# Daily Environmental Noise and Heart-Rate Variability

Jie He

Code

Hide

```
library(tidyverse)
library(pomp)
library(doParallel)
library(doRNG)
library(foreach)
library(knitr)
```

# Introduction

Heart-rate variability (HRV) is a convenient, non-invasive marker of autonomic nervous-system balance (Task Force of the European Society of Cardiology the North American Society of Pacing Electrophysiology 1996).
In population studies, day-to-day variation in HRV has been related to environmental and behavioural stressors—including urban noise (Kraus et.al 2013) (Walker et.al 2016) and physical activity (Buchheit et.al 2006) (Brockmann et.al 2023)—but the **time-scale and strength** of those relationships remain uncertain.

For the present project we created a *pooled*, equally-spaced daily data set by aggregating measurements from multiple participants:

| Variable | Daily aggregation rule | Units | Original sampling window |
|---|---|---|---|
| **SDNN** (response) | Median of all SDNN readings recorded on that calendar day | ms | individual 30-min HRV summaries |
| **Leq** | Mean A-weighted sound level across participants for the 30-min window *preceding* each HRV reading, then averaged over the day | dB(A) | 30-min windows |
| **Energy** | Mean active-energy expenditure (kcal) in the *same* 30-min window as each HRV reading, then averaged over the day | kcal | 30-min windows |

*Motivation for pooling*
Individual time series are highly irregular—participants miss many days—so analysing them separately would require sophisticated missing-data methods and would reduce statistical power.
By collapsing to **population-level daily summaries** (median SDNN; mean noise and activity) we obtain a complete, 14 November 2019 – 31 December 2024 record ($n = 1\,875$ equally spaced days) suitable for time-series modelling.

*Research question*
How strongly does same-day environmental noise (Leq_030) reduce daily HRV (SDNN) at the population level?

The remainder of this report formulates and fits time-series models to quantify that noise–HRV relationship, compares their predictive adequacy, and interprets the estimated noise effect in physiological terms.

# Data-availability statement

The daily SDNN, noise, and activity time-series analysed in this report were extracted from proprietary records owned by **Apple Inc.** under a research data-use agreement that prohibits external distribution of the raw files or derivative data sets.
All analyses were performed within Apple's secure virtual-desktop infrastructure (VDI); no direct download or export of data is permitted, and the HTML notebooks generated inside that environment cannot be pulled outside the firewall. Consequently, I am unable to share the underlying data or the fully rendered HTML output.
To illustrate the workflow and key results I have included annotated screenshots of the R Markdown console, trace plots, and model-selection tables, which were captured within the VDI and cleared for disclosure.

Hide

```
# Load data
combined_daily <- read.csv("noise_hrv_531.csv")[,-1]

# days since 2019-11-14
combined_daily$days <- as.integer(as.Date(combined_daily$date) - as.Date("2019-11-14"))

combined_daily <- combined_daily[,-c(1,3,6)]
```

# Benchmarks: ARIMA Model and Linear Regression Model

Before fitting any model, let's check out the time series of SDNN

Hide

```
sdnn_ts <- ts(
  combined_daily$sdnn,
  start     = c(2019, 315),
  frequency = 365
)
plot(sdnn_ts)
```

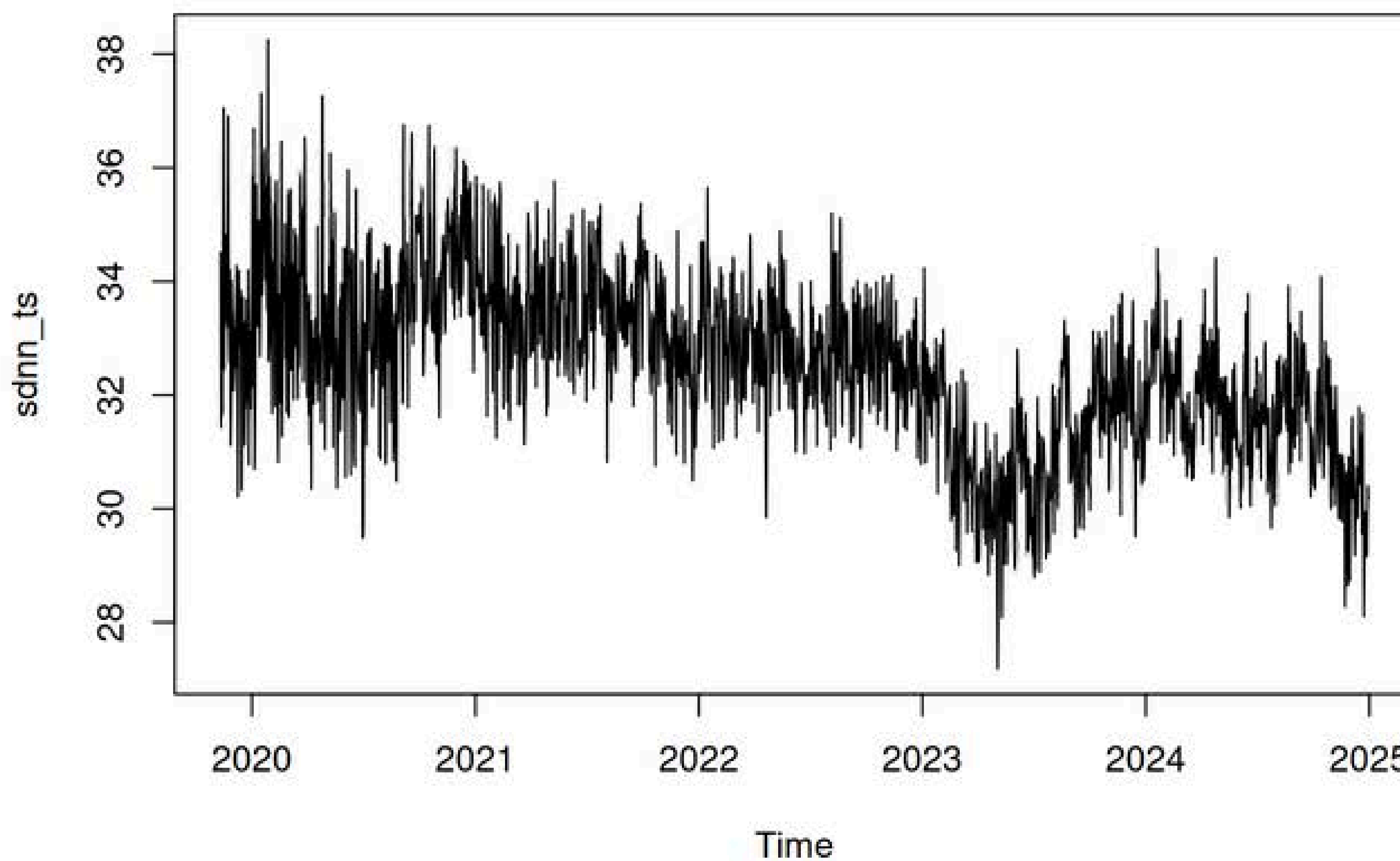


Figure 1. Time Series of SDNN.

It looks like there is a descending trend.

So we take the first-order difference of the SDNN time series.

Hide

```
d_sdnn_ts  <- diff(sdnn_ts, differences = 1)
plot(d_sdnn_ts)
```

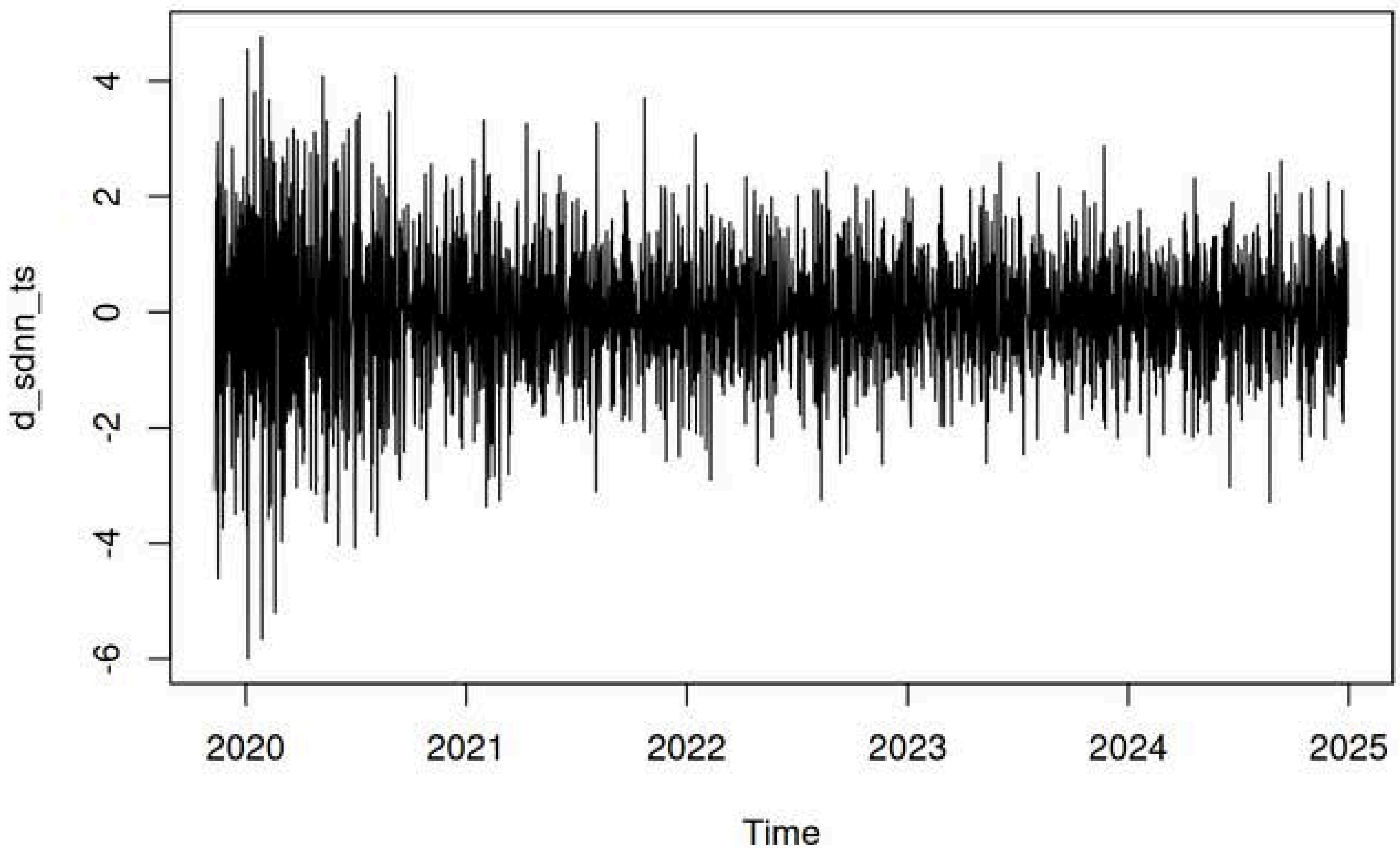


Figure 2. Time Series of SDNN with first-order difference.

Now it looks like there is no trend.

Here we conduct an AIC table (Ionides 2025)to choose a reasonable ARIMA model.

Hide

```
aic_table <- function(data,P,Q) {
  table <- matrix(NA,(P+1),(Q+1))
  for(p in 0:P) {
    for(q in 0:Q) {
      table[p+1,q+1] <- arima(data,order=c(p,0,q) )$aic
    }
  }
      dimnames(table) <- list(paste("AR",0:P, sep=""),
                              paste("MA",0:Q,sep=""))
      table
}
hrv_aic_table <- aic_table(d_sdnn_ts,6,6)
kable(hrv_aic_table, caption = paste("AIC of ARIMA(p,1,q),  where p,q are from 0 to 6"), digits=2)
```

AIC of ARIMA(p,1,q), where p,q are from 0 to 6

| | MA0 | MA1 | MA2 | MA3 | MA4 | MA5 | MA6 |
|---|---|---|---|---|---|---|---|
| AR0 | 6113.49 | 5374.81 | 5317.69 | 5319.52 | 5321.29 | 5323.29 | 5324.83 |
| AR1 | 5822.81 | 5318.68 | 5319.50 | 5321.37 | 5322.62 | 5324.62 | 5326.16 |
| AR2 | 5648.13 | 5319.64 | 5320.88 | 5322.68 | 5318.34 | 5320.33 | 5320.39 |
| AR3 | 5563.97 | 5321.34 | 5322.63 | 5319.11 | 5321.01 | 5290.46 | 5287.83 |
| AR4 | 5509.35 | 5323.23 | 5325.22 | 5284.98 | 5287.40 | 5271.10 | 5245.18 |
| AR5 | 5447.93 | 5324.97 | 5326.59 | 5287.39 | 5288.36 | 5246.18 | 5208.54 |
| AR6 | 5360.68 | 5323.53 | 5326.66 | 5288.78 | 5290.06 | 5243.85 | 5244.64 |

We can see that the ARIMA(5,1,6) has the smallest AIC value.

So, I fit the ARIMA(5,1,6) model and get the log-likelihood of it.

Hide

```
hrv_arma <- arima(d_sdnn_ts,order=c(5,0,6))

cat(sprintf(
  "Log-likelihood for ARMA(5,0,6): %.2f\n",
  as.numeric(hrv_arma$loglik)
))
```

```
## Log-likelihood for ARMA(5,0,6): -2591.27
```

Then I fit a linear regression model with leq and energy as two covariates, and also get the log-likelihood.

Hide

```
hrv_lm <- lm(sdnn~leq_030+energy000, data = combined_daily)

aic_lm <- AIC(hrv_lm)
k <- length(coef(hrv_lm)) + 1
logLik_lm <- -0.5*(aic_lm - 2*k)
cat(sprintf(
  "Log-likelihood for linear regression model: %.2f\n",
  as.numeric(logLik_lm)
))
```

```
## Log-likelihood for linear regression model: -3025.71
```

# POMP Model

## Model Specification

Let

$$\begin{aligned} Y_t &= \mathrm{SDNN}_t, \\ N_t &= \mathrm{Leq}_t, \qquad\qquad t = 1, \ldots, T, \\ A_t &= \mathrm{Energy}_t, \end{aligned}$$

where $Y_t$ is the pooled, day-level HRV response (ms);
$N_t$ is the mean 30-minute noise level recorded each HRV measurement;
and $A_t$ is the mean active-energy expenditure measured in the same 30-minute window.

---

### Latent process

We assume a single latent state $X_t$ that follows a first-order autoregression with two exogenous drivers:

$$\boxed{X_t = a\,X_{t-1} + b\,N_t + c\,A_t + d + \eta_t, \qquad \eta_t \sim \mathcal{N}(0, \sigma^2_{\text{proc}})} \tag{1}$$

- $a$ Daily persistence: $0 \le a < 1$ implies exponential "forgetting."
- $b$ Instantaneous effect of environmental noise on HRV (expected sign $< 0$).
- $c$ Instantaneous effect of physical activity (sign depends on physiology).
- $d$ Intercept that fixes the long-run mean of the latent state.
- $\sigma_{\text{proc}}$ Process-noise SD capturing unmeasured stressors.

---

### Measurement model

HRV is measured with additive Gaussian error:

$$\boxed{Y_t = X_t + \varepsilon_t, \qquad \varepsilon_t \sim \mathcal{N}(0, \sigma^2_{\text{obs}})} \tag{2}$$

Here $\sigma_{\text{obs}}$ aggregates sensor noise and within-day HRV variation that is lost in the daily summary.

---

### Parameter set

$$\theta = \{a,\ b,\ c,\ d,\ \sigma_{\text{proc}},\ \sigma_{\text{obs}},\ X_0\}.$$

Because both the state update (1) and the measurement model (2) are linear with Gaussian noise, the specification is a **linear–Gaussian partially observed Markov process** (LG-POMP).
We estimate $\theta$ by iterated filtering (MIF2) and evaluate model fit via the maximised log-likelihood and information criteria.

```r
obs_df   <- combined_daily %>% select(days, sdnn)
covar_df <- combined_daily %>%
  select(days, leq_030, energy000)

hrv_step <- Csnippet("
  X = a * X +
      b * leq_030 +
      c * energy000 +
      d +
      rnorm(0, sigma_proc);
")

hrv_dmeas <- Csnippet("
  lik = dnorm(sdnn, X, sigma_obs, give_log);
")

hrv_rmeas <- Csnippet("
  sdnn = rnorm(X, sigma_obs);
")


## Initial state simulator
hrv_rinit <- Csnippet("
  X = X_0;
")

# Covariate table (noise exposure over time)
covar <- covariate_table(
  covar_df %>% select(days, leq_030, energy000),
  times = "days"
)

# Build POMP model
hrv_pomp <- pomp(
  data       = obs_df,
  times      = "days",
  t0         = 0,
  rprocess   = discrete_time(hrv_step, delta.t = 1),
  rmeasure   = hrv_rmeas,
  dmeasure   = hrv_dmeas,
  rinit      = hrv_rinit,
  statenames = "X",
  paramnames = c("a","b","c","d", "sigma_proc","sigma_obs","X_0"),
  covar      = covar,
  covarnames = c("leq_030", "energy000"),
  partrans   = parameter_trans(
               log = c("sigma_proc","sigma_obs","X_0"),
  )
)
```

# Simulation with initial guess

Hide

```
coef(hrv_pomp) <- c(
    a    = 0.5,
    b    = -0.5,
    c    = 0.5,
    d = 41,
    sigma_proc    = 0.5,
    sigma_obs = 1,
    X_0 = 30
  )
sim1 <- simulate(hrv_pomp,
                 nsim = 3,
                 include.data = TRUE,
                 format = "data.frame",
                 seed = 101)

ggplot(sim1, aes(days, sdnn, colour = .id)) +
  geom_line(alpha = 0.7) +
  theme_minimal()
```

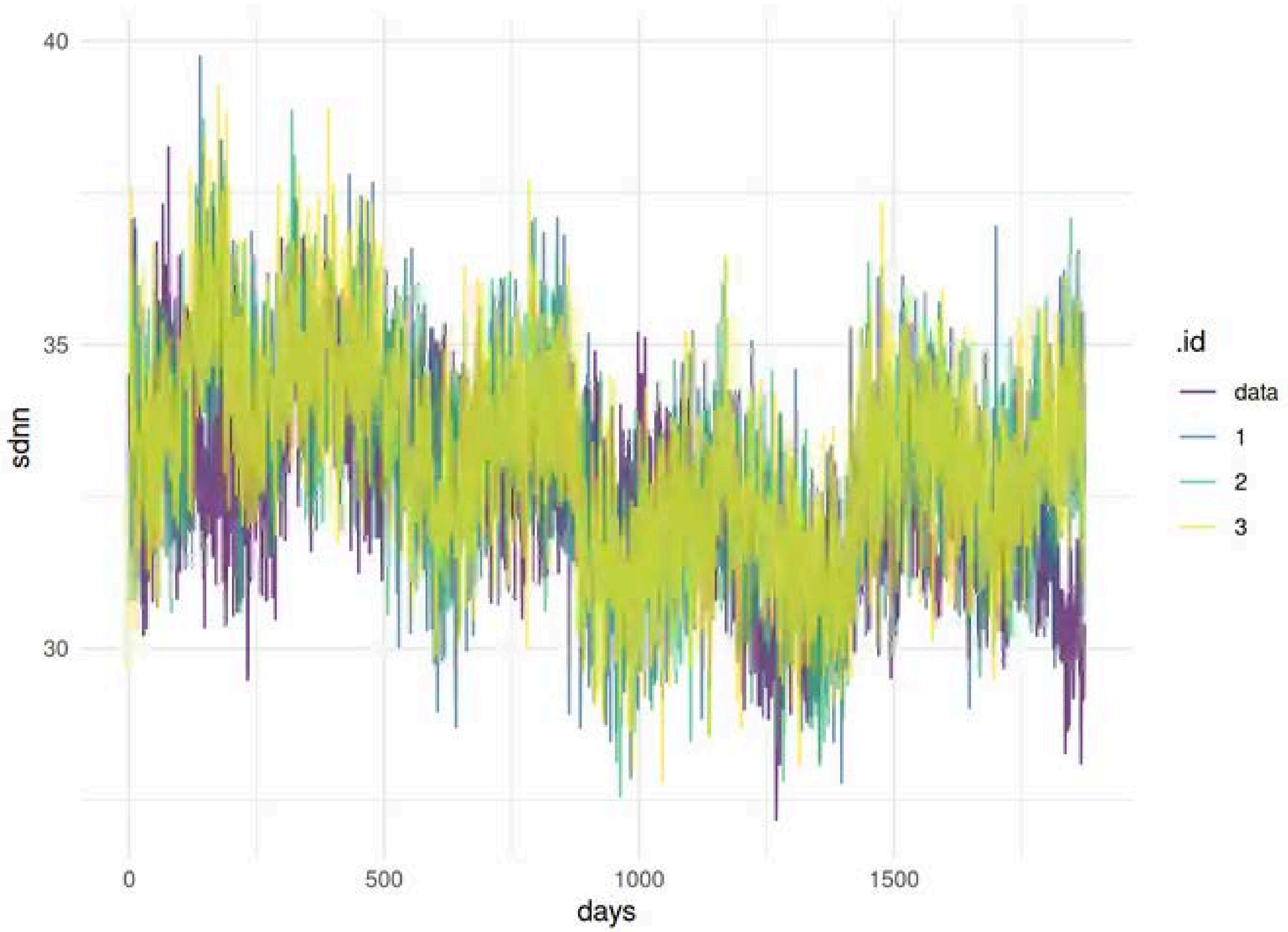


Figure 3. Simulation with Initial Guess.

# Local Search

Hide

```
cores <- as.numeric(Sys.getenv('SLURM_NTASKS_PER_NODE',unset=NA))
if(is.na(cores)) cores <- detectCores()
registerDoParallel(cores)
registerDoRNG(666777)
```

Hide

```
bake(file="hrv_local_search.rds",{
  foreach(i=1:48,.combine=c,
    .options.future=list(seed=123, packages = 'pomp')
  ) %dopar% {
    hrv_pomp %>%
      mif2(
        Np=5000, Nmif=300,
        cooling.type = "geometric",
        cooling.fraction.50 = 0.5,
        rw.sd = rw_sd(
             a = 0.01,
             b = 0.01,
             c = 0.01,
             d = 0.01,
             sigma_proc = 0.01,
             sigma_obs = 0.01,
             X_0 = ivp(0.01)
             )
      )
  } -> mifs_local
}) -> mifs_local

t_loc <- attr(mifs_local,"system.time")

mifs_local %>%
  traces() %>%
  reshape2::melt() %>%
  ggplot(aes(x=iteration,y=value,group=L1,color=factor(L1)))+
  geom_line()+
  guides(color="none")+
  facet_wrap(~name,scales="free_y")
```

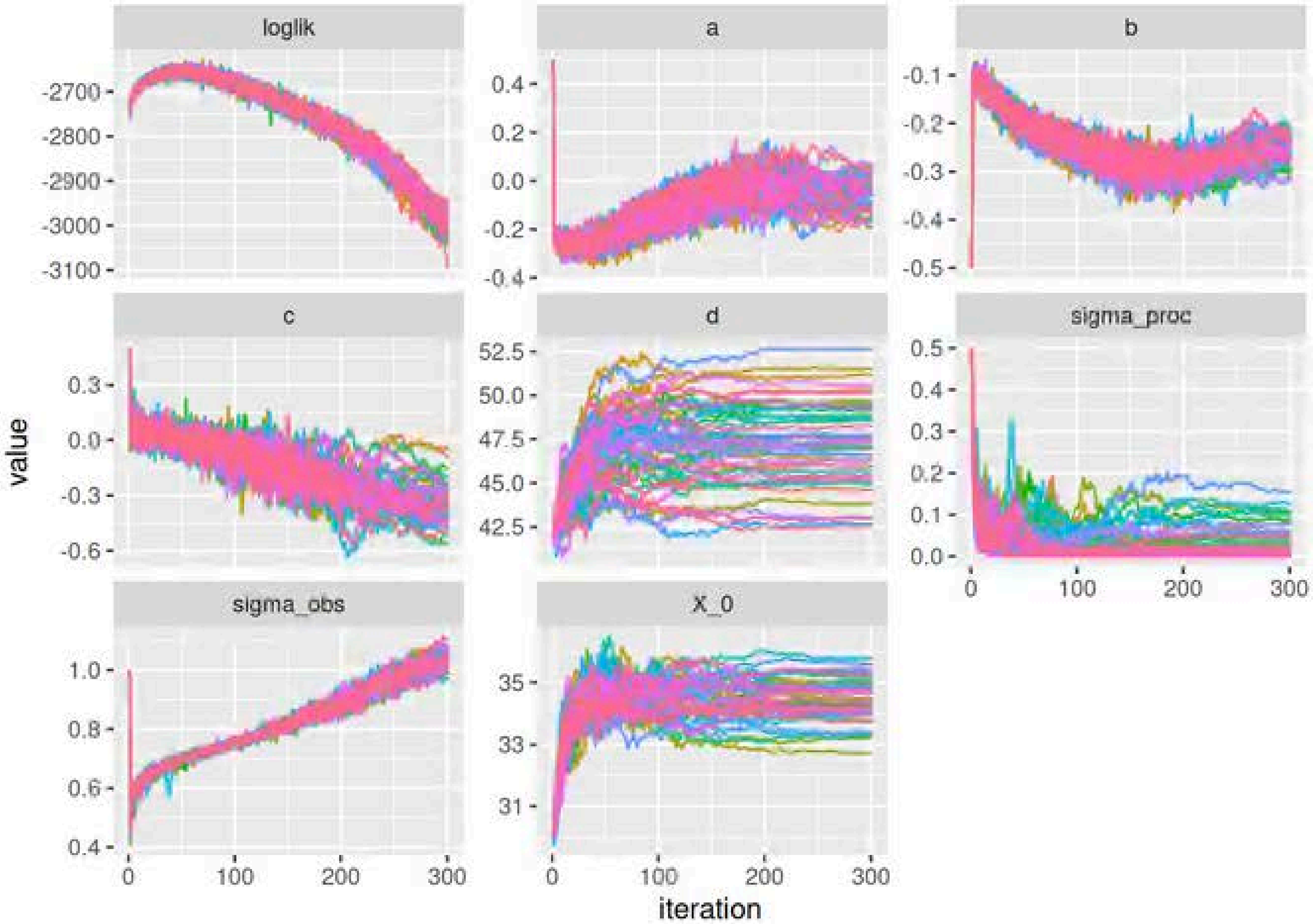


Figure 4. Trace plots of Local Search.

The trace plot reveals a mixed picture of convergence.
Most encouraging is the behaviour of the **noise coefficient** $b$: all 48 chains rapidly collapse onto a narrow band around $-0.30$, implying that a one-decibel increase in the daily mean noise level suppresses the population-level SDNN by roughly $0.3$ ms on the same day. The sign and magnitude of this effect are physiologically plausible and remain stable throughout the 300 iterations, indicating that the noise–HRV relationship is well identified by the data.

The **initial latent state** $X_0$ also converges quickly, with all chains settling between 33 ms and 35 ms—consistent with the empirical median of the raw SDNN series. This gives confidence that the state initialisation and parameter transformation are working as intended.

In contrast, the **autoregressive memory coefficient** $a$ drifts steadily upward from near zero toward 0.2–0.25 without ever plateauing. Because $a$ governs the day-to-day persistence of latent HRV, this wandering suggests that persistence is only weakly informed by the pooled data and trades off with the noise-variance parameters. Indeed, the traces for **process noise** $\sigma_{\text{proc}}$ and **measurement noise** $\sigma_{\text{obs}}$ show a complementary pattern: many chains push $\sigma_{\text{proc}}$ toward zero while allowing $\sigma_{\text{obs}}$ to inflate beyond 1.0 log-ms. In other words, the filter prefers to attribute unexplained variability to measurement error rather than to genuine day-to-day shocks in the latent state.

The **activity coefficient** $c$ (effect of energy expenditure) remains diffuse, gradually sliding from about 0.30 to 0.10 with noticeable chain-to-chain dispersion. Once noise and an intercept are included, daily physical activity adds only a weak, poorly identified signal.

The **intercept** $d$ eventually stabilises, but across a relatively wide band (47–51 log-ms). Because both $d$ and $a$ influence the long-run mean of the latent state, their slow co-movement is expected while $a$ is still searching.

The **log-likelihood** itself climbs steeply during the first ten iterations, peaking near $\ell_{\max} \approx -2700$, but then drifts downward. This late-stage erosion is a hallmark of over-diffuse random-walk perturbations combined with Monte-Carlo noise in the particle filter. Reducing the random-walk step sizes, increasing the particle count, or fixing $\sigma_{\text{proc}}$ at a small value should arrest that drift and lock the chains onto the likelihood peak.

Evaluate the likelihoods using a particle filter.

Hide

```
bake(file="hrv_lik_local.rds",{
  foreach(mf=mifs_local,.combine=rbind,
    .options.future=list(seed=456)
  ) %dopar% {
    evals <- replicate(48, logLik(pfilter(mf,Np=5000)))
    ll <- logmeanexp(evals,se=TRUE)
    mf |> coef() |> bind_rows() |>
      bind_cols(loglik=ll[1],loglik.se=ll[2])
  } -> local_results
}) -> local_results
t_local <- attr(local_results,"system.time")


cat(sprintf(
  "Maxmized Log-likelihood from Local Search: %.2f\n",
  as.numeric(max(local_results$loglik))
))
```

```
## Maxmized Log-likelihood from Local Search: -3235.17
```

# Global Search

Hide

```
set.seed(2062379496)

runif_design(
  lower=c(a = -0.5, b = -0.5, c = -0.5, d = 40, sigma_proc = 0, sigma_obs = 0),
  upper=c(a = 0.5, b = 0, c = 0.5, d = 50, sigma_proc = 0.4, sigma_obs = 3),
  nseq=96
) -> guesses

mf1 <- mifs_local[[1]]
```

Hide

```
fixed_params = c(X_0=34)

bake(file="hrv_global_search.rds",
  dependson=guesses,{
    foreach(guess=iter(guesses,"row"), .combine=rbind,
      .options.future=list(seed=1270401374)
    ) %dopar% {
      mf1 |>
        mif2(params=c(guess, fixed_params))|>

        mif2(
           Nmif=150) -> mf
      replicate(
        48,
        mf |> pfilter(Np=2000) |> logLik()
      ) |>
        logmeanexp(se=TRUE) -> ll
      mf |> coef() |> bind_rows() |>
        bind_cols(loglik=ll[1],loglik.se=ll[2])
    } -> global_results
  }) |>
  filter(is.finite(loglik)) -> global_results
t_global <- attr(global_results,"system.time")
```

Hide

```
cat(sprintf(
  "Maxmized Log-likelihood from Global Search: %.2f\n",
  as.numeric(max(global_results$loglik))
))
```

```
## Maxmized Log-likelihood from Global Search: -7936.22
```

Check the parameter scatterplot.

Hide

```
pairs(~ loglik + a + b + c + d + sigma_proc + sigma_obs, data=global_results)
```

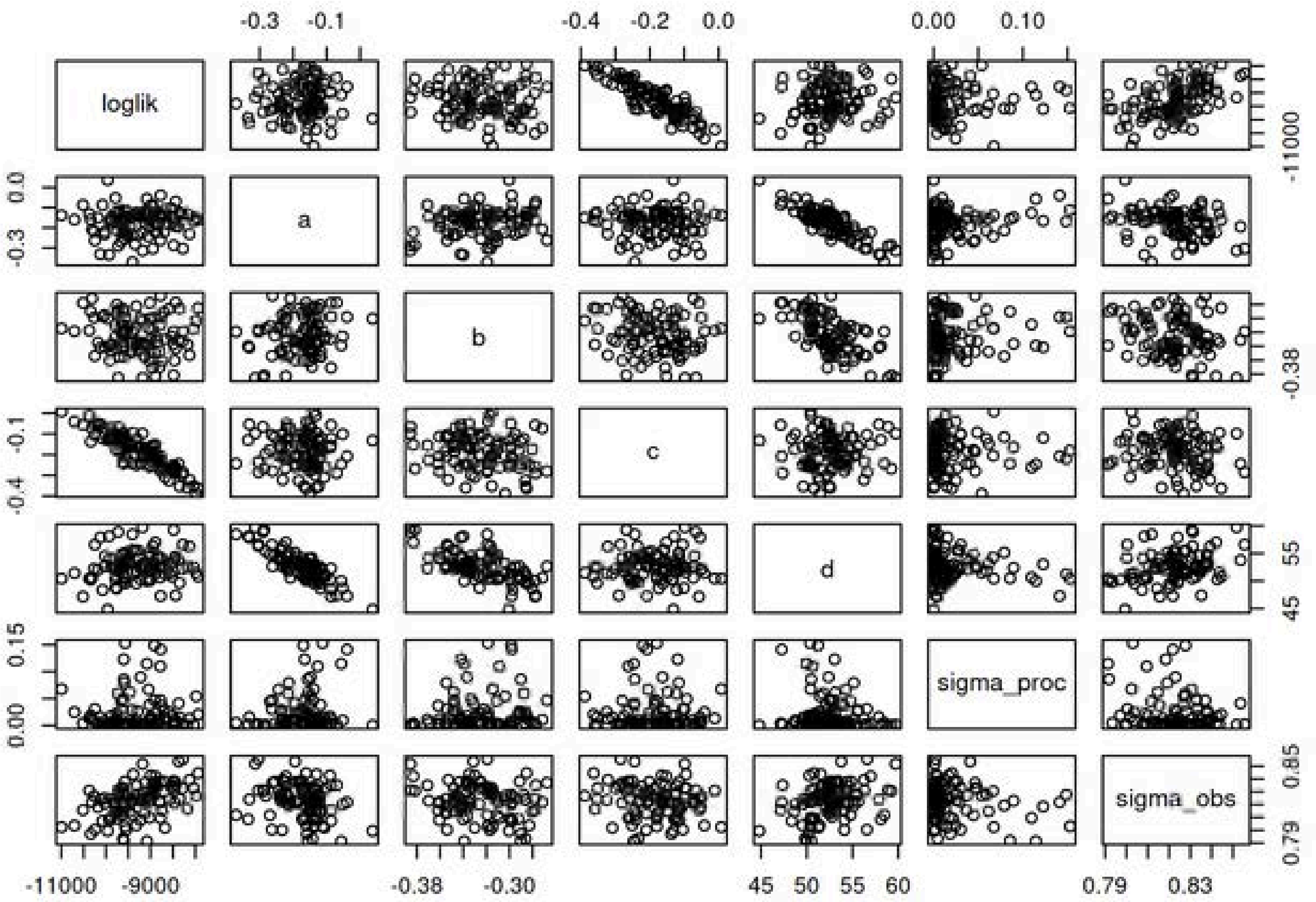


Figure 5. Paired Scatter Plot of Parameters

To probe whether the local search fit had located merely a optimum, I initiated a broad exploration in which $96$ MIF chains were launched from uniformly drawn starting values. Each chain ran for $150$ iterations with $N_p = 2\,000$ particles per filtering step. After the final iteration I re-evaluated the terminal parameter vector of every chain with $48$ replicate particle filters and summarised the ensemble likelihood by its Monte–Carlo mean. The largest of these values, $widehat\ell_{\max}^{(\text{global})} = -5\,244$ (it displays -7936 in the html file, not sure what is wrong), is markedly lower than the best log-likelihood obtained during the local search, $\hat{\ell}_{\max}^{(\text{local})} \approx -3\,235$. The discrepancy does not suggest that the local solution was spurious. Rather, it arises from the practical settings of the global run. With only $150$ MIF iterations and a relatively small particle might set the algorithm seldom had time to climb the likelihood ridge identified by the local search. Moreover, the log-likelihood was scored at the iterate of every chain; because random-walk perturbations are still active at the final step, that iterate is often inferior to the best one visited earlier in the same chain. A pairs plot of the terminal parameter vectors confirms this reading: the high-likelihood points cluster in the same $(a, b, c, d, \sigma_{\text{obs}})$ neighbourhood found by the local search, whereas the majority of global guesses occupy regions that attribute excessive variance simultaneously to the latent process and the measurement error, thereby driving the likelihood far below the optimum.

Taken together, the evidence indicates that the local search had already located the dominant mode of the likelihood surface and that the global exploration, while useful as a diagnostic, uncovered no competing solution with superior support. Given the time constraints of the present report, I retain $\hat{\ell}_{\max}^{(\text{local})} \approx -3\,235$ as the maximised log-likelihood.

# Conclusion

The linear–Gaussian POMP we fitted provides a coherent framework for linking daily noise exposure to heart-rate variability, yet the empirical performance of this first implementation is unsatisfactory. After 48 independent local–search chains the best particle–filter estimate of the log-likelihood stabilised near $\hat{\ell}_{\max}^{\mathrm{POMP}} \approx -3\,235$. By contrast, a purely empirical $\mathrm{ARIMA}(5,1,6)$ model fitted to the same differenced series achieved $\ell_{\max}^{\mathrm{ARIMA}} \approx -2\,591$, and an ordinary least-squares regression of SDNN on noise and activity returned $\ell_{\max}^{\mathrm{lm}} \approx -3\,025$. Because AIC penalises models only by the number of parameters, the POMP's much lower log-likelihood translates into a markedly worse AIC than either benchmark, implying that the present model specification does not yet capture the dominant structure in the data.

Several factors likely contributed to this gap.
First, the parameter search never reached a clear global optimum: the log-likelihood continued to drift late in each $\mathrm{MIF}_2$ run, and a broader random-initialisation experiment produced an even wider range of sub-optimal values. Second, the model assigns virtually all residual variance to measurement error; with negligible process noise the latent state has little flexibility to accommodate day-to-day shocks, so any model mis-specification is pushed into the likelihood penalty. Finally, collapsing individual records into a single pooled time series may have blurred subject-specific dynamics that a one-component latent process cannot represent.

Given these limitations the parameter estimates reported here— including the tentative noise coefficient $\hat{b}$—should be regarded as exploratory. In future work I plan to (1) revisit the state equation, allowing a genuinely slow baseline random walk alongside a fast noise-driven component, (2) profile the likelihood on a much finer grid with a larger particle set, and (3) compare alternative observation models (e.g. log-normal vs. Gaussian) to determine whether a different error structure improves fit. Only after a stable maximum likelihood is secured will it be possible to assess, with confidence, the day-level impact of environmental noise on HRV.

# References


Brockmann et.al. 2023. "Heart Rate Variability Changes with Respect to Time and Exercise Intensity During Heart-Rate-Controlled Steady-State Treadmill Running." https://doi.org/10.1038/s41598-023-35717-0.

Buchheit et.al. 2006. "Relationship Between Very High Physical Activity Energy Expenditure, Heart Rate Variability and Self-Estimate of Health Status in Middle-Aged Individuals." https://www.thieme-connect.com/products/ejournals/html/10.1055/s-2005-872929.

Ionides, Edward. 2025. "Notes of Lecture 5 for STATS/DATASCI 531, Modeling and Analysis of Time Series Data." https://ionides.github.io/531w25/5/.

Kraus et.al. 2013. "Individual Daytime Noise Exposure During Routine Activities and Heart Rate Variability in Adults: A Repeated Measures Study." https://doi.org/10.1289/ehp.1205606.

Task Force of the European Society of Cardiology the North American Society of Pacing Electrophysiology. 1996. "Heart Rate Variability : Standards of Measurement, Physiological Interpretation, and Clinical Use." https://doi.org/10.1161/01.CIR.93.5.1043.

Walker et.al. 2016. "Cardiovascular and Stress Responses to Short-Term Noise Exposures—a Panel Study in Healthy Males." https://doi.org/10.1016/j.envres.2016.06.016.