Register No.: 2548118

Course/Section:2MSTAT

MST415-2 [Stochastic Processes]

Name: Joyeeta Goswami

- **Derive the distribution of the compound process: S(t) when the interarrival times are exponential and X_i's are further independent exponential**

1. Problem setup

We have a compound Poisson process $S(t)$ defined as:

$$S(t) = \sum_{i=1}^{N(t)} X_i$$

where:

- $N(t)$ is a Poisson process with rate λ, so the interarrival times T_i are i.i.d. Exp(λ).
- X_i are i.i.d. Exp(μ), independent of $N(t)$.

We want the distribution of $S(t)$.

---

2. Known theory

For a compound Poisson process:

$$\mathbb{E}[e^{iuS(t)}] = \exp\left[\lambda t(\mathbb{E}[e^{iuX_1}] - 1)\right]$$

because $N(t) \sim$ Poisson(λt).

---

3. Moment generating function approach

Let's use the MGF $M_{S(t)}(s) = \mathbb{E}[e^{sS(t)}]$ for $s < \mu$ (since X_i are exponential, MGF exists for $s < \mu$).

For a compound Poisson process:

$$M_{S(t)}(s) = \exp\left[\lambda t(M_X(s) - 1)\right]$$

where $M_X(s) = \mathbb{E}[e^{sX_1}] = \frac{\mu}{\mu-s}$ for $s < \mu$.

Thus:

$$M_{S(t)}(s) = \exp\left[\lambda t\left(\frac{\mu}{\mu - s} - 1\right)\right]$$

$$= \exp\left[\lambda t \cdot \frac{\mu - (\mu - s)}{\mu - s}\right]$$
$$= \exp\left[\lambda t \cdot \frac{s}{\mu - s}\right]$$

So:

$$M_{S(t)}(s) = \exp\left[\frac{\lambda t s}{\mu - s}\right], s < \mu.$$

---

4. Recognizing the distribution

The MGF

$$M(s) = \exp\left[a \cdot \frac{s}{b - s}\right]$$

with $a = \lambda t, b = \mu$, is a known MGF: it's the MGF of a compound Poisson–Exponential sum, which turns out to be a Gamma distribution only in special cases? Let's check.

Actually, let's invert it.

---

5. Inverting the MGF / Using Laplace transform

Let $f_{S(t)}(x)$ be the pdf of $S(t)$. Its Laplace transform is:

$$\mathcal{L}\{f_{S(t)}\}(s) = \mathbb{E}[e^{-sS(t)}] = M_{S(t)}(-s) = \exp\left[\frac{-\lambda ts}{\mu + s}\right].$$

So:

$$\mathcal{L}\{f_{S(t)}\}(s) = \exp\left[-\lambda t\left(1 - \frac{\mu}{\mu + s}\right)\right].$$

---

6. Known result: special case

If $X_i \sim \text{Exp}(\mu)$, then $S(t)$ has a distribution known as compound Poisson with exponential jumps, which is a compound Poisson whose Laplace exponent is:

$$\Psi(s) = \lambda\left(1 - \frac{\mu}{\mu + s}\right) = \frac{\lambda s}{\mu + s}.$$

So the Laplace transform of $S(t)$ is:

$$\mathcal{L}\{f_{S(t)}\}(s) = e^{-t\Psi(s)} = \exp\left[-\frac{\lambda ts}{\mu + s}\right].$$

---

7. Identifying as a special compound distribution

We can think of $S(t)$ as having an atom at 0 (since $N(t) = 0$ with probability $e^{-\lambda t}$ gives $S(t) = 0$).

For $x > 0$, the density can be found by series expansion:

$$\mathcal{L}\{f_{S(t)}\}(s) = \exp\left[-\lambda t + \frac{\lambda t\mu}{\mu + s}\right] = e^{-\lambda t}\exp\left[\frac{\lambda t\mu}{\mu + s}\right].$$

Let $\alpha = \lambda t\mu, \beta = \mu$.

Then:

$$\mathcal{L}\{f_{S(t)}\}(s) = e^{-\lambda t}\exp\left[\frac{\alpha}{\beta + s}\right].$$

---

8. Series expansion

$$\exp\left[\frac{\alpha}{\beta + s}\right] = \sum_{n=0}^{\infty} \frac{\alpha^n}{n!}(\beta + s)^{-n}.$$

But $(\beta + s)^{-n}$ is the Laplace transform of $\frac{x^{n-1}e^{-\beta x}}{\Gamma(n)}$.

Thus, for $x > 0$:

$$f_{S(t)}(x) = e^{-\lambda t}\sum_{n=1}^{\infty} \frac{\alpha^n}{n!} \cdot \frac{x^{n-1}e^{-\mu x}}{\Gamma(n)}.$$

Since $\Gamma(n) = (n - 1)!$:

$$f_{S(t)}(x) = e^{-\lambda t} e^{-\mu x} \sum_{n=1}^{\infty} \frac{(\lambda t \mu)^n}{n!\,(n-1)!} x^{n-1}.$$

Let $k = n - 1$, then $n = k + 1$:

$$f_{S(t)}(x) = e^{-\lambda t} e^{-\mu x} \sum_{k=0}^{\infty} \frac{(\lambda t \mu)^{k+1}}{(k+1)!\,k!} x^{k}.$$

$$= \lambda t \mu\, e^{-\lambda t} e^{-\mu x} \sum_{k=0}^{\infty} \frac{(\lambda t \mu)^{k}}{(k+1)!\,k!} x^{k}.$$

---

9. Bessel function form

Recall the modified Bessel function $I_\nu(z)$:

$$I_\nu(z) = \sum_{k=0}^{\infty} \frac{(z/2)^{\nu+2k}}{k!\,\Gamma(\nu + k + 1)}.$$

Here, $\nu = 1$:

$$I_1(z) = \sum_{k=0}^{\infty} \frac{(z/2)^{1+2k}}{k!\,(k+1)!}.$$

Let $z = 2\sqrt{\lambda t \mu x}$, then $(z/2)^{1+2k} = (\sqrt{\lambda t \mu x})^{1+2k} = (\lambda t \mu x)^k \sqrt{\lambda t \mu x}$.

Check carefully:

We want $\frac{(\lambda t\mu)^k x^k}{(k+1)!k!}$ in our sum.

Compare with $I_1(z)$ term: $\frac{(z/2)^{1+2k}}{k!(k+1)!}$.

Let $z = 2\sqrt{\lambda t\mu x}$, then $(z/2)^{1+2k} = (\sqrt{\lambda t\mu x})^{1+2k} = (\lambda t\mu x)^k\sqrt{\lambda t\mu x}$.

So:

$$I_1(2\sqrt{\lambda t\mu x}) = \sqrt{\lambda t\mu x}\sum_{k=0}^{\infty} \frac{(\lambda t\mu x)^k}{k!\,(k+1)!}.$$

Thus:

$$\sum_{k=0}^{\infty} \frac{(\lambda t\mu)^k x^k}{k!\,(k+1)!} = \frac{I_1(2\sqrt{\lambda t\mu x})}{\sqrt{\lambda t\mu x}}.$$

---

10. Final density

Substitute into earlier expression:

$$f_{S(t)}(x) = \lambda t\mu\, e^{-\lambda t}e^{-\mu x} \cdot \frac{I_1(2\sqrt{\lambda t\mu x})}{\sqrt{\lambda t\mu x}}, x > 0.$$

And $P(S(t) = 0) = e^{-\lambda t}$.

So the distribution is:

$$\boldsymbol{S(t) \sim \textbf{Compound Poisson–Exponential}}$$

with pdf for $x > 0$:

$$f_{S(t)}(x) = \lambda t \sqrt{\frac{\mu}{\lambda t x}}\, e^{-\lambda t - \mu x} I_1(2\sqrt{\lambda t \mu x}).$$

Wait, simplify constant: $\lambda t \mu / \sqrt{\lambda t \mu x} = \sqrt{\frac{\lambda t \mu}{x}}$.

So:

$$f_{S(t)}(x) = \sqrt{\frac{\lambda t \mu}{x}}\, e^{-\lambda t - \mu x} I_1(2\sqrt{\lambda t \mu x}),\, x > 0.$$

---

11. Conclusion

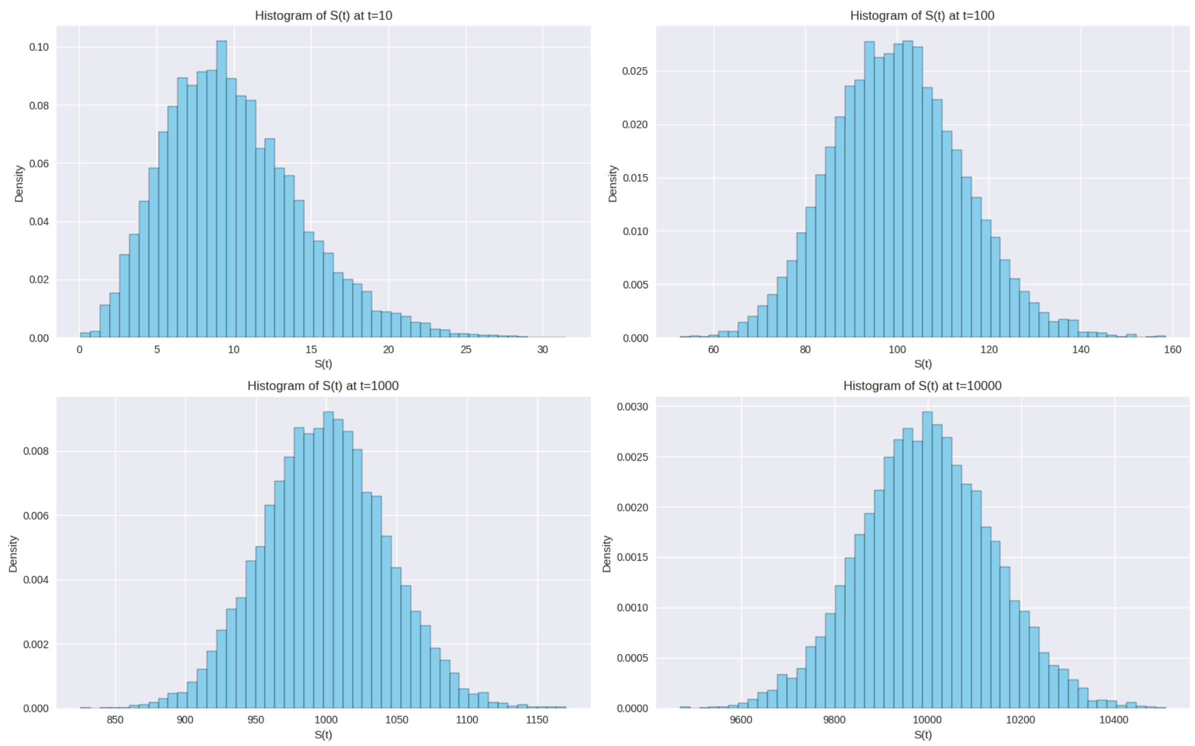
The distribution of $S(t)$ is mixed:

- **Point mass at 0: $e^{-\lambda t}$**
- **Continuous density for $x > 0$:**

$$\boxed{f_{S(t)}(x) = \sqrt{\frac{\lambda t \mu}{x}}\, e^{-\lambda t - \mu x}\, I_1(2\sqrt{\lambda t \mu x})}$$

where I_1 is the modified Bessel function of the first kind of order 1.

- Plot the histogram at t=10, 100, 1000, 10000



What the plots show

- **At $t = 10$:** The distribution is still quite spread out, with noticeable skewness. Many samples are close to zero, but the tail extends far to the right.
- **At $t = 100$:** The histogram begins to resemble a bell-shaped curve. By the Central Limit Theorem, the sum of many exponential jumps starts approximating normality.
- **At $t = 1000$:** The distribution is tightly concentrated around its mean ($\mathbb{E}[S(t)] = t$ since $\lambda = \beta = 1$). Variance grows linearly with t, but relative fluctuations shrink.
- **At $t = 10000$:** The histogram is very sharply peaked around 10,000, looking almost Gaussian. The law of large numbers dominates, so the distribution stabilizes near its mean.

Key insights

- **Mean growth:** $\mathbb{E}[S(t)] = t$. The center of the histogram shifts linearly with time.
- **Variance growth:** $\mathrm{Var}(S(t)) = 2t$. Spread increases, but relative variability decreases as $1/\sqrt{t}$.
- **Shape evolution:** Starts skewed (Gamma-like), then transitions toward normal as $t$grows.

📌 **Role of the Interarrival Parameter (λ)**

- **Definition:** Interarrival times are exponential with rate λ. This means the counting process $N(t)$is Poisson with mean λt.
- **Impact:**
  - **Frequency of jumps:** Larger λ→ more arrivals per unit time → more jumps in $S(t)$.
  - **Mean of $S(t)$:** $\mathbb{E}[S(t)] = \frac{\lambda t}{\beta}$. So increasing λlinearly increases the expected value of the process.
  - **Variance:** $\mathrm{Var}(S(t)) = \frac{2\lambda t}{\beta^2}$. Higher λincreases variability proportionally.
  - **Distribution shape:** With larger λ, the distribution of $S(t)$becomes more concentrated around its mean (relative fluctuations shrink as $1/\sqrt{\lambda t}$).

Role of the Jump Parameter (β)

- **Definition:** Each jump size $X_i \sim \mathrm{Exp}(\beta)$.
- **Impact:**
  - **Average jump size:** $\mathbb{E}[X_i] = 1/\beta$. Smaller β→ larger jumps on average.
  - **Mean of $S(t)$:** $\mathbb{E}[S(t)] = \frac{\lambda t}{\beta}$. So decreasing βincreases the mean.
  - **Variance contribution:** $\mathbb{E}[X_i^2] = 2/\beta^2$. Smaller β→ heavier variance per jump, so overall variance grows faster.
  - **Tail behavior:** Lower βproduces heavier tails in the distribution of $S(t)$, since jumps are larger and more spread out.

Combined Effect

- **Balance of frequency vs. size:**

- High λ, high β: many small jumps → smoother, more Gaussian-like distribution.
- Low λ, low β: few but large jumps → spiky, heavy-tailed distribution.
- High λ, low β: many large jumps → rapid growth of $S(t)$.
- Low λ, high β: few small jumps → process often near zero.

- **Scaling:** Both parameters appear in the mean and variance formulas:

$$\mathbb{E}[S(t)] = \frac{\lambda t}{\beta}, \mathrm{Var}(S(t)) = \frac{2\lambda t}{\beta^2}.$$

So the "speed" of growth is controlled by the ratio λ/β.

In short:

- λcontrols **how often jumps occur**.
- βcontrols **how big each jump is**. Together, they determine the **scale, variability, and shape** of the distribution of $S(t)$.

• Deploy everything in R Shiny to understand the sensitivity of the parameters on S(t) vs time

This app lets you experiment with the compound Poisson process S(t) with exponential interarrivals (rate λ) and exponential jumps (rate β). You can simulate trajectories, visualize histograms at a chosen time, and compare with the exact density and normal approximation.

What you can explore

- **Trajectories:** See how S(t) evolves under different λ (arrival rate) and β (jump size rate).
- **Histograms at time t⋆:** Compare simulation with the exact density and a large-t normal approximation.
- **Moments:** Track theoretical mean and variance curves over time.

R Shiny app code

```
# app.R
library(shiny)
library(ggplot2)

# ---------- Theory helpers ----------
# Exact density of S(t) for Exp(Î») arrivals and Exp(Î²) jumps (compound
Poisson with exponential jumps)
# f_S(s; t, Î», Î²) = e^{-Î» t - Î² s} * sqrt(Î» t Î² / s) * I1(2 sqrt(Î» t
Î² s)), s > 0
# Atom at 0: P(S(t)=0) = e^{-Î» t}
f_S_density <- function(s, t, lambda, beta) {
  dens <- numeric(length(s))
  idx <- which(s > 0)
  if (length(idx) > 0) {
    x <- s[idx]
    z <- 2 * sqrt(lambda * t * beta * x)
    term <- exp(-lambda * t - beta * x) * sqrt(lambda * t * beta / x) *
besselI(z, nu = 1)
    dens[idx] <- term
  }
  dens
}

p0_mass <- function(t, lambda) exp(-lambda * t)

# Theoretical mean and variance
S_mean <- function(t, lambda, beta) (lambda * t) / beta
S_var  <- function(t, lambda, beta) (2 * lambda * t) / (beta^2)

# Normal approximation at large t: N(mean, var)
f_normal_approx <- function(s, t, lambda, beta) {
  mu <- S_mean(t, lambda, beta); v <- S_var(t, lambda, beta)
  dnorm(s, mean = mu, sd = sqrt(v))
}

# ---------- Simulation helpers ----------
# Simulate one path of S(t) on [0, T] over a given grid of times
simulate_path <- function(T, lambda, beta, time_grid) {
  # Generate arrival times by exponential interarrivals
  nmax <- rpois(1, lambda * T) # number of arrivals by T (alternative:
sequential generation)
  if (nmax == 0) return(rep(0, length(time_grid)))
  # Arrival epochs: sort Uniform order statistic? Better: cum sum of
exponentials truncated
  interarrivals <- rexp(nmax, rate = lambda)
  arrivals <- cumsum(interarrivals)
  arrivals <- arrivals[arrivals <= T]
  k <- length(arrivals)
  if (k == 0) return(rep(0, length(time_grid)))
  jumps <- rexp(k, rate = beta)
  cumsums <- cumsum(jumps)
  # Piecewise constant right-continuous process
  S <- numeric(length(time_grid))
  ai <- 1
  for (i in seq_along(time_grid)) {
    t <- time_grid[i]
    while (ai <= k && arrivals[ai] <= t) ai <- ai + 1
    if (ai == 1) {
```

```r
      S[i] <- 0
    } else {
      S[i] <- cumsums[ai - 1]
    }
  }
  S
}

# Sample S(t*) by compounding: draw N ~ Pois(Î» t*), then sum N exponentials with rate Î²
sample_S_at_t <- function(tstar, lambda, beta, n_sims) {
  N <- rpois(n_sims, lambda * tstar)
  # Efficient: for each i, if N[i] > 0, sum that many Exp(Î²)
  Svals <- numeric(n_sims)
  positive_idx <- which(N > 0)
  if (length(positive_idx) > 0) {
    for (j in positive_idx) {
      Svals[j] <- sum(rexp(N[j], rate = beta))
    }
  }
  Svals
}

# ---------- UI ----------
ui <- fluidPage(
  titlePanel("Compound Poisson S(t) sensitivity: exponential interarrivals (Î») and jumps (Î²)"),
  sidebarLayout(
    sidebarPanel(
      sliderInput("lambda", "Interarrival rate Î»:", min = 0.1, max = 5, value = 1, step = 0.1),
      sliderInput("beta",   "Jump rate Î²:",        min = 0.1, max = 5, value = 1, step = 0.1),
      sliderInput("T",      "Time horizon T:",     min = 10,  max = 1000, value = 200, step = 10),
      sliderInput("nPaths", "Number of simulated paths:", min = 1, max = 200, value = 50, step = 1),
      sliderInput("nGrid",  "Points in time grid:", min = 100, max = 3000, value = 1000, step = 100),
      hr(),
      sliderInput("tstar",  "Histogram time tâ‹†:", min = 1, max = 1000, value = 100, step = 1),
      sliderInput("nSims",  "Histogram samples:", min = 1000, max = 50000, value = 10000, step = 1000),
      checkboxInput("showExact", "Overlay exact density", TRUE),
      checkboxInput("showNormal", "Overlay normal approximation", TRUE),
      checkboxInput("logY", "Log y-scale for histogram density", FALSE),
      hr(),
      checkboxInput("showMoments", "Show theoretical mean/variance curves", TRUE),
      helpText("Tip: Increase Î» for more frequent jumps; decrease Î² for larger jumps.")
    ),
    mainPanel(
      tabsetPanel(
        tabPanel("Trajectories",
                 plotOutput("trajPlot", height = "420px"),
                 verbatimTextOutput("trajStats")),
        tabPanel("Histogram at tâ‹†",
                 plotOutput("histPlot", height = "420px"),
                 verbatimTextOutput("histStats")),
```

```r
        tabPanel("Moments over time",
                 plotOutput("momentsPlot", height = "420px"))
      )
    )
  )
)

# ---------- Server ----------
server <- function(input, output, session) {

  time_grid <- reactive(seq(0, input$T, length.out = input$nGrid))

  # Simulate paths
  paths <- reactive({
    tg <- time_grid()
    lambda <- input$lambda; beta <- input$beta
    nP <- input$nPaths
    Smat <- matrix(0, nrow = length(tg), ncol = nP)
    for (j in 1:nP) {
      Smat[, j] <- simulate_path(T = input$T, lambda = lambda, beta = beta,
time_grid = tg)
    }
    list(t = tg, S = Smat)
  })

  output$trajPlot <- renderPlot({
    pr <- paths()
    df <- data.frame(t = rep(pr$t, times = ncol(pr$S)),
                     S = as.vector(pr$S),
                     path = factor(rep(seq_len(ncol(pr$S)), each =
length(pr$t))))
    ggplot(df, aes(x = t, y = S, group = path, color = path)) +
      geom_line(linewidth = 0.7, alpha = 0.7, show.legend = FALSE) +
      labs(title = "Simulated trajectories of S(t)",
           x = "Time t",
           y = "S(t)") +
      theme_minimal()
  })

  output$trajStats <- renderText({
    lambda <- input$lambda; beta <- input$beta; T <- input$T
    muT <- S_mean(T, lambda, beta); varT <- S_var(T, lambda, beta)
    paste0(
      "Theoretical at T = ", T, ":\n",
      "  Mean E[S(T)] = ", round(muT, 4), "\n",
      "  Var Var[S(T)] = ", round(varT, 4), "\n",
      "  Atom at zero P{S(T)=0} = ", format(p0_mass(T, lambda), digits = 4)
    )
  })

  # Histogram at tstar
  samples <- reactive({
    sample_S_at_t(input$tstar, input$lambda, input$beta, input$nSims)
  })

  output$histPlot <- renderPlot({
    svals <- samples()
    tstar <- input$tstar; lambda <- input$lambda; beta <- input$beta
    p0 <- p0_mass(tstar, lambda)

    # Build histogram (density-scaled)
```

```
    df <- data.frame(S = svals)
    p <- ggplot(df, aes(x = S)) +
      geom_histogram(aes(y = ..density..), bins = 50, fill = "#4C78A8",
color = "white", alpha = 0.9) +
      labs(title = paste0("Histogram of S(tâ‹†) at tâ‹† = ", tstar, "
(density scale)"),
           x = "S(tâ‹†)", y = ifelse(input$logY, "log-density", "density"))
+
      theme_minimal()

    # Overlays
    sgrid <- seq(1e-6, max(max(svals), S_mean(tstar, lambda, beta) + 6 *
sqrt(S_var(tstar, lambda, beta))), length.out = 1000)

    if (input$showExact) {
      dens_exact <- f_S_density(sgrid, tstar, lambda, beta)
      p <- p + geom_line(aes(x = sgrid, y = dens_exact), color = "#F58518",
linewidth = 1.1)
    }
    if (input$showNormal) {
      dens_norm <- f_normal_approx(sgrid, tstar, lambda, beta)
      p <- p + geom_line(aes(x = sgrid, y = dens_norm), color = "#54A24B",
linewidth = 1.1, linetype = "dashed")
    }
    if (input$logY) p <- p + scale_y_log10()

    p
  })

  output$histStats <- renderText({
    tstar <- input$tstar; lambda <- input$lambda; beta <- input$beta
    mu <- S_mean(tstar, lambda, beta); v <- S_var(tstar, lambda, beta)
    p0 <- p0_mass(tstar, lambda)
    paste0(
      "Theoretical at tâ‹† = ", tstar, ":\n",
      "  Mean E[S(tâ‹†)] = ", round(mu, 4), "\n",
      "  Var Var[S(tâ‹†)] = ", round(v, 4), "\n",
      "  Atom at zero P{S(tâ‹†)=0} = ", format(p0, digits = 4)
    )
  })

  # Moments over time
  output$momentsPlot <- renderPlot({
    lambda <- input$lambda; beta <- input$beta
    tgrid <- seq(0, input$T, length.out = 300)
    df <- data.frame(
      t = rep(tgrid, 2),
      value = c(S_mean(tgrid, lambda, beta), S_var(tgrid, lambda, beta)),
      kind = factor(rep(c("Mean", "Variance"), each = length(tgrid)))
    )
    ggplot(df, aes(x = t, y = value, color = kind)) +
      geom_line(linewidth = 1.1) +
      labs(title = "Theoretical mean and variance of S(t)",
           x = "Time t", y = "Value") +
      scale_color_manual(values = c("Mean" = "#E45756", "Variance" =
"#72B7B2")) +
      theme_minimal() +
      if (input$showMoments) NULL
  })
}
```

```
shinyApp(ui, server)
```

Notes on the theory used

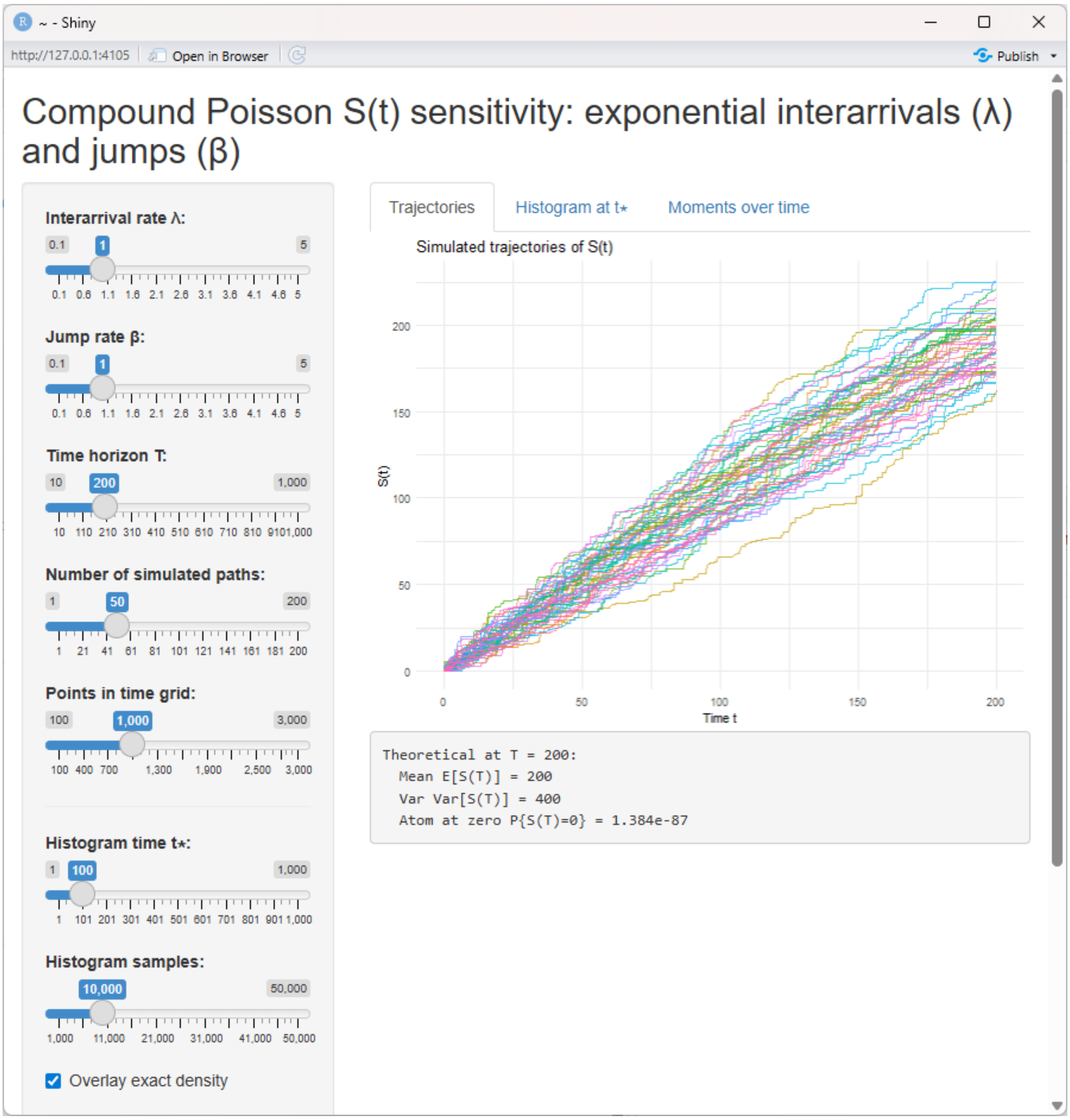
- Exact density: For $s > 0$, the density uses the modified Bessel function I_1: $f_S(s) = \exp(-\lambda t - \beta s) \times \sqrt{\lambda t \beta / s} \times I_1(2\sqrt{\lambda t \beta s})$.
- Atom at zero: $P\{S(t)=0\} = \exp(-\lambda t)$ from $N(t) \sim \text{Poisson}(\lambda t)$.
- Moments: $E[S(t)] = (\lambda t)/\beta$ and $Var[S(t)] = (2 \lambda t)/\beta^2$.
- Normal approximation: For large t, $S(t) \approx \text{Normal}(\text{mean}, \text{variance})$ by the Central Limit Theorem for compound Poisson sums.

Tips to explore sensitivity

- High λ, high β: Many small jumps; trajectories look smooth; histogram resembles Gaussian earlier.
- Low λ, low β: Few large jumps; heavy right tail; higher atom at zero for small t.
- Increase t⋆: The normal overlay should track the histogram more closely.
- Toggle log scale: Useful to see tail behavior, especially when β is small.

How to run

- Save the code as app.R.
- Run in R: install.packages(c("shiny","ggplot2")); then run shiny::runApp("path/to/app.R").
- Interact: Adjust λ, β, T, and t⋆ to see how frequency vs. size of jumps shape S(t) over time.


~ - Shiny
http://127.0.0.1:4105
Open in Browser
Publish
Compound Poisson S(t) sensitivity: exponential interarrivals (λ) and jumps (β)
Interarrival rate λ:
Jump rate β:
Time horizon T:
Number of simulated paths:
Points in time grid:
Histogram time t⋆:
Histogram samples:
Overlay exact density
Trajectories
Histogram at t⋆
Moments over time
Simulated trajectories of S(t)
S(t)
Time t
Theoretical at T = 200:
Mean E[S(T)] = 200
Var Var[S(T)] = 400
Atom at zero P{S(T)=0} = 1.384e-87
