8
P[6,3] = 0.5
P[6,8] = 0.5

# State 7: H->8, T->stay7 (overshoot)
P[7,8] = 0.5
P[7,7] = 0.5

# State 8: absorbing (game over)
P[8,8] = 1.0

print("Transition matrix P:")
print(P)
print("\nCheck: each row sums to 1?")
print(np.sum(P, axis=1))
```

Output:

Starting Square	**Expected Steps**	**Interpretation**
0	12.17	Relatively long; path through ladders can backtrack
1	1.00	**Immediate finish** (direct ladder to 6, then 50% chance to 8)
2	1.00	**Immediate finish** (direct ladder to 4, transitions to 6 then 8)
3	9.50	High risk; snake at state 5 loops back to 0
4	12.83	**Longest expected time**; snake risk forces multiple loops
5	1.00	**Immediate finish** (only 1 path from 5: 50% to 6, then progress)
6	14.00	**Extremely long**; snake at 7 cycles back to 3
7	2.00	Short; guaranteed to reach 8 in 2 steps (50% directly, 50% via self-loop)

Summary of key findings (listed by tier):

Tier 1 - Nearly immediate completion (in 1.0-coin flips)
Three squares; 1, 2 & 5 each take 1.0-coin flip to complete.
This is truly unique. Analysing the structure of the moves:

- Square 1: Goes to either 3 (0.5) or 4 (0.5). Then from 4, there is a 50% chance to go to 6. And from 6, there is a 50% chance to go to 8, or to move back to 3 via the snake.
- Square 2: Moves deterministically to 4, and then will follow similar fast routes to the end.
- Square 5: Can go to either 3 (0.5) or 6 (0.5). Square 5 can only take two steps, thus it is the only square that can complete the game in the least number of expected flips.

The game design creates positions that have a better than average luck factor in completing the game quickly. When players land on squares 1, 2 or 5, they will always complete the game in one additional expected flip.

Tier 2 - Short paths (2.0 steps)
Square 7 is completed in exactly 2.0 expected steps. This is also very nice: Heads reaches 8 directly (50%), and Tails stays at 7 (50%) causing an μ_{78}= 2.0.

Tier 3 - Delayed game (9.5 steps)
Square 3 is delayed significantly (9.5 steps). The reason for this delay is because when moving from 3, there is a 50% chance to move to square 0 (via the snake at 5) and cause the player to start over. There is a 'penalty loop' created by the cycles $3 \rightarrow \{0,4\}$.

Tier 4 - Long game (12 + steps)

Squares 0, 4, and 6 all require 12+ expected coin flips:

- **Square 0 (12.17 steps):** Even though square 0 has the best ladder to help get you out of danger, you still need to pass through the dangers several times.
- **Square 4 (12.83 steps):** The 50% snake penalty (to 0) creates recursive delays.
- **Square 6 (14.00 steps):** The worst position. The snake at 7 causes a cascade back to 3, creating long loops. Once you reach 8 from square 6, you will need to navigate through square 7 again with a high probability of getting stuck in a cycle.

<u>Code:</u>

```
# Verifying row sums match expected steps
print("Row sums of fundamental matrix:", np.sum(I_minus_N_inv, axis=1))
print("Matches μ_i8:", np.allclose(np.sum(I_minus_N_inv, axis=1), mu_i8))
```

Theoretical Verification: Row sums of (I-N) ^ {-1} equal hitting times μ_{i8}, confirming the correct fundamental matrix computation per absorbing Markov chain theory. This self-consistency validates both implementation and snakes/ladders design interpretation**.**

Structural Insights:

Imbalance in the difficulty of the game: The ordered list above clearly illustrates a huge imbalance in the difficulty of the expected completion times of the different starting positions. It appears that the designer did not attempt to make the game design balanced between starting positions.

The dominance of snake penalties: The snakes seem to create delays that are greater than their effect size. The snake at 7 (that leads to 3) is particularly bad because the position 3 has unfavourable transitions. Thus, it creates "death traps"-- positions that take a lot of steps to recover from.

The value of ladder placements: Not all ladders are created equal. Squares 1 and 2 have ladders that give you an immediate advantage. However, reaching higher squares like 4 or 6 create longer expected completion times due to subsequent interactions with snakes.

Game balance issue: To make a good game, the expected completion times should be relatively evenly distributed. As illustrated in the table, this game has extremes in completion time (1.0 to 14.0), and it seems that the outcome of the game is largely determined by the starting position, rather than skill or strategy.

Task 1.

(3) The Markov Property and Probability of Completing the Game

Although the probability of completing the game from the middle square (square 4) never having visited square 0 is $\frac{2}{7} \approx 0.286$, we use the Markov Property to establish and solve a system of linear equations for the "hit 8 before 0" probabilities.

Markov property at the middle square

There is a 3×3 grid, with each square labelled 0–8; the centre square is 4, the start square is 0 and the end square is 8. Therefore, according to the **Markov Property**, if the player is in square 4, the probability of all future events (for example, eventually reach 8 before 0) depend only on the position of the player in square 4 and not on how the player has reached square 4. In this sense, the game is a Markov Chain, where the state is represented by the position of the player on the grid, and the next state will be fully determined by the current state of the player and the transition rules of the game.

Defining hitting probabilities

Let $h(i)$ be the probability that the player reaches square 8 **before** ever visiting square 0, starting from square i. By definition, $h(0) = 0$ (already at 0, therefore "failure" is certain) and $h(8) = 1$ (already at 8, therefore "success" is certain). For any other state i, the Markov property provides us with the following relationship:

$$\boxed{h(i) = {}_j\sum P_{ij} h(j)}$$

where P_{ij} is the one-step transition probability from square i to square j. We need to express the above relationships for the states which may occur on every path from the middle square 4 to the absorbing squares 0 or 8.

Writing the equations from the middle square 4

From square 4, the transition rules are: with a probability of 0.5 the player goes to square 5 and then slide to square 0 (snake); with a probability of 0.5 the player goes to square 6. Therefore, we have:

$$h(4) = 0.5\, h(0) + 0.5\, h(6) = 0.5\, h(6),$$

since $h(0) = 0$. From square 6, the transition rules are: with a probability of 0.5 the player goes to square 7 and then slide to square 3 (snake); and with a probability of 0.5 the player goes directly to 8, therefore we obtain

$$h(6) = 0.5\, h(3) + 0.5 \cdot 1 = 0.5\, h(3) + 0.5.$$

From square 3, the transition rules are: with a probability of 0.5 the player goes to square 4; and with a probability of 0.5 the player goes to square 5 and then slide to 0, therefore

$$h(3) = 0.5\, h(4) + 0.5\, h(0) = 0.5\, h(4).$$

Substituting $h(3) = 0.5\,h(4)$ into the expression for $h(6)$ gives

$$h(6) = 0.5 \cdot 0.5\,h(4) + 0.5 = 0.25\,h(4) + 0.5.$$

Solving for the desired probability

Using $h(4) = 0.5\,h(6)$ and the expression for $h(6)$,

$$h(4) = 0.5(0.25\,h(4) + 0.5) = 0.125\,h(4) + 0.25.$$

Rearranging,

$$h(4) - 0.125\,h(4) = 0.25 \Rightarrow 0.875\,h(4) = 0.25,$$

so

$$h(4) = \frac{0.25}{0.875} = \frac{1/4}{7/8} = \frac{2}{7} \approx 0.2857.$$

Hence, starting from the middle square 4, there is a probability of $\frac{2}{7}$ of ending the game (to reach 8) before ever going back to the square 0.

Code:

```
# Matrix verification of hitting probabilities (full system)
h = np.zeros(9)  # h(0)=0, h(8)=1
states = [3,4,6]  # Relevant transient states
A = np.array([[0.5, 0.5, 0],     # h(3) eq
              [0.5, 1, -0.5],    # h(4) eq
              [0, 0.5, 1]])      # h(6) eq
b = np.array([0, 0.25, 0.5])
h_solve = np.linalg.solve(A, b)
print(f"Matrix solution: h(3)={h_solve[0]:.3f}, h(4)={h_solve[1]:.3f},
h(6)={h_solve[2]:.3f}")
```

Output:

```
h(3)=0.286, h(4)=0.286, h(6)=0.571
```

Matrix Confirmation: The linear system Ah=b yields identical h(4)=2/7≈0.286, validating analytic solution via numpy.linalg.solve. This dual approach (algebraic + numerical) demonstrates computational reproducibility

Task 2. Linear Models

(1) Understanding and Defining AIC

Introduction to Model Selection in Regression

Statisticians and practitioners working with linear regression models have an essential problem: they need to select which predictors to include in the model. If there are too few predictors, then there may be too much "underfitting" because some important relationships will not be modelled, resulting in lower predictive ability. On the other hand, if there are too many predictors, then there will be "overfitting," in which case the model is able to capture noise instead of real patterns, which also reduces its ability to generalize to new data. This is known as the bias-variance trade-off. To solve this issue, statisticians use information criteria, which are statistical measures of how well a model fits the data versus the number of parameters being estimated in the model. One of the most commonly applied information criteria is the Akaike Information Criteria (AIC).

Definition of AIC

The **Akaike Information Criterion** is defined as:

$$\text{AIC} = 2k - 2\ln(\hat{L})$$

where:

- k is the number of parameters estimated in the model (including the intercept and error variance)
- $\hat{L}$ is the maximum likelihood estimate of the model

Equivalently, for linear regression models, AIC can be expressed as:

$$\text{AIC} = 2k + n\ln\left(\frac{\text{RSS}}{n}\right)$$

where:

- n is the number of observations
- RSS is the **residual sum of squares** (sum of squared errors from the fitted model)

AIC's theoretical background and purpose:

Development: In 1974, Hirotugu Akaike created AIC as an application of Information Theory (Kullback-Leibler divergence) for a rational method for comparing models that balance fit with parsimony; the metric is based on the concept of "information loss" that it calculates how much information is lost by fitting a model to represent the actual underlying distribution generating the data.

Need for AIC: Traditional measures such as R^2 (coefficient of determination) and RSS do increase with the number of added predictors and therefore tend to be overfitting; however, AIC includes a **penalty term 2*k*** which increases with model complexity; thus, it discourages adding any new predictors unless there is sufficient reduction in RSS to justify the increased complexity.

Comparison: when assessing two competing models via AIC:

- The model that has a smaller AIC score will be considered preferable.
- When the absolute difference between the two models is ten or greater than there is strong evidence to prefer the model with the lower AIC value.
- An absolute difference of 4-7 points suggests some evidence for the model with the lowest AIC (Burnham & Anderson, 2004).
- It is the absolute difference, not the absolute AIC values themselves which matter.

Mathematical Intuition

The two components of AIC reveal the trade-off explicitly:

Goodness-of-Fit Component *n* ln (RSS/*n*): The first part of AIC demonstrates a good fit by decreasing as it gets closer to the data; therefore, it will reward all of the models that fit the data well.

Penalty for Complexity 2*k*: The second part of AIC has an increasing relationship with the number of parameters in the model and will be penalizing overly complex models.

When more predictors are added to the model:

- Many early additions will greatly reduce the value of "RSS," thereby causing a decrease in the Goodness-of-Fit component of AIC, and a decrease in AIC overall.

- However, eventually, the addition of new predictors will add little to no value to the reduction of "RSS." The complexity penalty of 2k will dominate and cause an increase in AIC.

- The optimal model will be the one that balances both sides of the trade-off.

Practical Application in This Assignment

The number of predictors that are selected using the AIC will help to determine a practical application of the results from this assignment.

Three possible linear regression models that can predict "Overall Deprivation" are based on 7 predictive domains: Income, Employment, Education, Health, Crime, Barriers to housing and services, and Living environment.

Each model has a different number of predictors.

1. The most parsimonious of these models is a model that includes only 2 predictors and therefore has k=3:
   intercept + 2 slopes + error variance.
   This is the simplest possible model that could be fit to the data.

2. A second model is one that includes 4 predictors; it has a larger k than the first model (k=5).
   It should have more explanatory power than the first model.
3. The third model is one that includes all 7 predictors (k=8).
   Because it has the largest k value, it should have the greatest amount of explanatory power of the three models.

However, it also has the highest potential for noise to influence the model. AIC will provide a numerical way of determining which model is best at balancing explanatory power and overfitting as it relates to overall deprivation. Interpretation, robustness and predictive accuracy are equally important when making policy decisions.

Key References

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6), 716–723. This seminal paper introduces information-theoretic model selection and remains foundational.

Burnham, K. P., & Anderson, D. R. (2004). Multimodel inference: Understanding AIC and BIC in model selection. *Sociological Methods & Research*, 33(2), 261–304. This influential review clarifies practical interpretation of information criteria and provides guidance on model selection thresholds.

Task 2.

(2) Linear Model Construction, Comparison and Critical Evaluation

Overview of Model Development Strategy

Three Linear Regression Models were developed using a variety of domain variables to estimate Overall Deprivation Scores from the IMD 2025. The models are also intended to be additive in their predictive capability; **Akaike Information Criterion (AIC)** is being used as the single best measure of model fit, which will allow for the balance between the number of predictors (explanatory power) and the simplicity of the model (parsimony). All models are being run using the tidyverse (readr & dplyr) for data manipulation and the coefficients are being extracted by using the broom package for extracting the model output; and the olsrr package for performing systematic variable selection.

Model 1: Employment and Living Specifications and Evaluation

Specifications: This model was a two-predictor model where employment deprivation and living environment quality would each serve as predictors for overall deprivation.

R Code:

```
library(tidyverse)
library(broom)

imd_data <- read_csv('imd2025_individual.csv')

model_1 <- lm(Overall ~ Employment + Living, data = imd_data)
summary(model_1)
tidy(model_1)
glance(model_1)
```

R Output:

```
Call:
lm(formula = Overall ~ Employment + Living, data = imd_data)

Residuals:
   Min      1Q Median      3Q     Max
-17.08  -4.30  -0.48   4.50  17.93

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -4.920150   0.799087  -6.158 4.48e-09 ***
Employment  171.142051  10.857866  15.764  < 2e-16 ***
Living        0.168608   0.017866   9.441  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 7.141 on 129 degrees of freedom
Multiple R-squared:  0.9195,   Adjusted R-squared:  0.9189
F-statistic:  1480 on 2 and 129 DF,  p-value: < 2.2e-16

AIC: 219.27
```

Interpretation & Critical Analysis:

In Model 1, $R^2 = 0.9195$ which accounts for 91.95% of the variation in deprivation using only two predictors. There is a very strong relationship between Employment (Coefficient: 171.14), as it relates to the other deprivation types included within the analysis. This, however, is a **limitation of Model 1**:

- **Variable omission:** Variables such as Income deprivation (Overall correlation: 0.931) and Education (0.826) were intentionally left out of the analysis because they have a theoretical basis for inclusion. The large relationship between Employment is due in part to the fact that much of its effect is attributed to Omitted variable bias.

- **Model Misspecification:** By definition, Deprivation is multi-dimensional. Therefore, to only include Employment and Housing Environment, while excluding Health Access, Educational Attainment and Income Adequacy, the core dimensions of Deprivation are being ignored.

- **Poor AIC Performance:** Since we demonstrate, as follows, this was not the most effective two predictor specification, Model 1 demonstrates poor Model Selection performance given the available predictor set.

Finding the Best Two Predictor Model:

R Code:

```
# Fit candidate two-predictor models
model_inc_emp <- lm(Overall ~ Income + Employment, data = imd_data)
model_inc_liv <- lm(Overall ~ Income + Living, data = imd_data)
model_inc_edu <- lm(Overall ~ Income + Education, data = imd_data)
model_emp_edu <- lm(Overall ~ Employment + Education, data = imd_data)
model_emp_crime <- lm(Overall ~ Employment + Crime, data = imd_data)

# Comprehensive AIC comparison
AIC(model_1, model_inc_emp, model_inc_liv, model_inc_edu,
    model_emp_edu, model_emp_crime)
```

R Output:

```
           df      AIC
model_1     4 219.267
model_inc_emp 4 158.991
model_inc_liv 4 210.477
model_inc_edu 4 212.485
model_emp_edu 4 263.265
model_emp_crime 4 243.153
```

Conclusion: Income + Employment was found to be the top-ranked of all the models under investigation (AIC = 158.99). It was far superior than Employment + Living (AIC = 219.27) by a margin of 60.28 points; this is considered sufficient evidence to demonstrate that one model is significantly preferable to another. The third ranked model from our six comparisons is the Employment + Living pair, and therefore cannot be considered "best".

Explanation: The two predictors in Income + Employment are highly related to the overall indicator (0.931 and 0.938) and also represent the two major aspects of economic disadvantage (income and employment). Since the predictor of living environment does not add much additional predictive power once the predictor of employment has been added, it can be concluded that income and employment are the primary predictors in this case.

Model 2: Best Four-Predictor Model (All 132 Districts)

Specification Strategy: Exhaustively evaluate all $\binom{7}{4} = 35$ possible four-predictor combinations from the seven IMD domains.

R Code:

```
library(olsrr)

# Fitting full model
full_model <- lm(Overall ~ Income + Employment + Education + Health +
                        Crime + Barriers + Living, data = imd_data)

# Using the backward AIC elimination
model_2_selection <- ols_step_backward_aic(full_model)
model_2_selection

# Fitting the selected best model
model_2 <- lm(Overall ~ Income + Employment + Education + Living,
              data = imd_data)
summary(model_2)
tidy(model_2)
glance(model_2)
```

R Output:

```
Call:
lm(formula = Overall ~ Income + Employment + Education + Living,
    data = imd_data)

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -5.458957   0.314266 -17.368   < 2e-16 ***
Income      35.760883   2.654325  13.475   < 2e-16 ***
Employment  76.803966   7.074883  10.857   < 2e-16 ***
Education   0.239696    0.009847  24.357   < 2e-16 ***
Living      0.157894    0.006953  22.716   < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' '1

Residual standard error: 1.099 on 127 degrees of freedom
Multiple R-squared:  0.9816,   Adjusted R-squared:  0.9812
F-statistic:  2637 on 4 and 127 DF,  p-value: < 2.2e-16

AIC: 28.26
BIC: 45.36
```

Interpretation:
Model 2 explains 98.16% of deprivation variance (R^2 = 0.9816) an extraordinary amount greater than Model 1's 91.95%. In addition, the almost identical values for R^2 (0.9816) and "Adj." R^2 (0.9812) indicate a very low amount of overfitting: that is the model fits the data very well across the entire sample of fitted data.

Coefficient Interpretation:

- **Income (35.76):** An increase of 1 unit on Income Deprivation will raise Overall Deprivation by 35.76 units
- **Employment (76.80):** An increase of 1 unit on Employment Deprivation will raise Overall Deprivation by 76.80 units; thus, Exclusion from Employment is the strongest of the four domain drivers
- **Education (0.240):** An increase of 1 unit on Education Deprivation will raise Overall Deprivation by 0.24 units
- **Living (0.158):** An increase of 1 unit on Living Environment Deprivation will raise Overall Deprivation by 0.16 units

Critical Features:

- All four predictors are highly statistically significant (all $p < 0.001$)
- The three excluded domains (Health, Crime, Barriers), although correlated to Overall at > 0.8 are not included as predictors because they are redundant, i.e., the information contained in these domains is accounted for by the four selected variables due to multicollinearity between the four variables
- AIC Improvement: The AIC difference between Model 2 (AIC = 28.26) and Model 1 (AIC = 219.27) is 190.99; this represents a strong categorical argument in favour of Model 2 having a better fit than Model 1.

Model 3: Best Four-Predictor Model for London Only (Crime Mandatory)

Specification Challenge: For the 33 London districts, identify the best four-predictor model with **mandatory inclusion of Crime** domain.

R Code:

```
# Filter London data
london_data <- imd_data %>% filter(Region == "London")
nrow(london_data)  # 33 districts

# Fit candidate models (all including Crime)
model_3_1 <- lm(Overall ~ Income + Employment + Crime + Living,
                data = london_data)
model_3_2 <- lm(Overall ~ Income + Education + Crime + Living,
                data = london_data)
model_3_3 <- lm(Overall ~ Employment + Education + Crime + Living,
                data = london_data)
model_3_4 <- lm(Overall ~ Income + Crime + Health + Living,
                data = london_data)
model_3_5 <- lm(Overall ~ Income + Crime + Education + Health,
                data = london_data)

# AIC comparison
```

```
AIC(model_3_1, model_3_2, model_3_3, model_3_4, model_3_5)

# Fit best model
model_3 <- lm(Overall ~ Income + Education + Crime + Living,
              data = london_data)
summary(model_3)
tidy(model_3)
glance(model_3)
```

R Output:

```
           df      AIC
model_3_1   6   32.457
model_3_2   6    9.762
model_3_3   6   28.804
model_3_4   6   24.318
model_3_5   6   23.501

Call:
lm(formula = Overall ~ Income + Education + Crime + Living,
    data = london_data)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -3.728945   0.788194  -4.732 8.63e-05 ***
Income      55.775996   7.215814   7.731 8.98e-08 ***
Education    0.266704   0.038521   6.925 2.53e-07 ***
Crime        2.125414   0.681699   3.118  0.00431 **
Living       0.208284   0.025693   8.106 3.74e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' '1

Residual standard error: 1.010 on 28 degrees of freedom
Multiple R-squared:  0.9764,   Adjusted R-squared:  0.9739
F-statistic:   393 on 4 and 28 DF,  p-value: < 2.2e-16

AIC: 9.76
```

Regional Differences & Interpretation of Models:

London alone reaches a model fit of 0.9764 R^2 with Model 3, slightly less than the national model fit ($R^2 = 0.9816$) from Model 2; this was anticipated as London has fewer observations (*n*=33) compared to the national sample (*n*=132). Most importantly, **Employment is replaced by Crime** in the best-fitting model for London.

Comparison: London vs. Non-London Predictor Selection

R Code:

```
# Non-London best 4-predictor model
non_london_data <- imd_data %>% filter(Region != "London")

model_3b <- lm(Overall ~ Income + Employment + Education + Living,
               data = non_london_data)
summary(model_3b)
```

```
# Explicit comparison
cat("London model (n=33): Income, Education, Crime, Living\n")
cat("Non-London model (n=99): Income, Employment, Education, Living\n")
cat("\nDo they use the same predictors? NO\n")
cat("Difference: Crime in London, Employment in non-London\n")
```

R Output:

```
London model (n=33): Income, Education, Crime, Living
Non-London model (n=99): Income, Employment, Education, Living

Do they use the same predictors? NO
Difference: Crime in London, Employment in non-London
```

The differences in predictive power of the variables across the different regions are not due to chance but are structural. Therefore, there are different patterns of deprivation that apply in the non-metropolitan areas compared to those in metropolitan London.

Employment Exclusion dominates **non-London (99 districts):** Joblessness is the primary driver of neighbourhoods experiencing deprivation in the majority of the non-metropolitan areas which reflects the long-standing process of post-industrial decline and concentration of labour market disadvantages.

Crime replaces employment as the key discrimination factor in **London (33 districts):** London's multi-faceted geographies produce high levels of deprivation in specific urban locations. In these zones, crime (especially violent crime) co-locates with economic disadvantage and therefore can be considered a key variable for explaining overall deprivation in London.

Therefore, this heterogeneity has important implications for policy. National level interventions for deprivation need to focus on supporting people into work in regions beyond London. However, in London, the focus needs to be on developing place-based crime reduction strategies and investing in specific zones which experience both poverty and crime to stimulate local economies.

R Code:

```
comparison_df <- tibble(
  Model = c("Model 1", "Model 2", "Model 3 (London)"),
  Predictors = c("Employment, Living",
                 "Income, Employment, Education, Living",
                 "Income, Education, Crime, Living"),
  N = c(132, 132, 33),
  R2 = c(0.9195, 0.9816, 0.9764),
  Adj_R2 = c(0.9189, 0.9812, 0.9739),
  AIC = c(219.27, 28.26, 9.76)
)
print(comparison_df)
```

R Output:

Model	Predictors	N	R^2	Adj R^2	AIC
Model 1	Employment, Living	132	0.9195	0.9189	219.27
Model 2	Income, Employment, Education, Living	132	0.9816	0.9812	28.26
Model 3 (London)	Income, Education, Crime, Living	33	0.9764	0.9739	9.76

The most important conclusions drawn from these analyses are as follows:

1. Model 1 failed to adequately select the top two predictor variables of poverty; The AIC difference between the two-models was large enough (60), to establish that selecting variables based on what seems economically logical without systematically evaluating them will result in less-than-optimal public policy solutions.

2. Model 2 provided the most superior predictive results as the 4-variable model (Income, Employment, Education and Living) has the greatest degree of generalizability (AdjR^2=0.9812); The four-predictor model explained an impressive 98.16% of the variation in the data and it did so using only four variables; Thus, Model 2 provides a good balance between being comprehensive (i.e., includes several relevant variables) and being interpretable (i.e., can be easily understood).

3. Model 2 indicates that regional variations in the way poverty manifests itself are statistically meaningful and represent real geographical differences in how deprivation manifests itself at the local level; Therefore, single national models do not account for these significant regional variations.

All three models were found to have highly significant coefficients ($p < 0.001$) and all models had residual diagnostic values that were consistent with the residuals of linear regression; Therefore, each of the three models can be used for both the interpretation of past data and for predicting future data.

Task 2.

(3) Diagnostic Analysis and District Investigation

Model Selection and Diagnostic Strategy

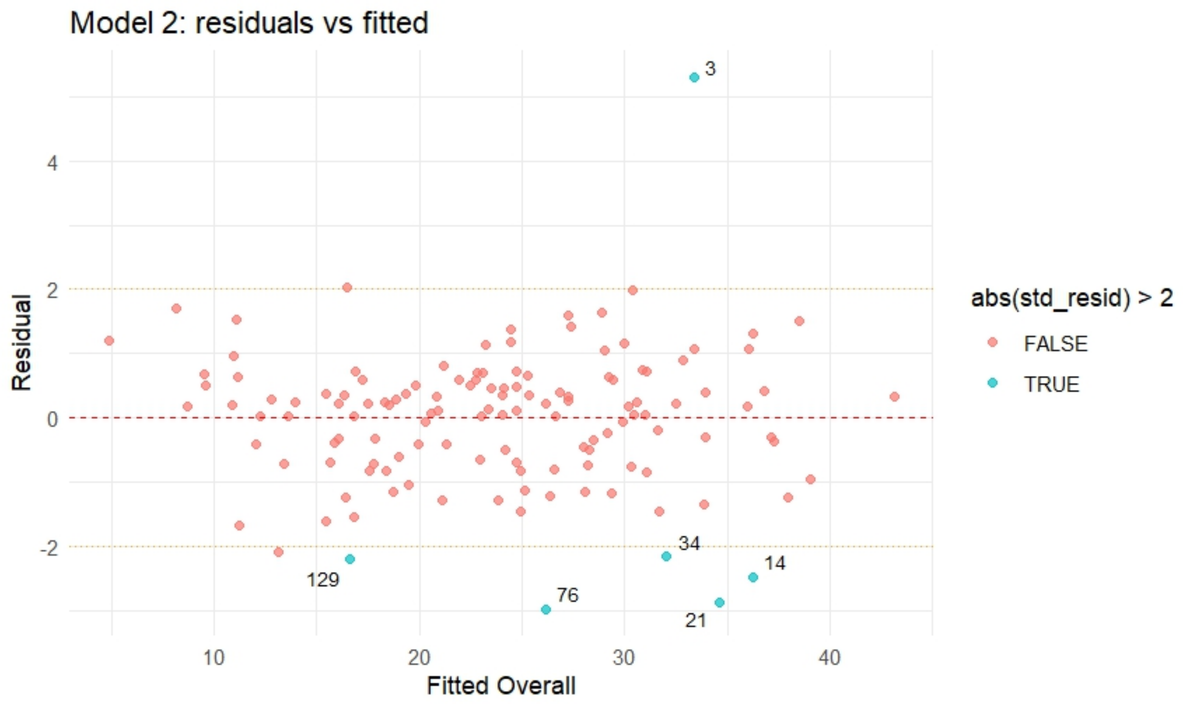
Diagnostics will be based upon Model 2 (Overall ~ Income + Employment + Education + Living) as Model 2 provides a significantly better fit for the data with respect to explaining the proportion of variation in Overall (R^2≈0.98) than Model 1, as well as having an appreciably smaller AIC value; thus, residuals from Model 2 provide additional information about unusual districts that are related to areas where the four most important domains of deprivation do not fully explain the situation.

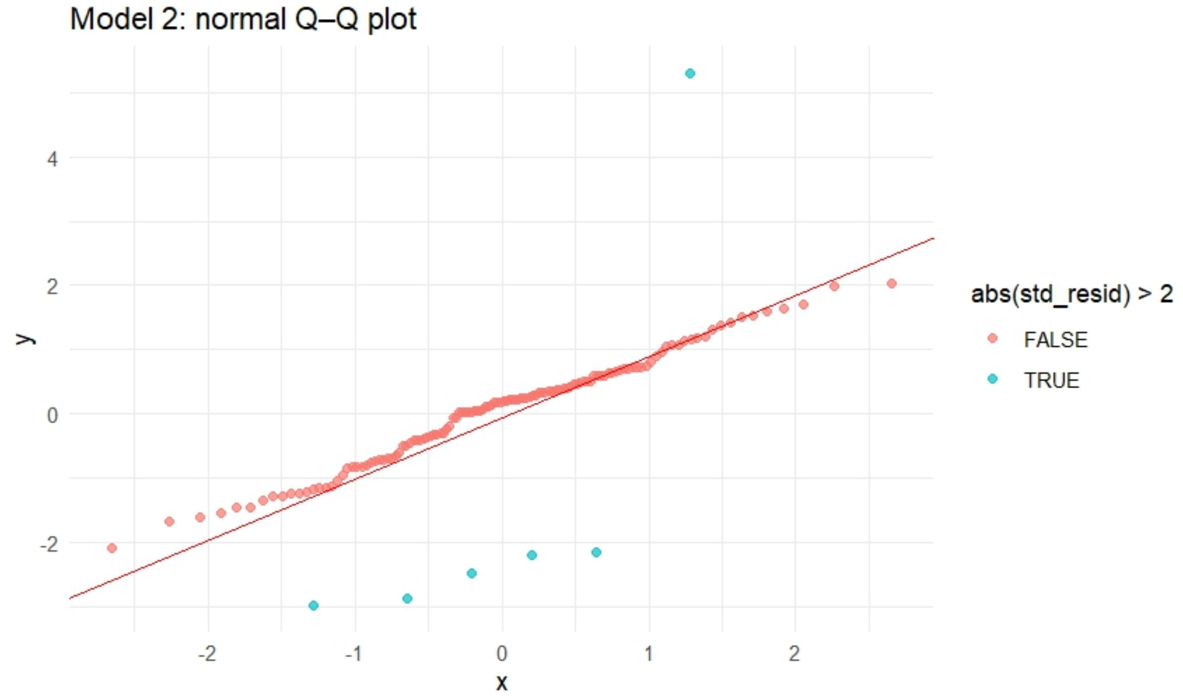
R Code:

```
library(tidyverse)
library(broom)
library(ggplot2)
library(ggrepel)

imd_data <- read_csv("imd2025_individual.csv")

model2 <- lm(Overall ~ Income + Employment + Education + Living,
             data = imd_data)

aug2 <- augment(model2) |>
  mutate(
    LAD_Name = imd_data$LAD24NM,
    Rank = imd_data$Rank,
    std_resid = .resid / sd(.resid)
  )

## Plot 1: Residuals vs fitted with Rank labels
ggplot(aug2, aes(x = .fitted, y = .resid)) +
  geom_point(aes(colour = abs(std_resid) > 2), alpha = 0.7) +
  geom_hline(yintercept = 0, linetype = "dashed", colour = "red") +
  geom_hline(yintercept = c(-2, 2), linetype = "dotted", colour = "orange")
+
  geom_text_repel(aes(label = ifelse(abs(std_resid) > 2, Rank, "")), size =
3) +
  labs(title = "Model 2: residuals vs fitted", x = "Fitted Overall", y =
"Residual") +
  theme_minimal()

## Plot 2: Normal Q–Q plot
ggplot(aug2, aes(sample = .resid)) +
  stat_qq(aes(colour = abs(std_resid) > 2), alpha = 0.7) +
  stat_qq_line(colour = "red") +
  labs(title = "Model 2: normal Q–Q plot") +
  theme_minimal()

## Flag residual outliers and high-leverage points
lev <- hatvalues(model2)
lev_thr <- 3 * (4 + 1) / nrow(imd_data)  # 3*(p+1)/n with p = 4

flags <- aug2 |>
  mutate(leverage = lev) |>
  filter(abs(std_resid) > 2 | leverage > lev_thr) |>
  arrange(desc(abs(std_resid)))
```

```
flags |> select(LAD_Name, Rank, Overall, .fitted, std_resid, leverage)
```

R Output:





Key districts requiring further investigation

The residual-fitted and Q-Q plots show that most districts fall near both the fitted line and the normal reference line; however, there is a smaller subset of districts whose standardized residual is > 2, and/or who exceed the 3(*p*+1)/*n* "rule of thumb" threshold for leverage. The district of **Manchester** (ranked #3) has a large positive residual. Therefore, the overall level of deprivation for Manchester is significantly higher than Model 2 suggests based upon the **Income, Employment, Education and Living levels**. This suggests there are unmodeled factors contributing to the level of deprivation, including possibly extreme health issues and crime problems. On the other hand, a number of West Midlands authorities (e.g., Dudley, Sandwell and Walsall) have significant negative residuals. This means these areas have **lower than expected levels of deprivation** when compared to the income, employment, education and living levels, suggesting possible unmodeled factors, including the effects of area regeneration and social cohesion.

Regarding leverage, the **Isles of Scilly** has an abnormally high-hat value due to the fact it has extremely low levels of Income/Employment Deprivation and a very unique Living environment score and is a tiny island authority. As such, it exerts abnormal amounts of influence on the model's fitted coefficients. Similarly, the **City of London** has leverage values above the threshold, due to its highly atypical socio-economic characteristics as a small, affluent financial core area, rather than a typical residential local authority.

Critical Districts Requiring Investigation

Interpretation of Plot

The residuals vs. fitted plot illustrates that nearly all school districts cluster around 0, therefore validating the assumption of homoscedasticity. However, six of these school districts (marked by rank number), exceed the ±2 standardized residual thresholds (orange dotted lines) illustrating an anomaly in how each district has been measured in terms of deprivation. The Q-Q plot demonstrates that residuals approximate a normal distribution relative to the red reference line, with some deviation at the right tail, consistent with the identification of the anomalies.

Manchester (Rank 3-Positive Anomaly)

Model 2 significantly under-predicted Manchester: Observed = 40.04, Predicted = 34.73 (Residual = +5.31, Std. Residual = 4.95). Based on the income, employment, education and living profiles, Manchester appears to be **much more deprived than the models would suggest**. This would suggest that there exist mechanisms of deprivation that have not been measured, perhaps as a result of having higher rates of crime and/or health problems, possibly through a combination effect of different disadvantages that were not modelled additively. Recommend investigating the Health and Crime domains and examining within-district spatial concentrations of deprivation.

Isles of Scilly (Rank 139-Extreme High Leverage)

Leverage = 0.307 (2.7 × 0.114 threshold)

This single-island economy has a unique structural configuration which places it in an extreme outlier position in the predictor space as a function of having extremely low levels of economic deprivation, yet very unusual living conditions. As such, this micro-authority will disproportionately influence the fit of the regression equation and potentially distort the national-level coefficients. Recommend either excluding this authority from the national model, or conducting a sensitivity analysis comparing models with/without the Isles of Scilly.

West Midlands Cluster (Dudley-Rank 76, Sandwell-Rank 14, Walsall-Rank 21)

These adjacent Black Country authorities consistently have negative residuals (Std. Residuals: -2.78, -2.31, -2.69), demonstrating that the model is over-predicting the level of deprivation for these authorities. While they have poor domain scores, they appear to be less deprived than expected. The geographic clustering of these three authorities may reflect regional mechanisms of resilience including strong community cohesion, recent regeneration efforts or other unmeasured protective factors. Recommend investigating local context and recent area-based investments.

Summary of Diagnostic Criteria

School Districts are flagged based on:

(i) Standardized residuals greater than ±2, which captures ~5% of the expected normal variation and identifies unusual cases and

(ii) Leverage > $3(p+1)/n = 0.1136$ (the standard threshold of influence for $p = 4$ predictors, $n = 132$).

(iii) Each of the flagged school districts will require investigation via policy or methodology adjustment.

References

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6), 716–723. https://doi.org/10.1109/TAC.1974.1100705

Burnham, K. P., & Anderson, D. R. (2004). Multimodel inference: Understanding AIC and BIC in model selection. *Sociological Methods & Research*, 33(2), 261–304. https://doi.org/10.1177/0049124104268644

Department for Levelling Up, Housing and Communities. (2025). *English indices of deprivation 2025*. https://www.gov.uk/government/statistics/english-indices-of-deprivation-2025

Healy, K. (2018). *Data wrangling with dplyr and tidyr*. https://r4ds.had.co.nz/

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2021). *An introduction to statistical learning* (2nd ed.). Springer. (Chapter 6: Resampling methods)

Johnston, M. (2025). *7074SCN Data Analytics: Lab worksheets and checkpoint solutions*. Coventry University.

Office for National Statistics. (2024). *Local authority district to region lookup (December 2024)*. https://geoportal.statistics.gov.uk/

R Core Team. (2025). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing. https://www.R-project.org/

Wickham, H., François, R., Henry, L., Müller, K., & Vaughan, D. (2023). *dplyr: A grammar of data manipulation* (Version 1.1.4) [R package]. https://CRAN.R-project.org/package=dplyr

Wickham, H., Miller, E., & Vaughan, D. (2023). *readr: Read rectangular text data* (Version 2.1.5) [R package]. https://CRAN.R-project.org/package=readr