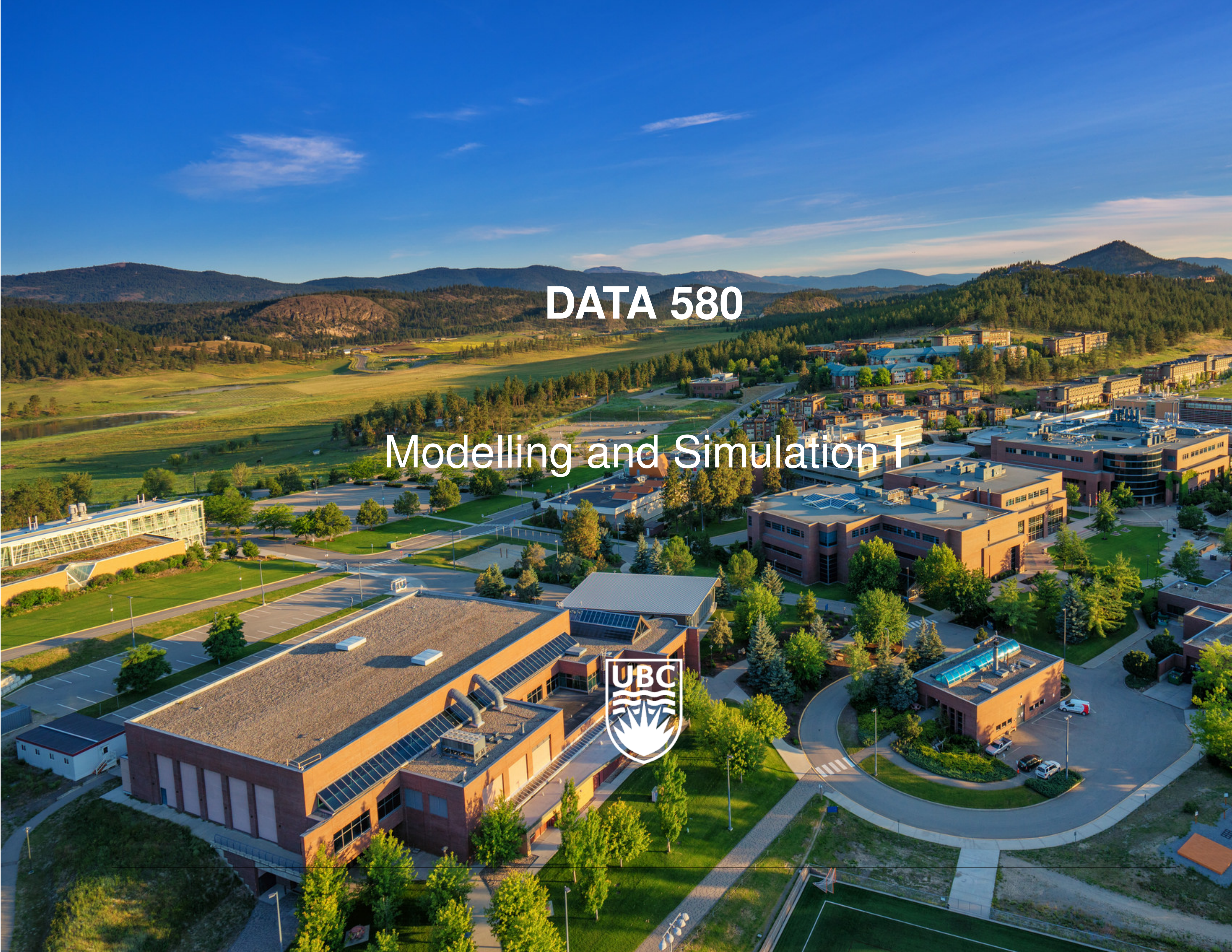


DATA 580

Modelling and Simulation I

Poisson Processes and Extensions

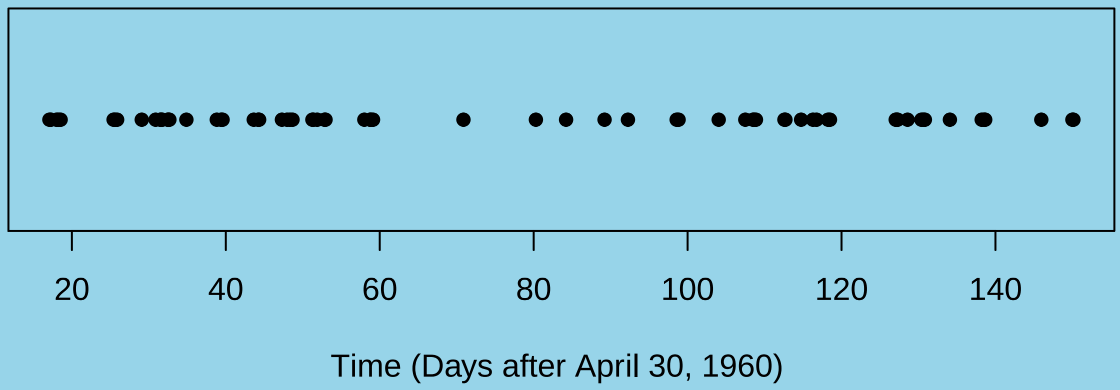
Point Events in Time

Examples:

- Times when a neuron fires.
- Times when a river floods its banks.
- Times when an earthquake occurs in a local area, such as San Francisco.
- Times when windspeed exceeds 160 km/h.
- Times when rain begins to fall.

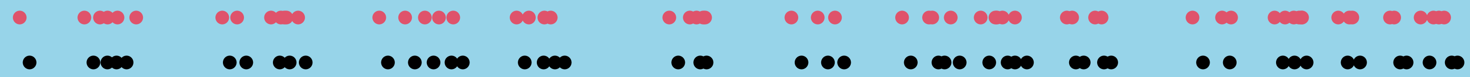
Point Events in Time

Example - Onset of rain in Winnipeg



Think – why might we care about the onset of rain in Winnipeg?

A Simple Reaction Time Experiment



The first 60 seconds of an experiment where a subject was shown flashes and had to respond by pressing a button. The red dots show the times of the flashes, and the black dots show the times of the responses.

The red dots were simulated from a Poisson process.

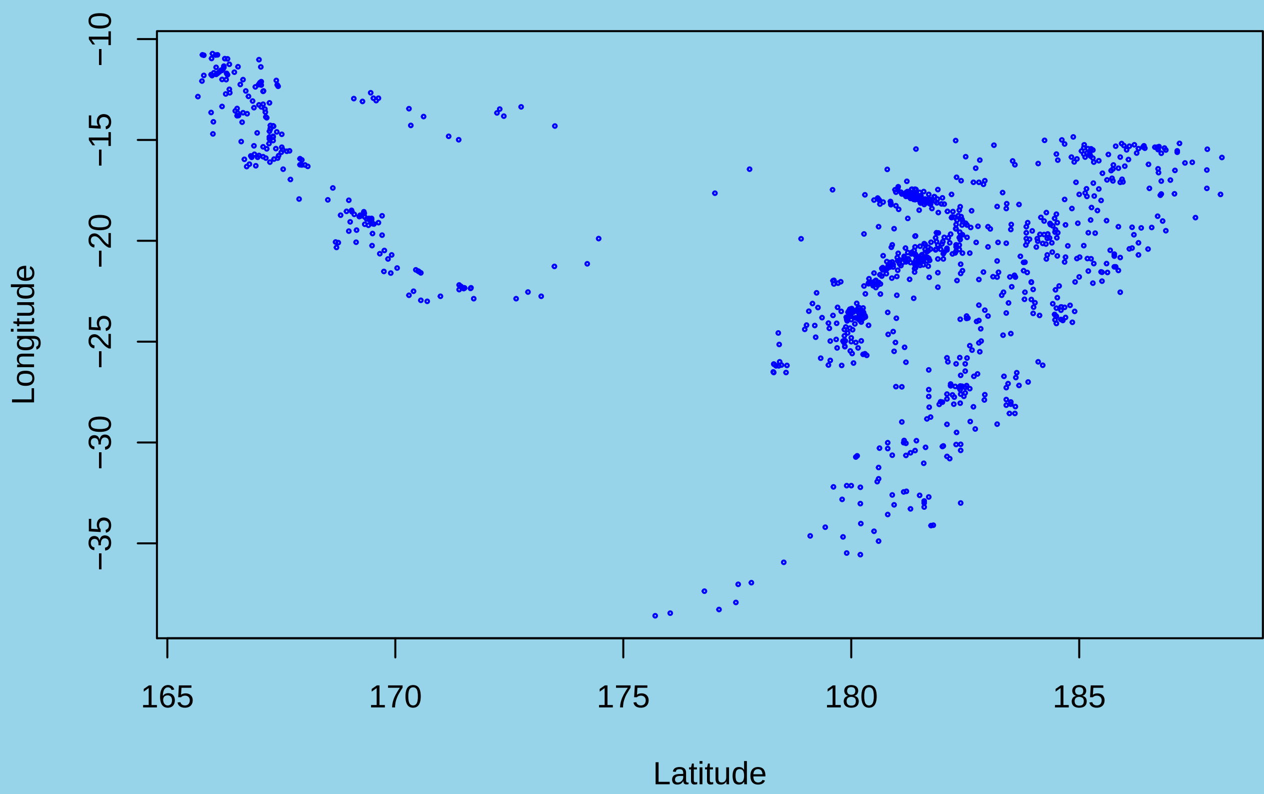
Point Events in Space

Examples:

- **Locations of wildland fire ignitions.**
- **Locations of diseased trees.**
- **Locations of space debris.**
- **Locations of galaxies.**
- **Locations of earthquake epicentres.**

Point Events in Space

Example - locations of earthquakes in the Fiji area:



Point Events in Space

Where in the World is Fiji?

Point Events in Space

And where were those earthquakes?

Modelling Point Events - Poisson Processes

A temporal Poisson process is a simple model of the collection of point events that occur during an interval of time.

A spatial Poisson process models collections of point events in a spatial region.

A way of thinking about a Poisson process is to think of a random collection of points on a line or in the plane (or in higher dimensions, if required).

Poisson Processes

The homogeneous Poisson process has the following properties:

1. The distribution of the number of points in a set is Poisson with rate proportional to the size of the set.
2. The numbers of points in non-overlapping sets are independent of each other.

In particular, for a Poisson process with rate λ the number of points on an interval $[0, T]$ is Poisson distributed with mean λT.

Poisson Processes

One way to simulate a Poisson process is as follows.

1. Generate N as a Poisson pseudorandom number with parameter λT.
2. Generate N independent uniform pseudorandom numbers on the interval $[0, T]$.
3. Sort the points to see their time ordering.

Poisson Process Simulation Algorithm

Algorithm to generate Poisson events on an interval $[0, t]$ at rate λ:

```
N <- rpois(1, lambda =  rate*time)  # The number of events
     has a Poisson lambda*time distribution
U <- runif(N, max = time)
sort(U)
```

Poisson Process Simulation Example

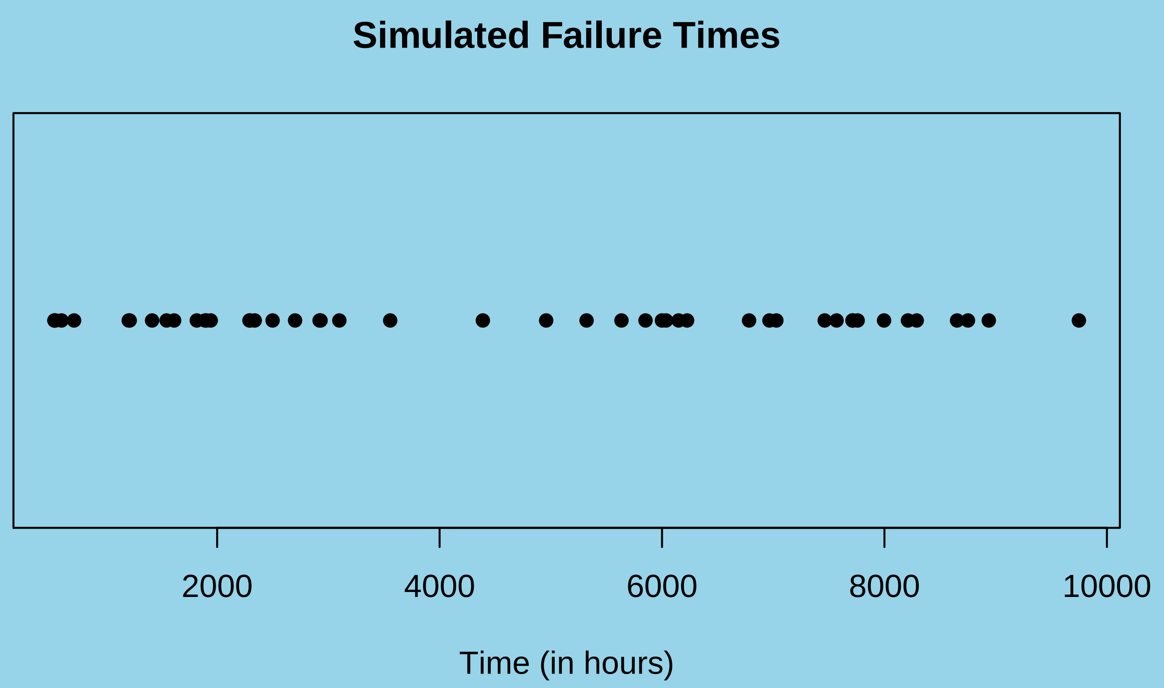
Simulate points of a homogeneous Poisson process having rate 1.5 on the interval $[0, 10]$.

```
rate <- 1.5; time <- 10
lambda <-
N <- rpois(1, lambda =  rate*time)
U <- runif(N, max = time)
sort(U)

##  [1] 0.6592 0.7170 1.1708 1.7424 2.3767 3.1335
##  [7] 3.4566 3.7615 4.1955 5.5967 7.2766 7.2778
## [13] 7.4157 9.5172
```

Another Poisson Process Simulation Example

Generate failure times on the interval $[0, 10000]$ for a machine having a failure rate of 1 every 200 hours. Display with a strip chart.

```
time <- 10000
rate <- 1/200
N <- rpois(1, lambda = time*rate)  # The number of failures
U <- runif(N, max = time)
stripchart(U, pch=16, main="Simulated Failure Times",
   xlab="Time (in hours)")
```



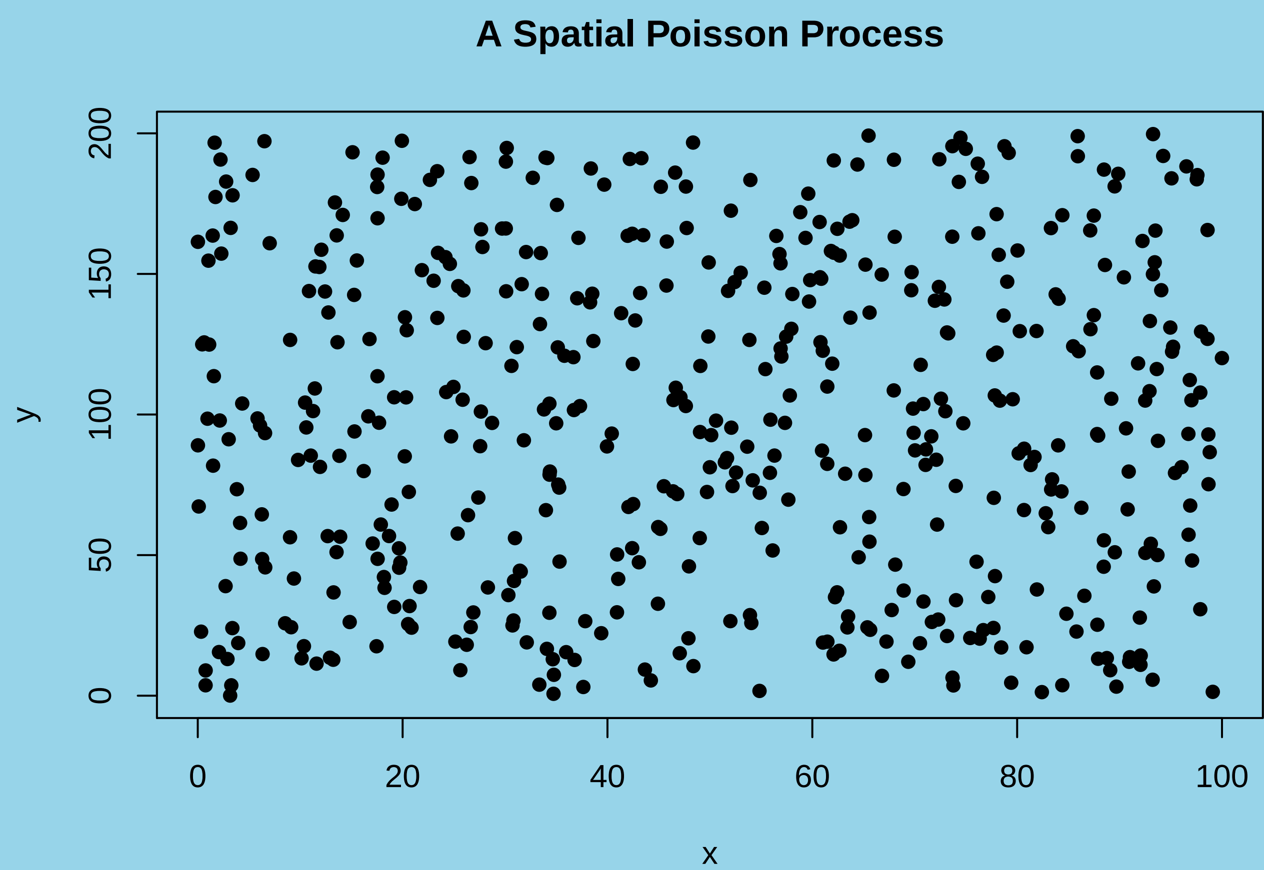
Simulating Poisson Processes in Higher Dimensions

To simulate a Poisson process in a rectangle, just simulate another set of N uniform numbers on the appropriate interval.

For example, to simulate a Poisson process in a 200 $\times$ 100 rectangle, use:

```
xlimit <- 100
ylimit <- 200
intensity <- 1/40  # "rate" no longer makes sense
N <- rpois(1, lambda = xlimit*ylimit*intensity)
U1 <- runif(N, max = xlimit) # x-coordinates of
     # Poisson points
U2 <- runif(N, max = ylimit) # y-coordinates
plot(U2 ~ U1, pch = 16, main = "A Spatial Poisson Process",
     xlab = "x", ylab = "y")
```

Simulating Poisson Processes in Higher Dimensions



Properties of 2-Dimensional Poisson Processes

1. The probability distribution of the number of points in any region of area A is Poisson with parameter $\lambda = A\mu$ where μ is the intensity (i.e. the mean number of events per unit area).

2. The numbers of points in non-overlapping regions are independent of each other.

Properties of 2-Dimensional Poisson Processes

Exercise:

Consider a Poisson process on a 3×5 rectangle with intensity $\lambda = 7$.

What is the expected number of points in the rectangle?

What is the variance of the number of points in the rectangle?

Find the probability that there are more than 120 points in the rectangle.

Consider the top half of the rectangle. Find the probability that there are more than 60 points in that region.

Properties of 2-Dimensional Poisson Processes

Answers:

What is the expected number of points in the rectangle?*

What is the variance of the number of points in the rectangle?†

Find the probability that there are more than 120 points in the rectangle.‡

Consider the top half of the rectangle. Find the probability that there are more than 60 points in that region.§

---

*The expected number of points is the intensity times the area: 7(15) = 105.

†The number of points is Poisson distributed and the mean and variance are equal, so the variance is 105.

‡Use `1 - ppois(120, lambda = 105)` to get the result 0.0677.

§Use `1 - ppois(60, lambda = 52.5)` to get the result 0.136 .

Properties of 2-Dimensional Poisson Processes

Exercise:

Consider a Poisson process inside a circle of radius 2 having intensity 3.

What is the expected number of points in the circle?

What is the variance of the number of points in the circle?

Find the probability that there are more than 50 points in the circle.

Consider the top half of the circle. Find the probability that there are more than 25 points in that region.

Properties of 2-Dimensional Poisson Processes

Answers:

What is the expected number of points in the circle?

```
lambda <- 3*pi*2^2  # area of circle times intensity
lambda

## [1] 37.7
```

This is the variance as well.

Properties of 2-Dimensional Poisson Processes

Find the probability that there are more than 50 points in the circle.

```
1 - ppois(50, lambda = lambda)

## [1] 0.02238
```

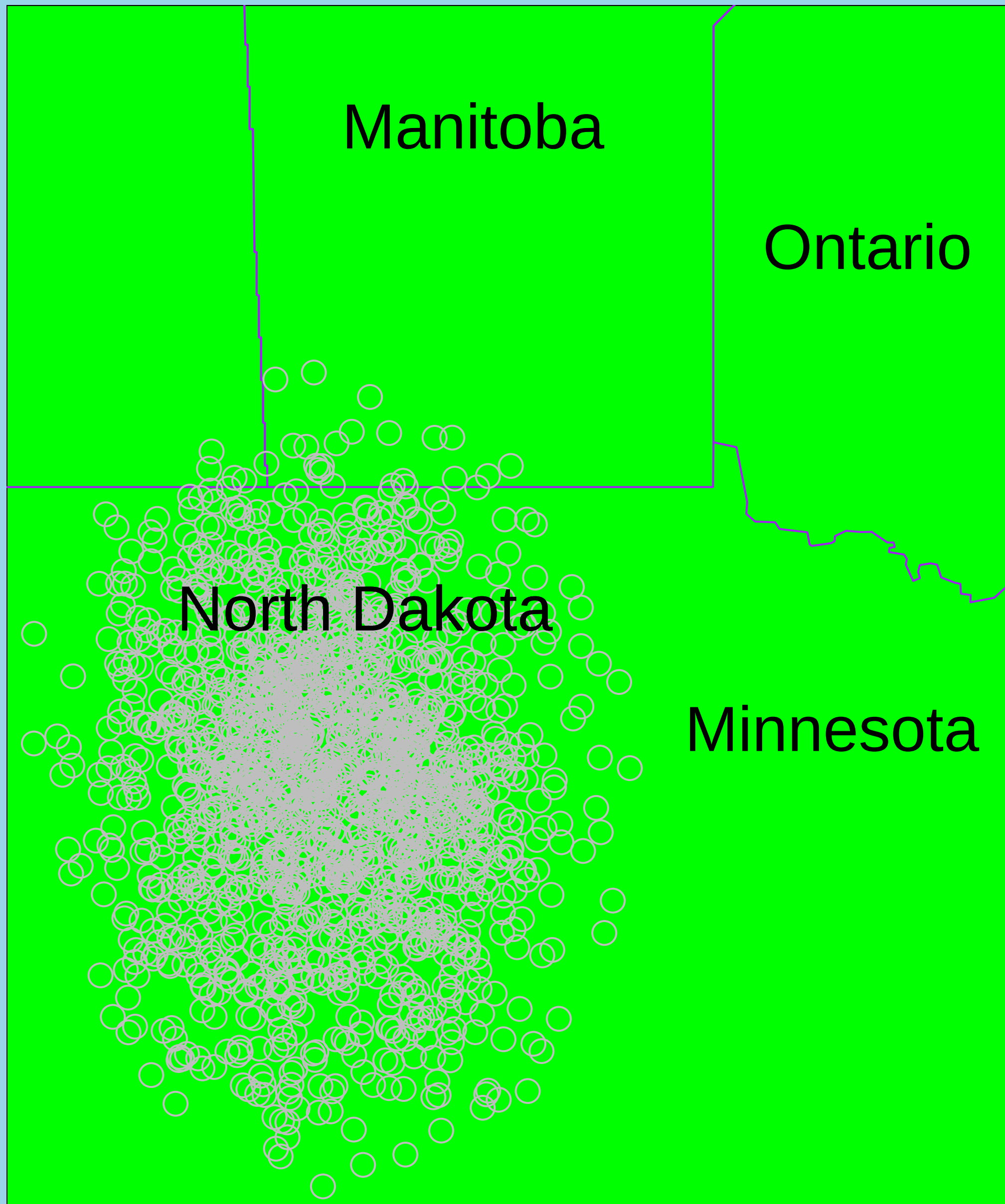
Consider the top half of the circle. Find the probability that there are more than 25 points in that region.

```
1 - ppois(25, lambda = lambda/2)

## [1] 0.06819
```

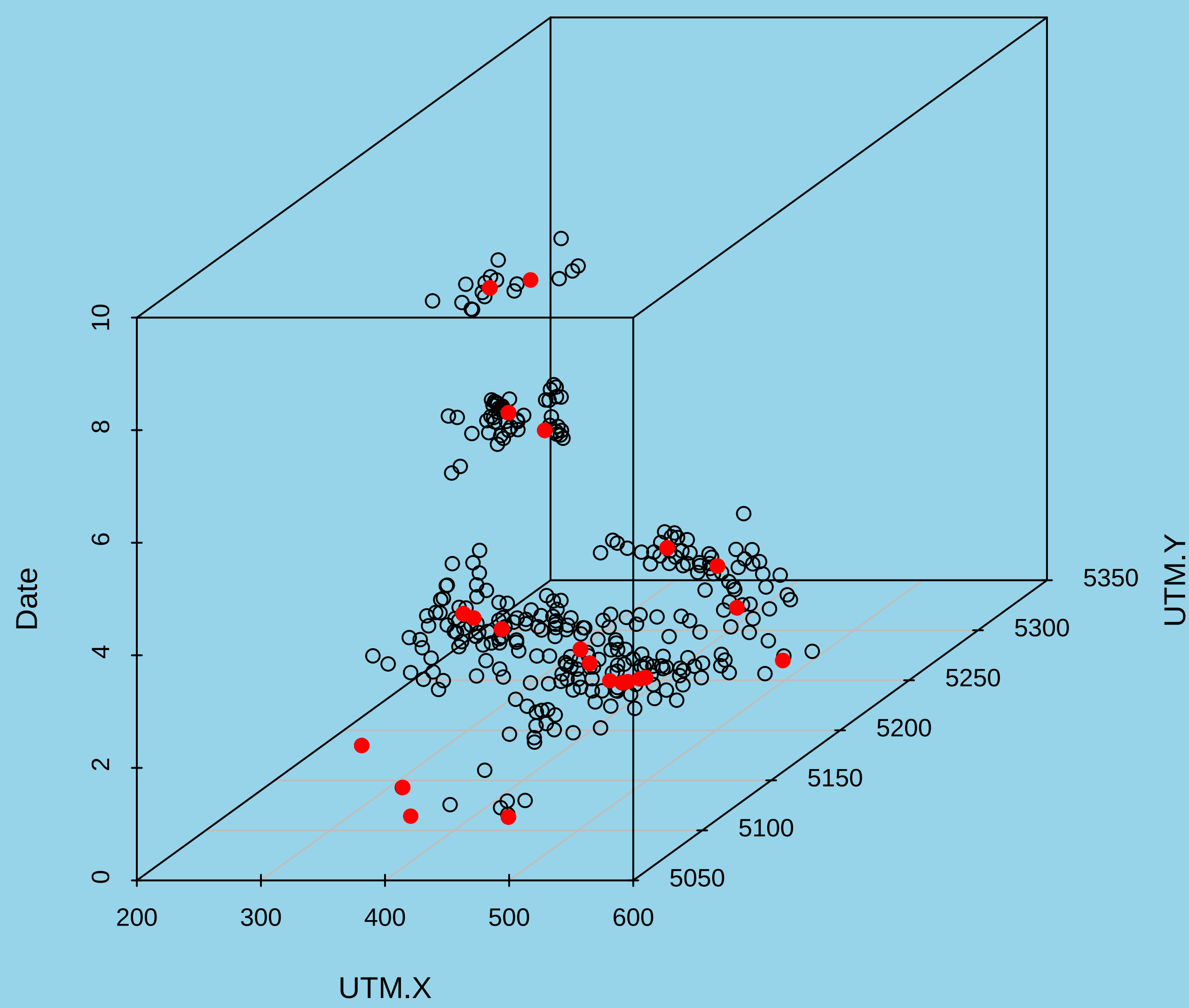
Simulating Poisson Processes in Higher Dimensions

To simulate in a 3rd dimension, such as time, just simulate another set of N uniform numbers on the appropriate interval. This time, we are thinking of a 120 hour period. And we suddenly have a simple-minded simulator of storm cells on a 100 by 200 km region in North Dakota.

```
xlimit <- 100
ylimit <- 200
tlimit <- 120
intensity <- 1/4800
N <- rpois(1, lambda = tlimit*xlimit*ylimit*intensity)
    # N is the number of events
U1 <- runif(N, max = xlimit) # x-coordinates
    # of Poisson points
U2 <- runif(N, max = ylimit) # y-coordinates
time <- runif(N, max = tlimit)  # time coordinates
```

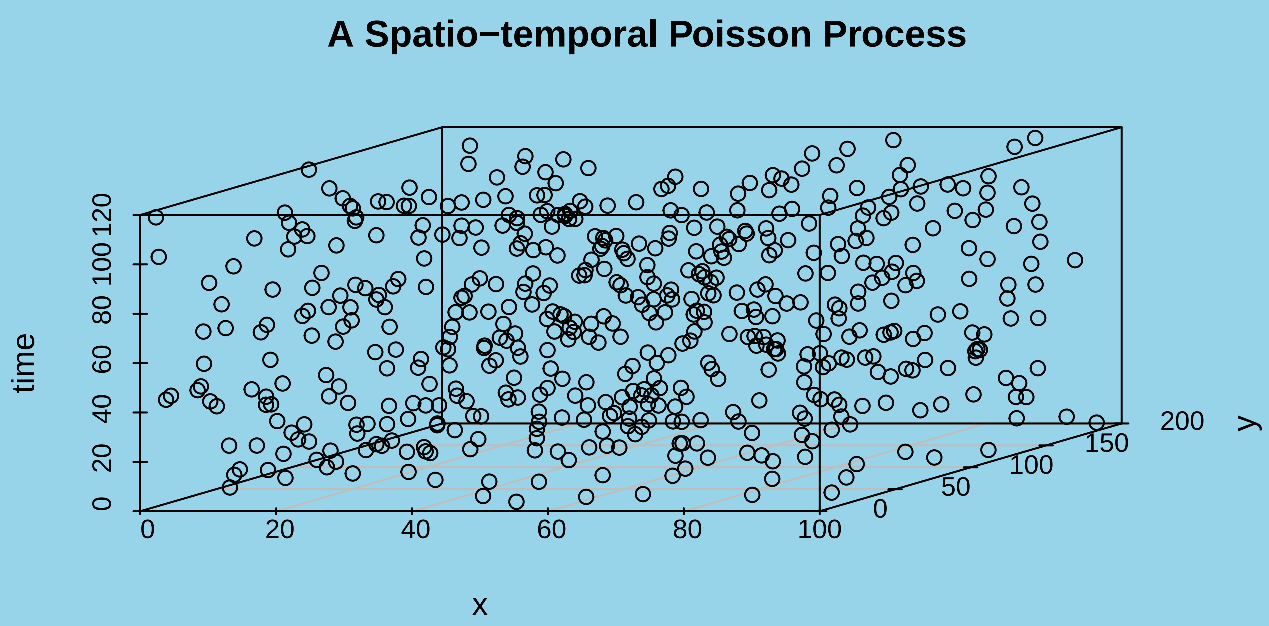

Manitoba
Ontario
North Dakota
Minnesota

Sharpened Storm Cells –– A Subset of 1998



Simulating Poisson Processes in Higher Dimensions

```r
library(scatterplot3d)
scatterplot3d(U1, U2, time, main =
      "A Spatio-temporal Poisson Process ",
     xlab = "x", ylab = "y")
```



but not realistic looking

Exponential Random Variables

Exponential random variables are used as basic models for such things as failure times of mechanical or electronic components, or for the time it takes a server to complete service to a customer.

The exponential distribution is characterized by a constant *failure rate*, denoted by λ.

T has an exponential distribution with rate $\lambda > 0$ if

$$\mathbf{P}(T \leq t) = 1 - e^{-\lambda t}$$

for any non-negative t.

Exponential Probabilities

The pexp() function can be used to evaluate the distribution function.

```
pexp(q, rate)
```

The output from this is the value of $P(T \leq q)$, where T is an exponential random variable with parameter `rate`.

Example

Suppose the service time at a bank teller can be modeled as an exponential random variable with rate 3 per minute.

Then the probability of a customer being served in less than 1 minute is

```
pexp(1, rate = 3)

## [1] 0.9502
```

Thus, $P(X \leq 1) = 0.95$, when X is an exponential random variable with rate 3.

Exponential Density Function and Moments

Differentiating the right hand side of the distribution function with respect to t gives the exponential probability density function:

$$f(t) = \lambda e^{-\lambda t}.$$

The dexp() function can be used to evaluate this.

It takes the same arguments as the pexp() function.

The expected value of an exponential random variable is $1/\lambda$, and the variance is $1/\lambda^2$.

Exponential Pseudorandom Numbers

A simple way to simulate exponential pseudorandom variates is based on the *inversion* method.

To see how the method works, note that if T is a random variable with CDF $F(x)$, for any $x \in (0, 1)$, we have

$$P(F(T) \le x) = P(T \le F^{-1}(x)) = F(F^{-1}(x)) = x.$$

Thus, $F(T)$ is a uniform random variable on the interval $(0, 1)$.

This means that if we generate a uniform variate U, there will be a corresponding value of T such that $U = F(T)$, and we can calculate that value using the relation

$$T = F^{-1}(U).$$

Exponential Pseudorandom Numbers

For an exponential random variable $F(x) = 1 - e^{-\lambda x}$, so

$$F^{-1}(U) = -\frac{\log(1-U)}{\lambda}.$$

Therefore, the formula to convert uniform random variates to exponential random variates is

$$T = -\frac{\log(1-U)}{\lambda}.$$

Algorithm to Compute Exponential Pseudorandom Numbers

1. Generate a uniform pseudorandom variable U on [0,1].

2. Set

$$T = -\frac{\log(1-U)}{\lambda}.$$

T has an exponential distribution with rate λ.

Example - Exponential Pseudorandom Numbers

Generate a random sample of 20 exponential random numbers with mean 10.

```
lambda <- 1/10  # lambda = 1/mean
U <- runif(20)
T <- -log(1-U)/lambda
T

##  [1]  3.0266  1.1037 11.1095  5.2451  0.9983
##  [6] 20.5270  7.9081  2.7129  9.4855  7.1716
## [11]  0.7501  2.9836 16.1389  1.2238  6.9700
## [16]  0.9728  1.6155 70.1343  0.2521  1.9512
```

This algorithm is built into R.

Exponential Pseudorandom Numbers

The R function rexp() can be used to generate n random exponential variates.

rexp(n, rate)

Example

A bank has a single teller who is facing a lineup of 10 customers. The time for each customer to be served is exponentially distributed with rate 3 per minute. We can simulate the service times (in minutes) for the 10 customers.

```
servicetimes <- rexp(10, rate = 3)
servicetimes
```

```
##  [1] 0.02225 1.89071 0.49009 0.48029 0.16842
##  [6] 0.18249 0.26900 0.32565 1.03268 0.16716
```

```
sum(servicetimes)
```

```
## [1] 5.029
```

The total time until these 10 simulated customers will complete service is 301.7232 **seconds.**

Another way to simulate a Poisson process

It can be shown that the points of a homogeneous Poisson process with rate λ on the line are separated by independent exponentially distributed random variables which have mean $1/\lambda$.

This leads to another way of simulating a Poisson process on the line.

Another way to simulate a Poisson process

Suppose $X_1, X_2, \ldots$ is a sequence of independent random variables with distribution function $F(x) = 1 - e^{-\lambda x}$. Set

$$S_0 = 0$$

$$S_n = \sum_{i=0}^{n} X_i, \qquad n = 1, 2, 3, \ldots$$

and

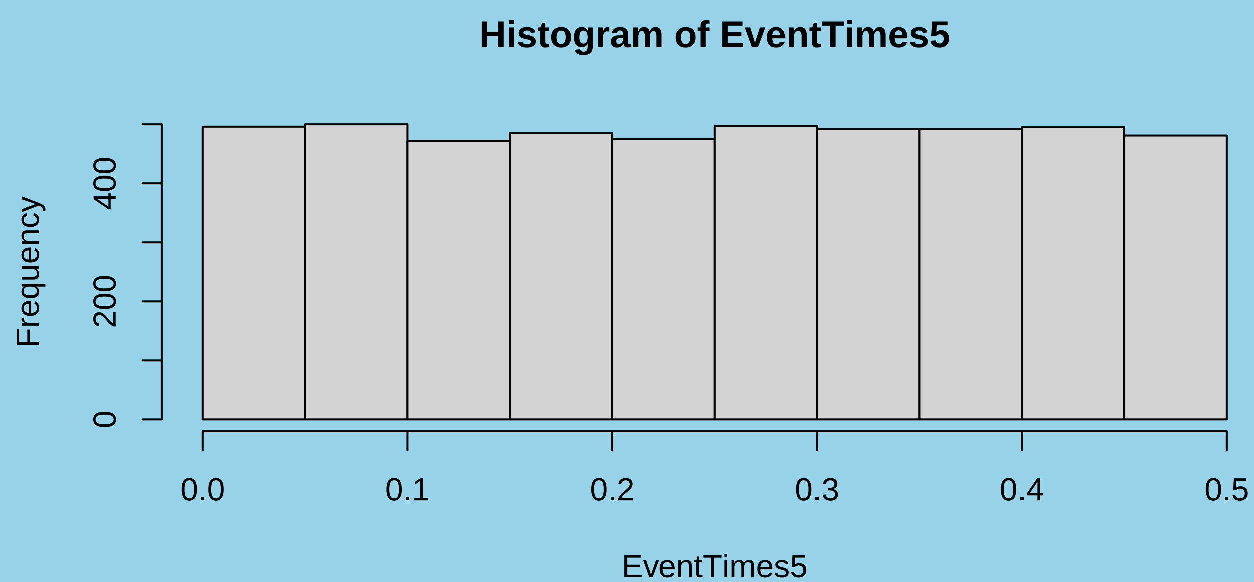
$$N(t) = \max\{n \geq 0 : S_n \leq t\}, \qquad \text{for } t \geq 0.$$

$N(t)$ is called a *Poisson Process* (generated by the sequence $\{X_n\}$).

An Alternative Simulation Method – Use Exponential Numbers

To see that this will work, simulate a Poisson process with a lot of point events:

```
X <- rexp(10000, rate = 10000)  # 10000 events expected in 1 time unit
EventTimes <- cumsum(X)    # Simulating the Poisson events
EventTimes5 <- EventTimes[EventTimes < 0.5]  # Events up to time .5
hist(EventTimes5) # What is the distribution of these event times?
```



It looks essentially like a uniform distribution on $[0, 0.5)$. This is not a coincidence.

Example

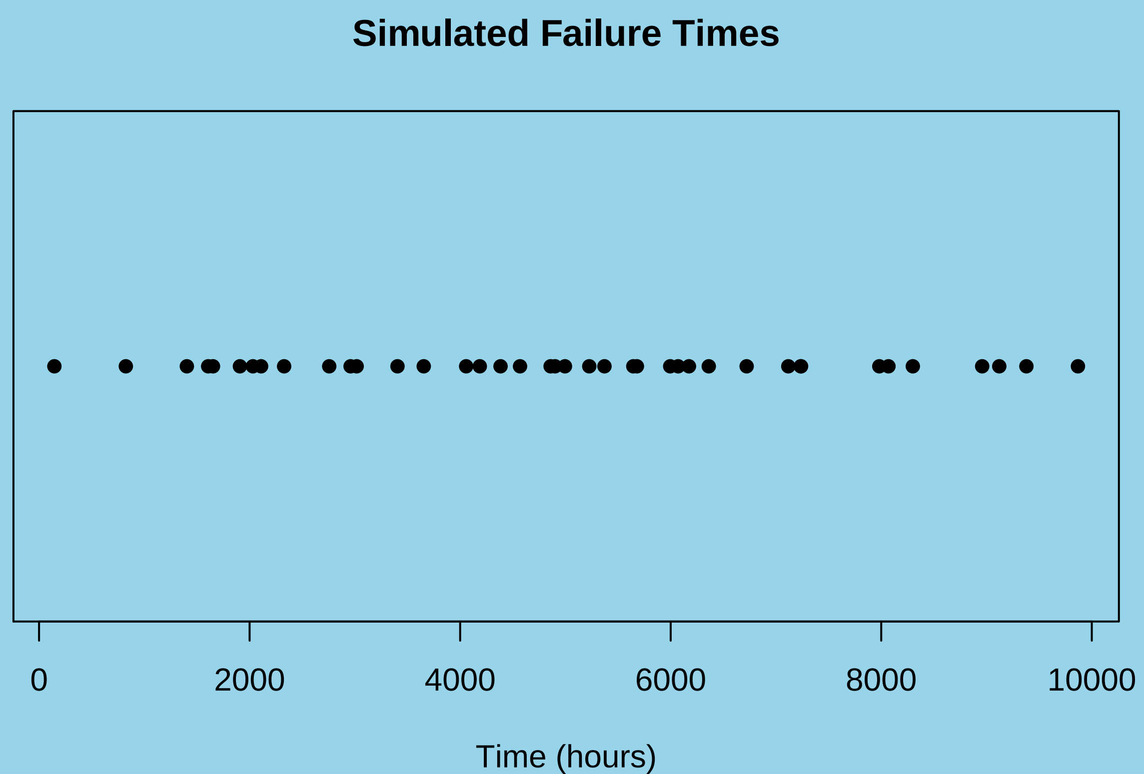
A machine is subject to failures. Suppose the time between the nth and $(n+1)$st failure is X_n, for $n = 1, 2, 3, \ldots$, and the times between failure are independent of each other.

Suppose repairs can be made instantaneously. Then the number of failures by time t, $N(t)$, is a Poisson process if and only if $\{X_n\}$ is exponentially distributed with same mean.

Example

Suppose the mean of X_n is 200 hours. This means that the average rate of failure is 1/200. Simulate the next 40 failure times for such a machine.

```
X <- rexp(40, rate = 1/200)    # The X's
FailureTimes <- cumsum(X)    # Cumulative sums of the X's
stripchart(FailureTimes, pch=16, xlab="Time (hours)",
    main="Simulated Failure Times")
```



Normal random variables

A normal random variable X has a probability density function given by

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

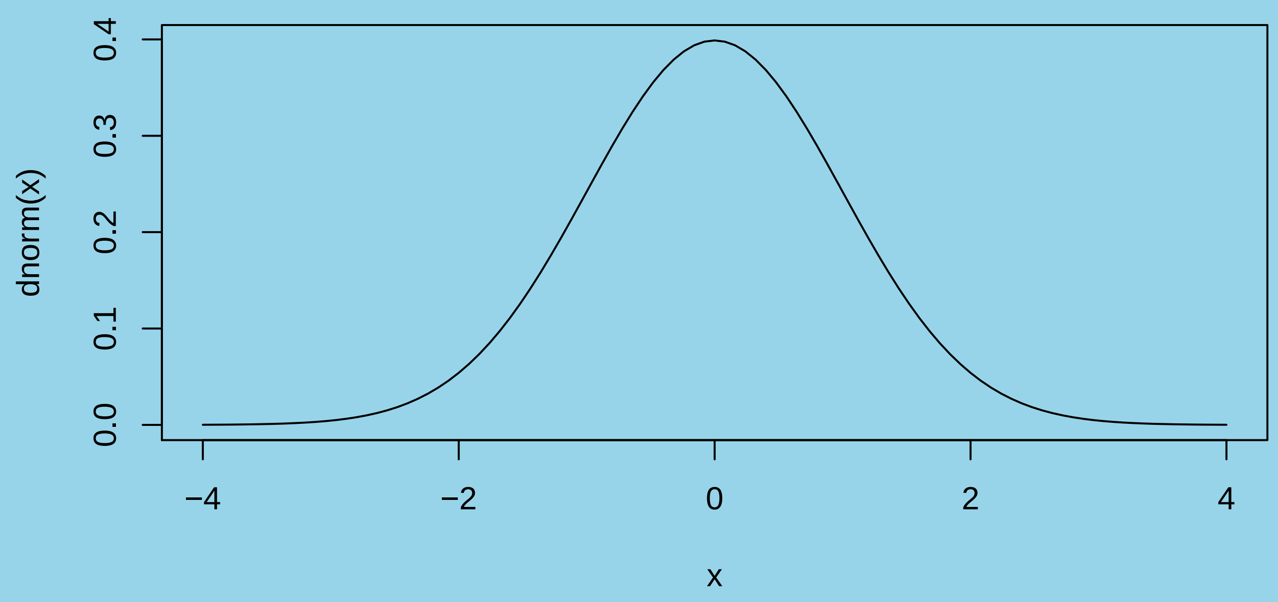
where μ is the expected value of X, and σ^2 denotes the variance of X.

The *standard normal* random variable Z has expected value $\mu = 0$ and standard deviation $\sigma = 1$.

Normal random variables

The normal density function can be evaluated using the dnorm() function:

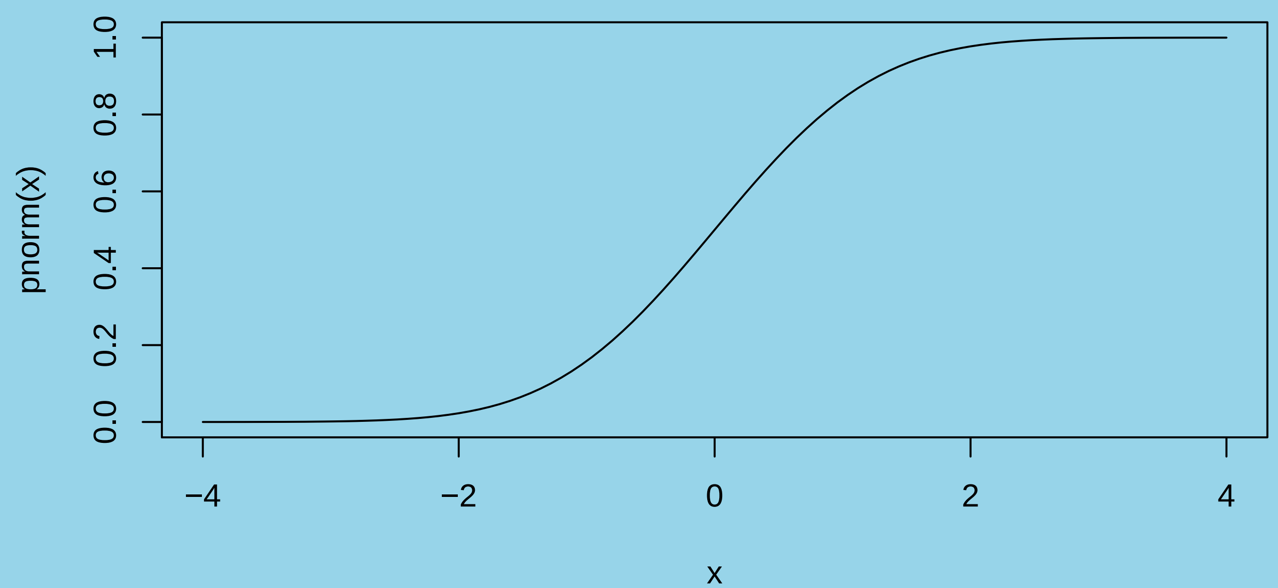
```r
curve(dnorm(x), -4, 4)
```



Normal random variables

The distribution function can be evaluated using pnorm():

```r
curve(pnorm(x), -4, 4)
```



Normal random variables

Exercise:

Find the probability that a standard normal random variable exceeds 1.96.

```r
1 - pnorm(1.96)
```

```
## [1] 0.025
```

Normal random variables

Exercise:

Find the probability that a normal random variable with mean 7.0 and standard deviation 3 takes a value between 5 and 8.

```r
pnorm(8, mean=7, sd=3)  - pnorm(5, mean = 7, sd = 3)

## [1] 0.3781
```

Normal random variables

The quantiles of the normal distribution can be obtained using qnorm().

For example, the 95th percentile of the normal distribution with mean 2.7 and standard deviation 3.3 is:

```r
qnorm(0.95, mean = 2.7, sd = 3.3)
```

```
## [1] 8.128
```

Normal Pseudorandom Numbers

Normal pseudorandom variables can be generated using the rnorm() function in R.

```
rnorm(n, mean, sd)
```

This produces n normal pseudorandom variates which have mean `mean` and standard deviation `sd`.

Example

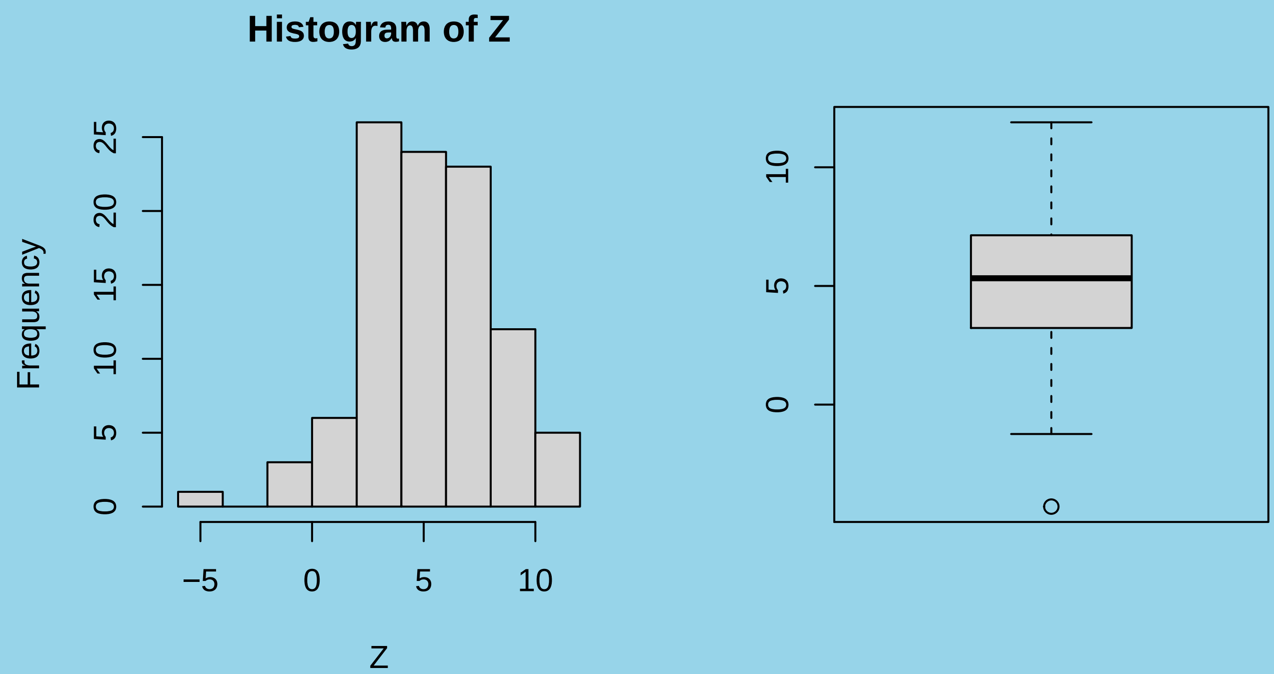
We can simulate 10 independent normal variates with a mean of -3 and standard deviation of 0.5 using

```r
rnorm(10, -3, 0.5)
```

```
##  [1] -2.635 -2.974 -2.984 -3.379 -2.914 -3.133
##  [7] -4.158 -3.603 -3.037 -2.266
```

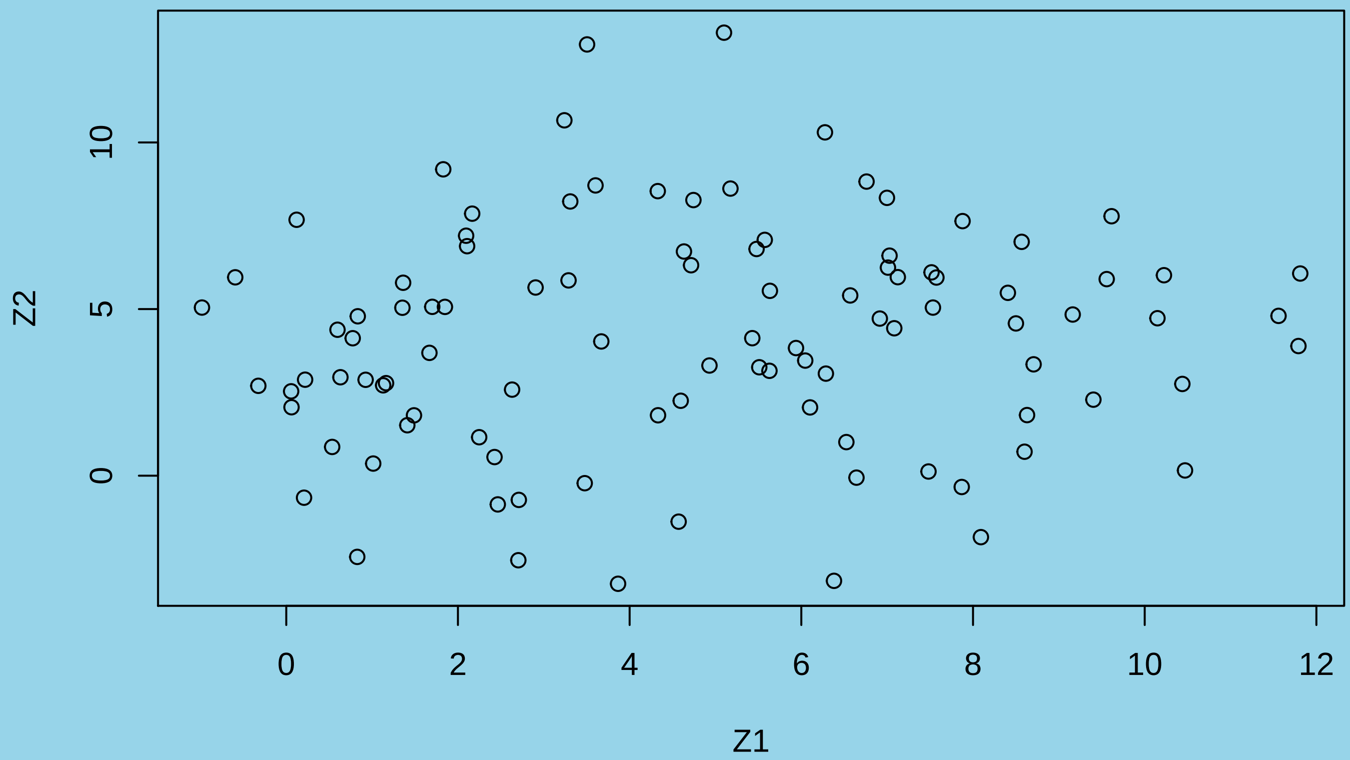
Visualizing Normally Distributed Data

```r
Z <- rnorm(100, mean=5, sd = 3)
par(mfrow=c(1, 2))
hist(Z)
boxplot(Z)
```



Visualizing Normally Distributed Data - Independent Samples

```r
Z1 <- rnorm(100, mean=5, sd = 3)
Z2 <- rnorm(100, mean=5, sd = 3)
plot(Z2 ~ Z1)
```

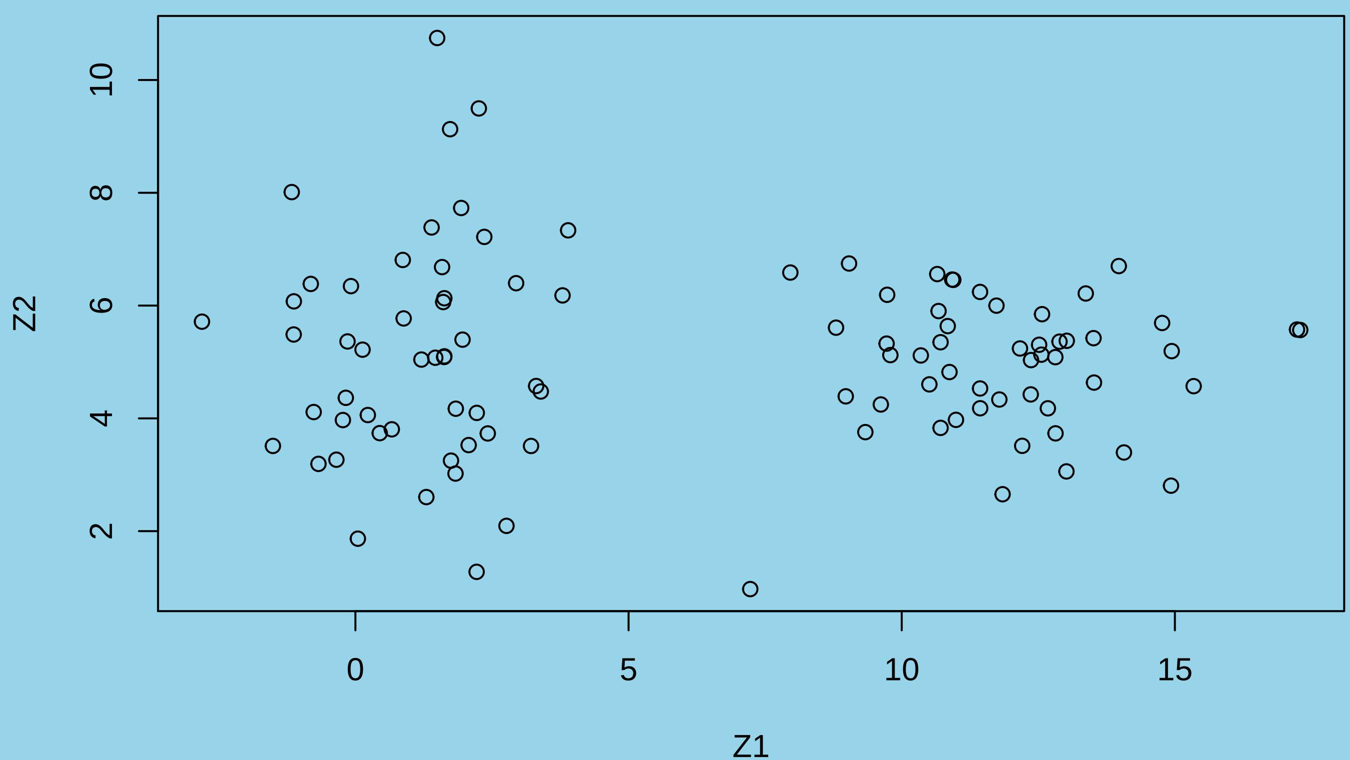


This is a special case of *bivariate* normal data.

Visualizing Bivariate Normal Data

Different means and standard deviations provide a mechanism for clustering:

```
Z1 <- rnorm(100, mean=c(rep(1, 50), rep(12, 50)), sd = 2)
Z2 <- rnorm(100, mean=5, sd = c(rep(2, 50), rep(1, 50)))
plot(Z2 ~ Z1)
```

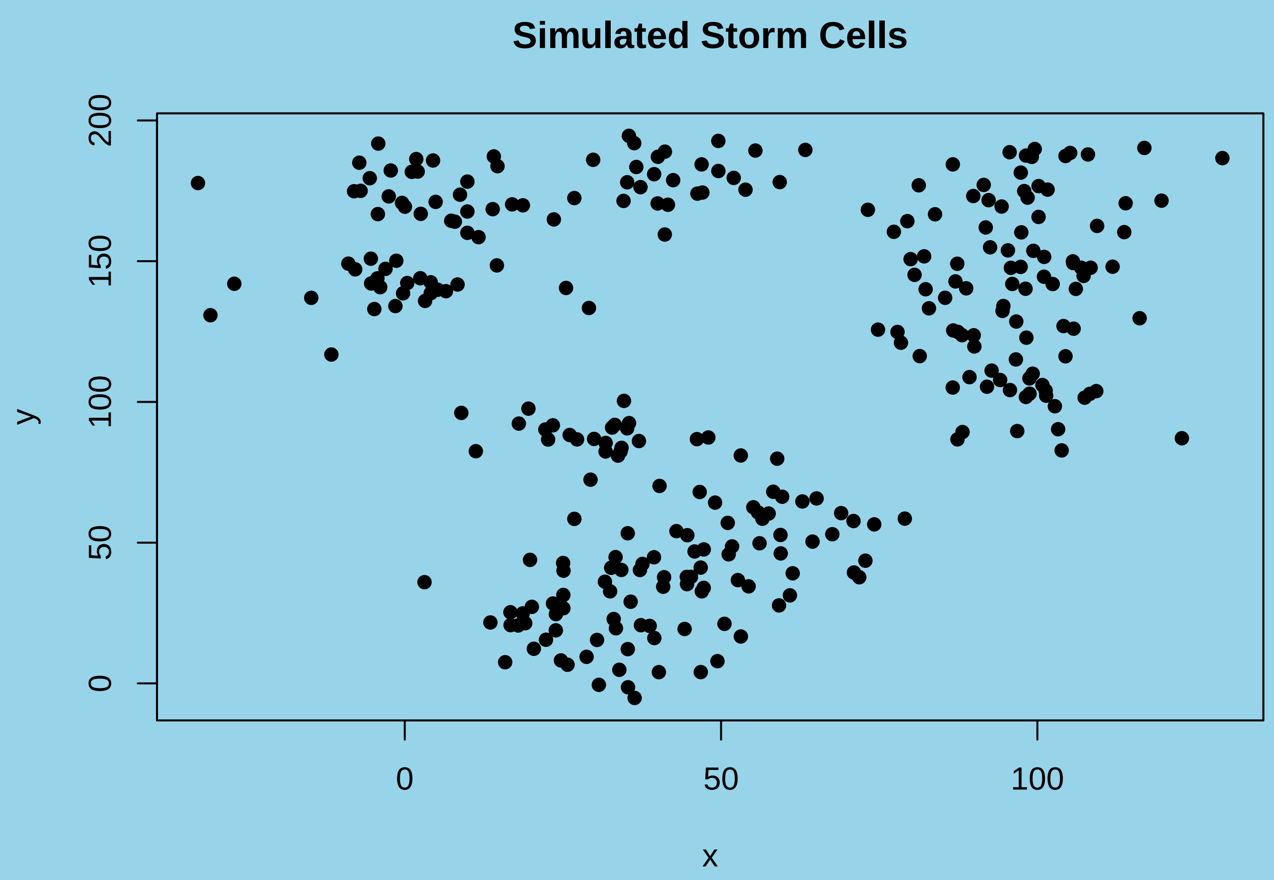


Poisson Cluster Process in 3 Dimensions

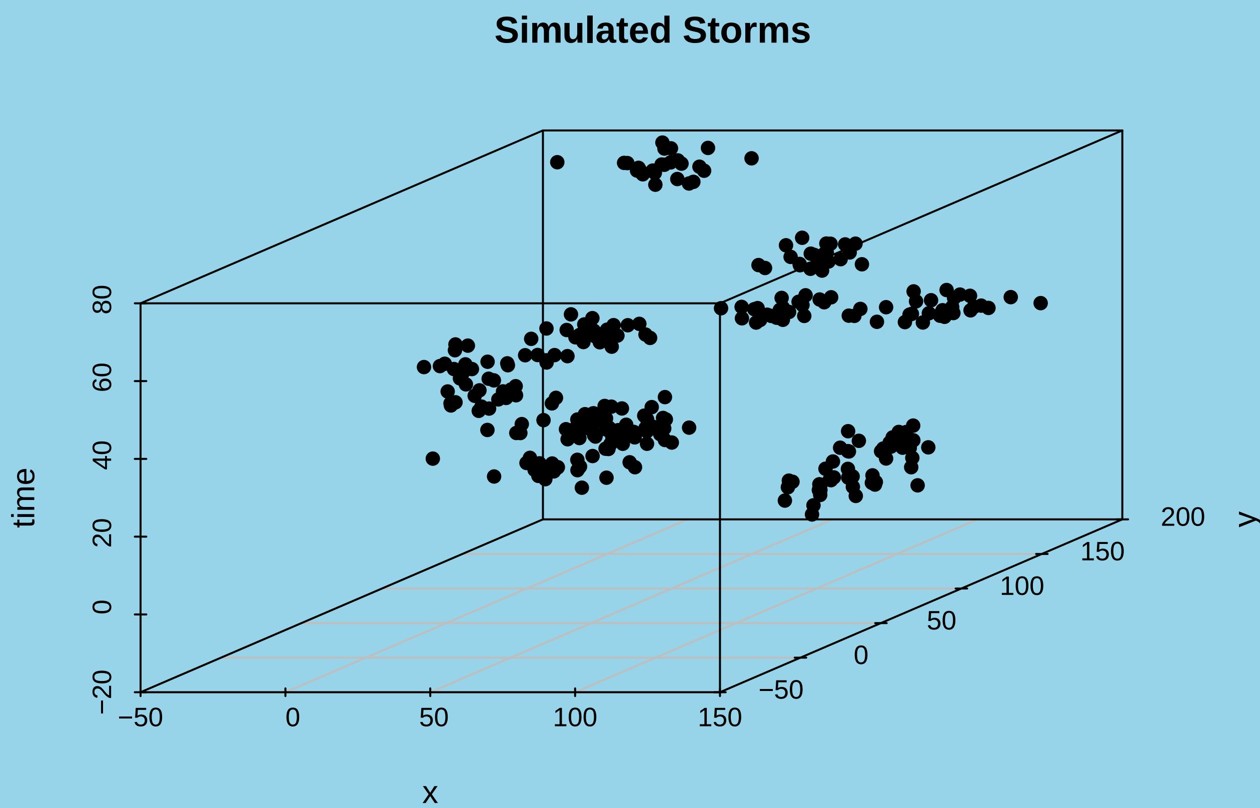
```r
set.seed(13327)
N <- rpois(1, 10) # Number of clusters
U1 <- runif(N, min=0, max=100)
U2 <- runif(N, min=0, max=200)
times <- runif(N, min = 0, max = 120)

M <- rpois(N, 25)  # numbers of points in each cluster
storms1 <- NULL
storms2 <- NULL
stormtimes <- NULL
for (i in 1:N) {
  storms1 <- c(storms1, rnorm(M[i], mean=U1[i], sd=10))
  storms2 <- c(storms2, rnorm(M[i], mean=U2[i], sd=10))
  stormtimes <- c(stormtimes, rnorm(M[i], mean = times[i],
   sd = 2))
}
```

Poisson Cluster Process in 3 Dimensions

Aerial View:



3-D View


Simulated Storms
time
80
60
40
20
0
−20
−50
0
50
100
150
x
200
150
100
50
0
−50
y

What to Take Away from this Lecture

You should be able to:

- understand that point event data can be modelled and simulated by the Poisson process and related extensions.
- calculate expected values, variances and probabilities of counts of points in intervals and regions for a Poisson process model in any number of dimensions.
- understand the concept of the exponential distribution and how to simulate from it.
- understand the concept of the normal distribution and how to simulate from it.
- appreciate that relatively sophisticated models can be built up from simple components.

New R Functions

```
dexp()
pexp()
rexp()

dnorm()
pnorm()
rnorm()
qnorm()    # percentiles (quantiles)

stripchart()
curve()
boxplot()

sort()
log()
cumsum()
```