# Introduction

## Context and background

## Aim and importance

## Outline

# Materials and methods

## Materials

## Methods

### Experimental infection protocol

### Statistical analysis

## Models :

**Bayes Formula:**

: Random variable (real observation data)

* + = Number of positives per experiment 🡪 Likelihood = Binomiale.
  + = Infectious status (0 / 1) 🡪 Likelihood = Bernoulli.

**Model 1 : Baseline model 🡪**

* number of susceptible individuals per experiment.
* , the probability of getting a positive after the period , where represents the period of the experiment in minutes.
* : constant probability.

**Model 2 : Segmented model 🡪**

* Filtering distance: 1 m, 1.5 m, and 2 m (based on expert opinion).
* , where represents the cumulative time spent between an individual and the seeder under a fixed distance d.

**Model 3 : Enveloppe model 🡪**

* Filtering distance: 1 m, 1.5 m, and 2 m (based on expert opinion).
* Identifiying “enveloppes” (clusters of consecutive time points with stable distances) in each distance segment.
* , r is the median distance of the enveloppe j for individual i.
* maximum probability when distance is null.
* : decay rate.

.

## Prior distributions.

Beta distribution: Uninformative (semi\_informative)

Log Normal distribution (for positive rates)

### Posterior distribution inference

* we utilized the No-U-Turn Sampler (NUTS), an extension of the Hamiltonian Monte Carlo (HMC) method within the Markov chain Monte Carlo (MCMC) bayesian framework (Hoﬀman and Gelman, 2014)
* Sampling mplemented on an Intel Core i5 10th generation CPU to accelerate the sampling process.
* All implementations were on Python.
* In our analysis we used the pymc.sampling.jax.sample\_blackjax\_nuts function from the PyMC library, which is a popular tool for probabilistic programming in Python (Abril-Pla et al., 2023).
* This function integrated BlackJAX to provide efficient and fast sampling using NUTS, thus leveraging the speed benefits of JAX.
* Draws = 10000: The number of samples drawn from the posterior distribution; it is the number of iterations for which the sampler will run.
* Tune = 1000: The number of tuning steps before the actual sampling begins. Tuning helps adjust the sampler’s parameters to improve efficiency and convergence.
* Chains=10: The number of independent chains run by the sampler. Multiple chains help ensure that the sampling process explores the posterior distribution thoroughly.
* Target\_accept = 0.95: The target acceptance rate for the sampler, a value between zero and 1, typically set near 1. This parameter influences the step size of the sampler, balancing between exploration and efficiency.

### Model convergence

Diagnostics such as the Effective Sample Size (ESS), Gelman-Rubin convergence diagnostic (), were used to assess the convergence and mixing properties of chains.

In both the segmented-model and the envelope-model the chains mixed well with all ESS > 10000, all > 1.

# Results

## Experimental infection results

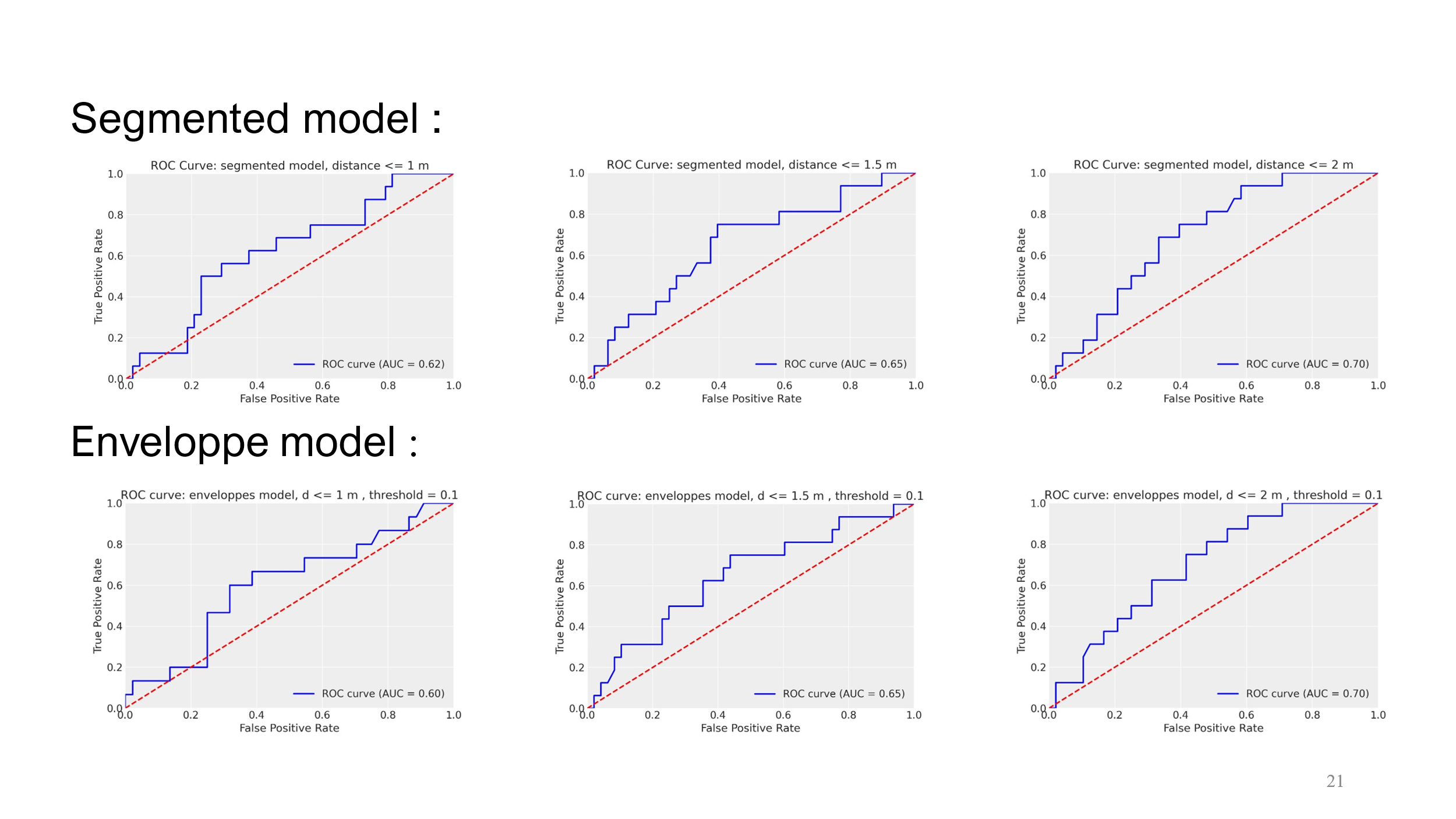
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Experiment | Duration | Surface | Date | Batch size  (seeder included) | Positive  (seeder excluded) |
| 1 | 1h | all | 27/06/2021 | 7 | 0 |
| 6h | all | 27/06/2021 | 7 | 0 |
| 24h | all | 27/06/2021 | 7 | 0 |
| 2 | 24h | all | 26/08/2021 | 7 | 1 |
| 44h | all | 27/08/2021 | 7 | 4 |
| 3 | 24h | all | 02/03/2022 | 6 | 3 |
| 4 | 24h | all | 28/04/2022 | 7 | 0 |
| 48h | all | 29/04/2022 | 6 | 4 |
| 5 | 24h | all | 14/06/2022 | 7 | 1 |
| 48h | all | 15/06/2022 | 7 | 3 |
| 6 | 8h | reduced | 25/07/2022 | 7 | 0 |
| Total |  | | | 75 | 16 |

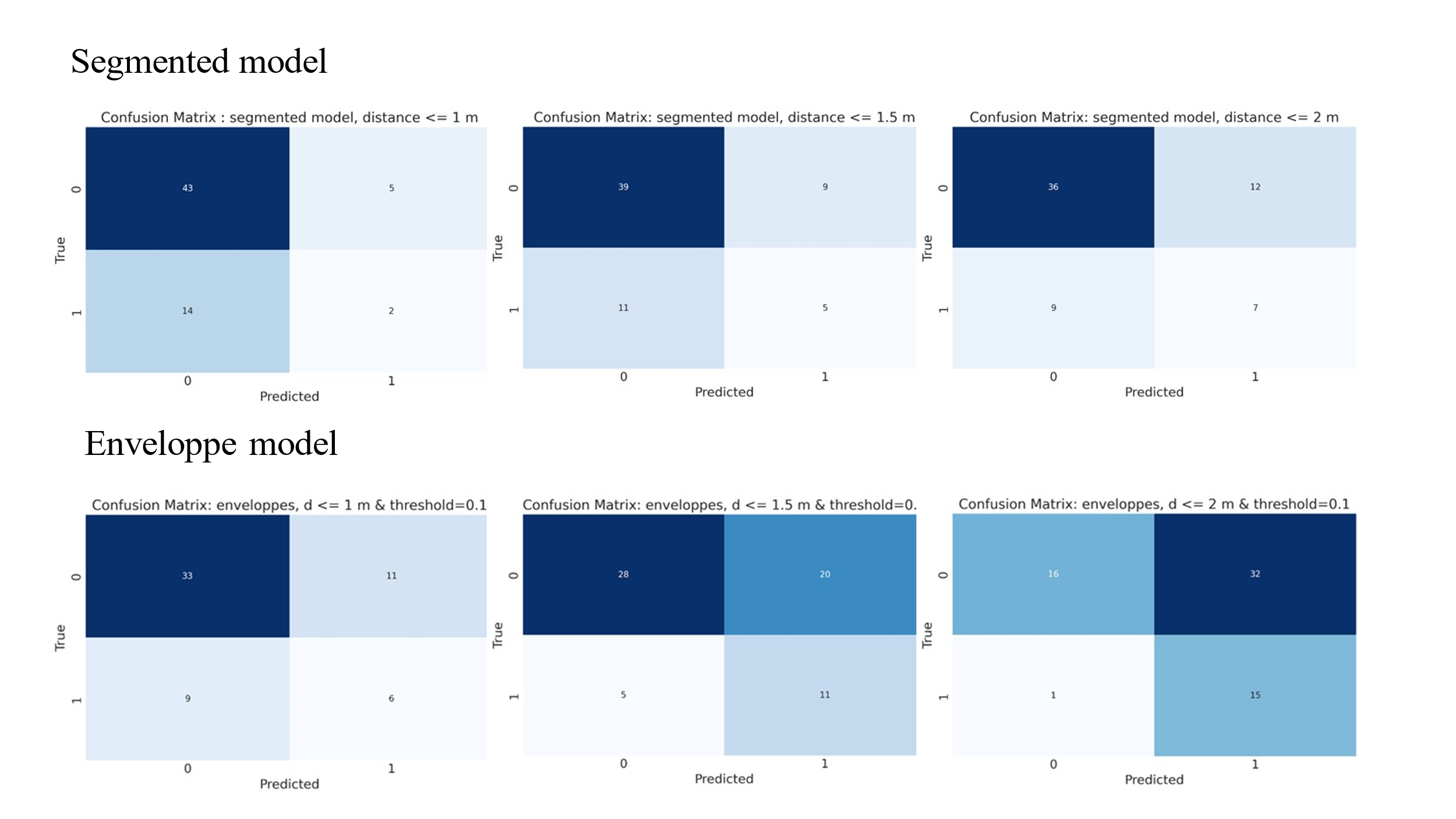
|  |  |  |  |
| --- | --- | --- | --- |
| **Characteristic** | **0, N = 35 1** | **1, N = 12 1** | **p-value 2** |
| Duration Class |  |  | < 0.001 |
| < 24 | 12 (100%) | 0 (0%) |  |
| 24 | 20 (83%) | 4 (17%) |  |
| > 24 | 3 (27%) | 8 (73%) |  |
| 1 n (%)  2 Fisher’s exact test | | | |

## Parameter estimation results

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Models | Distance  cut-off | Parameters | Enveloppe  Threshold | Mean | SD | 95% HDI  Lower – Upper |
| Model 1 | - |  | - | 0.00023 | 0.00006 | 0.00013 – 0.00034 |
| Model 2 | 1 m |  | - | 0.0049 | 0.0012 | 0.0027 – 0.0073 |
| 1.5 m |  | - | 0.0031 | 0.0008 | 0.0017 – 0.0046 |
| 2 m |  | - | 0.002 | 0.0005 | 0.0011 – 0.003 |
| Model 3 | 1 m |  | 0.1 | 0.0078 | 0.0020 | 0.0041 – 0.0116 |
| 0.2 | 0.0081 | 0.0020 | 0.0043 – 0.0122 |
| 0.3 | 0.0086 | 0.0022 | 0.0045 – 0.0128 |
|  | 0.1 | 0.3708 | 0.0374 | 0.2994 – 0.4448 |
| 0.2 | 0.3707 | 0.0375 | 0.3002 – 0.4468 |
| 0.3 | 0.3708 | 0.0375 | 0.299 – 0.4451 |
| 1.5 m |  | 0.1 | 0.0044 | 0.0011 | 0.0024 – 0.0066 |
| 0.2 | 0.0045 | 0.0011 | 0.0024 – 0.0066 |
| 0.3 | 0.0054 | 0.0013 | 0.0029 – 0.0080 |
|  | 0.1 | 0.3710 | 0.0374 | 0.3002 – 0.4455 |
| 0.2 | 0.3711 | 0.0374 | 0.3002 – 0.4452 |
| 0.3 | 0.3710 | 0.0374 | 0.2992 – 0.4445 |
| 2 m |  | 0.1 | 0.0034 | 0.0008 | 0.0019 – 0.0051 |
| 0.2 | 0.0035 | 0.0009 | 0.0019 – 0.0052 |
| 0.3 | 0.0036 | 0.0009 | 0.0019 – 0.0053 |
|  | 0.1 | 0.3712 | 0.0375 | 0.3002 – 0.4463 |
| 0.2 | 0.3712 | 0.0374 | 0.2997 – 0.4456 |
| 0.3 | 0.3712 | 0.0374 | 0.2987 – 0.4445 |

## Model comparison





|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Model** | **Distance** | **TP** | **TN** | **FP** | **FN** | **AUC** |
| Segmented model | 1 m | 2 | **43** | **5** | 14 | 0.62 |
| 1.5 m | 5 | 39 | 9 | 11 | 0.65 |
| 2 m | 7 | 36 | 12 | 9 | **0.70** |
| Enveloppe model | 1 m | 6 | 33 | 11 | 9 | 0.60 |
| 1.5 m | 11 | 28 | 20 | 5 | 0.65 |
| 2 m | **15** | 16 | 32 | **1** | **0.70** |

A graph of infection

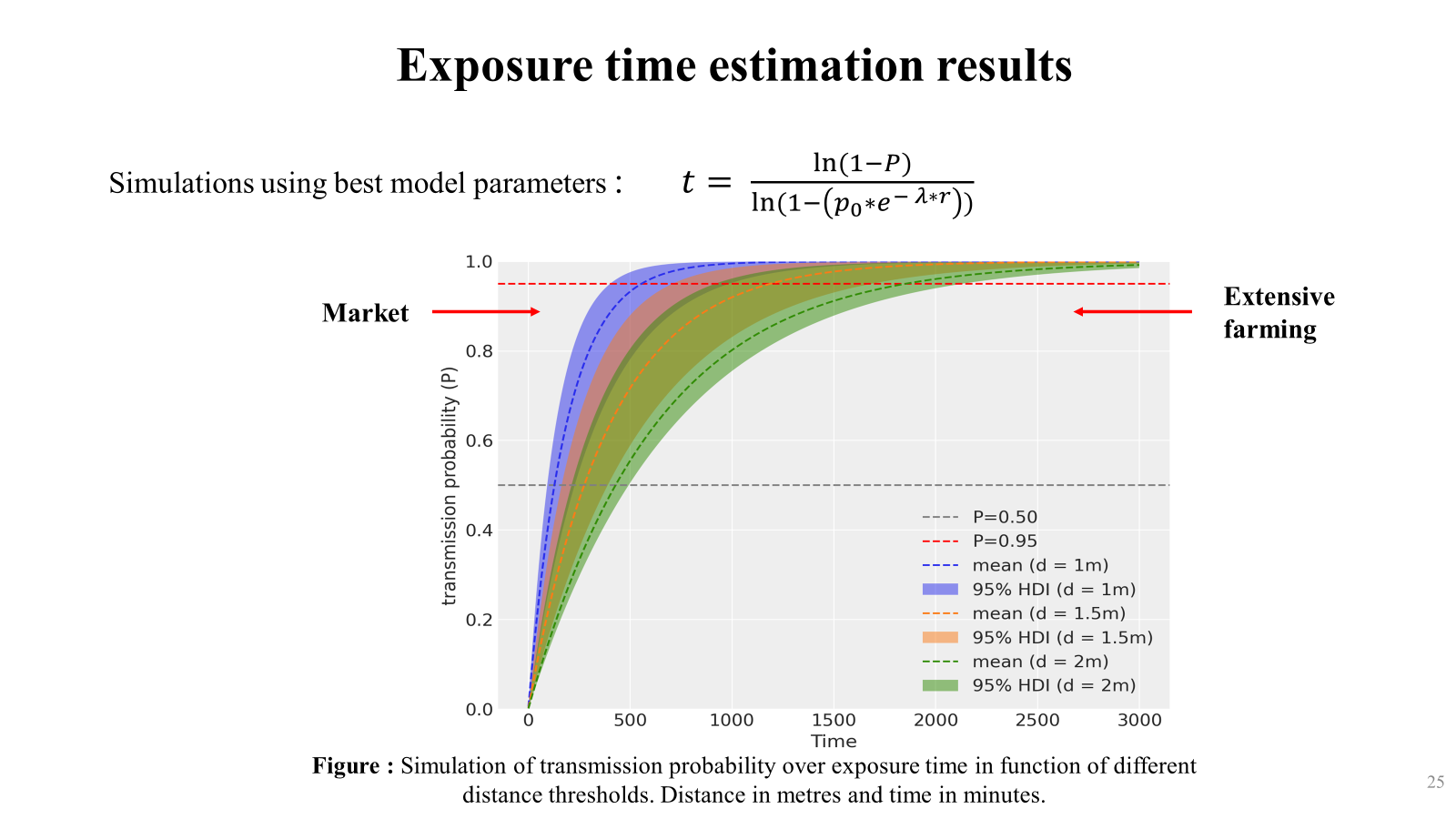
Description automatically generated

Une image contenant Tracé, diagramme, texte, ligne

Description générée automatiquement

Une image contenant texte, Tracé, diagramme, ligne

Description générée automatiquement



# References

Abril-Pla, O., Andreani, V., Carroll, C., Dong, L., Fonnesbeck, C.J., Kochurov, M., Kumar, R., Lao, J., Luhmann, C.C., Martin, O.A., Osthege, M., Vieira, R., Wiecki, T., Zinkov, R., 2023. PyMC: a modern, and comprehensive probabilistic programming framework in Python. PeerJ Comput. Sci. 9, e1516. https://doi.org/10.7717/peerj-cs.1516

Hoﬀman, M.D., Gelman, A., 2014. The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo 15, 1593–1623.