```
library(spatstat)
```

```
##
## spatstat 1.42-2       (nickname: 'Barking at Balloons')
## For an introduction to spatstat, type 'beginner'
```

```
library(spptrees)
```

Spatial Point Processses and the Poisson Process

Informal Definitions and Comments

A *spatial point process*, subsequently referred to as a point process, is a stochastic mechanism which generates a countable set of events in the plane ($\mathbb{R}^2$). Simple point processes are typically *stationary* and *isotropic*. The properties of a stationary point process are invariant under translation, while the properties of an isotropic point process are invariant under rotation. These definitions are less restrictive than they might seem. For instance, they allow for random heterogeneity in the underlying environment. Though defined over the entire plane, spatial point processes are only applied to data from finite regions. Furthermore, in practice, models assuming isotropy or stationarity are typically appropriate for point processes that exhibit approximate stationarity or isotropy.

Statistical analysis for spatial point pattern data typically involves comparisons between empirical summary descriptions of the data and the corresponding theoretical summary descriptions of a model. Importantly, these theoretical summary descriptions must be derived from an underlying model, rather than being models themselves. The most common comparisons pit the empirical summary descriptions against the corresponding summary description under a homogeneous poisson process.

Point Processes

A *spatial point process* X is a random countable subset of a space S. In this thesis, S will always be a subset of $\mathbb{R}^d$ and d will typically be 2. Often S will either be a d-dimensional box or all of $\mathbb{R}^d$. In pratice, we only observe the points in an observation window $W \subseteq S$.

I restrict my attention to point processes X whose realisations are locally finite subsets of S. For any subset $x \subset S$, let $n(x)$ denote the cardinality of x, setting $n(x) = \infty$ if x is not finite. x is said to be *locally finite*, if $n(x_B)$ is finite whenever $B \subset S$ is bounded, where

$$x_B = x \cap B$$

is the restriction of a point configuration x to B. Consequently, X takes values in the space defined by

$$N_{lf} = \{x \subseteq S | n(x_B) < \infty \text{ for all bounded } B \subseteq S\}.$$

Elements of N_{lf} are called *locally finite point configurations*, and they will be denoted by $x, y, \ldots$ while $\xi, \eta, \ldots$ will denote points in S. I will typically write $x \cup \xi$ for $x \cup \{\xi\}$, $x \setminus \eta$ for $s \setminus \{\eta\}$, when $x \in N_{lf}$ and $\xi, \eta \in S$.

Poisson Point Processes

The Poisson process plays a central role in the point process literature. It serves as the tractable model for "completely spatially random" point processes. Real processes which exhibit complete spatial randomnesss are undoubtebly rare. However, by comparing empirical summary statistics to those under a theorhetical Poisson, statisticians and scientists are able to detect clustering, regularity, and inhomogeneity in the underlying environments. Researchers typically begin data analysis with these comparisons.

Basic Properties

First, lets consider a Poisson point process defined on a space $S \subseteq \mathbb{R}^d$ and specified by a *intensity functon* $\rho : S \to [0, \infty]$ which is *locally integrable*, i.e. $\int_B \rho(\xi) d\xi < \infty$ for all bounded $B \subseteq S$. For the following definition, we use the *intensity measure* μ defined by

$$\mu(B) = \int_B \rho(\xi) d\xi, \quad B \subseteq S.$$

Clearly, this measure is locally finite. It is also *diffuse*, i.e. $\mu(\{\xi\}) = 0$ for all $\xi \in S$.

Definition: Let f be a density function on a set $B \subseteq S$, and let $n \in \mathbb{N}$. A point process X consisting of n points in B with density f is called a *binomial point process* of n points in B with density f. We write $X \sim \text{binomial}(B, n, f)$ ('$\sim$' means "distributed as").

Clearly, B in the above distribution has strictly positive volume.

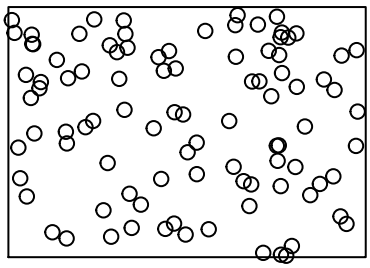
Definition A point process X on S is a *Poisson point process* with intensity function ρ if the following properties are satisfied: 1. For any $B \subseteq S$ with $\mu(B) < \infty$, $N(B) \sim \text{poisson}(\mu(B))$, the Poisson distribution with mean $\mu(B)$. 2. For any $n \in \mathbb{N}$ and $B \subseteq S$ wih $0 \leq \mu(B) < \infty$, conditional on $N(B) = n$, $X_B \sim \text{binomial}(B, n, F)$ with $f(\xi) = \rho(\xi)/\mu(B)$ We then write $X \sim \text{Poisson}(S, \rho)$.

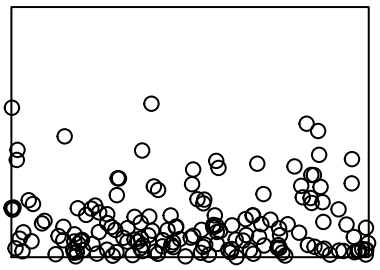
Consequently, μ determines the expected number of points in any $B \subseteq S$. $\rho(\xi)d\xi$ can be thought of as the proability for the occurance of an event in an infinitesimally small ball with center ξ and volume $d\xi$.

Definition: If ρ is constant, the process $\text{Poisson}(S, \rho)$ is called a *homogeneous Poisson process* on S with *rate* or *intensity* ρ; otherwise it is an *inhomoheneous Poissson processs on S.

Figure 1.1 displays examples of both homogeneous and inhomogeneous Poisson processes. Homogeneous Poisson processes are both stationary and isotropic.

Definition: A point process X on $\mathbb{R}^d$ is *stationary* if its distribution is translation invariant. In other words, if the distribution of $X + s = \{\xi + s : \xi \in X\}$ is the same as X for any $s \in \mathbb{R}^d$. X is *isotropic* if its distribution is rotation invariant about the origin.





Sometimes (ii) is replaced with the condition that $N(B_1), N(B_2), \ldots N(B_n)$ are independent for disjoint sets $B_1, B_2, \ldots, B_n \subseteq S$ and $n \geq 2$. Thisis called *independent scattering*. The following expansion is often useful.

Proposition 1.1:

(i) $X \sim \text{Poisson}(S, \rho)$ if and only if for all $B \subseteq S$ with $\mu(B) = \int_S \rho(\xi) d\xi < \infty$ and all $F \subseteq N_{lf}$,

$$P(X_B \in F) = \sum_{n=0}^{\infty} \frac{exp(-\mu(B))}{n!} \int_B \cdots \int_B \mathbf{1}[\{x_1, x_2, \ldots, x_n\} \in F] \prod_{i=1}^{n} \rho(x_i) dx_1 dx_1 \ldots dx_n$$

where the integral for $n = 0$ is $\mathbf{1}[\emptyset \in F]$.

(ii) If $X \sim \text{Poisson}(S, \rho)$, then for functions $h : N_{lf} \to [0, \infty)$ and $B \subseteq S$ with $\mu(B) < \infty$,

$$\mathbb{E}[h(X_B)] = \sum_{n=0}^{\infty} \frac{exp(-\mu(B))}{n!} \int_B \cdots \int_B h(\{x_1, \ldots, x_n\}) \prod_{i=1}^{n} dx_1 \cdots dx_n.$$

A second proposition demonstrates the validity of the independent scattering property of Poisson processes.

Proposition 1.2: If X is a Poisson process on S, then $X_{B_1}, X_{B_2}, \ldots$ are independent for disjoint sets $B_1, B_2, \cdots \subseteq S$.

Proof: Suppose X is a Poisson process on S and lets $B_1, \ldots, B_n \subseteq S$ be disjoint where $n \geq 2$. First, suppoes $n = 2$.

Superpositioning and Thinning

What follows are two important operations for point processes. They will be an important tool for model fitting later in the thesis.

Definition: A disjoint union $\cup_{i=1}^{\infty} X_i$ of point processes $X_1, X_2, \ldots$ is called a *superposition.*

Definition: Let $p : S \to [0, 1]$ be a function and X a point process on S. The point process $X_{\text{thin}} \subseteq X$ obtained by including $\xi \in X$ in X_{thin} with probability $p(\xi)$, where points are included/excluded independently of each other, is said to be an *independent thinning* of X with *retention probabilities* $p(\xi)$, $\xi \in S$. Formally,

$$X_{\text{thin}} = \{\xi \in X : \mathcal{R}(\xi) \leq p(\xi)\}$$

where $\mathcal{R}(\xi) \sim \text{Uniform}[0, 1]$, $\xi \in S$, are mutually independent and independent of X.

The following to propositions demostrate that the class of Poisson processes is closed under superpositioning and independent thinning.

Lists

It's easy to create a list. It can be unordered like

- Item 1
- Item 2

or it can be ordered like

1. Item 1
2. Item 2

Notice that I intentionally mislabeled Item 2 as number 4. *Markdown* automatically figures this out! You can put any numbers in the list and it will create the list. Check it out below.

To create a sublist, just indent the values a bit (at least four spaces or a tab). (Here's one case where indentation is key!)

1. Item 1
2. Item 2
3. Item 3
    - Item 3a
    - Item 3b

Line breaks

Make sure to add white space between lines if you'd like to start a new paragraph. Look at what happens below in the outputted document if you don't:

Here is the first sentence. Here is another sentence. Here is the last sentence to end the paragraph. This should be a new paragraph.

Now for the correct way:

Here is the first sentence. Here is another sentence. Here is the last sentence to end the paragraph.

This should be a new paragraph.

R chunks

When you click the **Knit** button above a document will be generated that includes both content as well as the output of any embedded **R** code chunks within the document. You can embed an **R** code chunk like this (`cars` is a built-in **R** dataset):

```
summary(cars)
```

```
##      speed           dist
##  Min.   : 4.0   Min.   :  2.00
##  1st Qu.:12.0   1st Qu.: 26.00
##  Median :15.0   Median : 36.00
##  Mean   :15.4   Mean   : 42.98
##  3rd Qu.:19.0   3rd Qu.: 56.00
##  Max.   :25.0   Max.   :120.00
```

Inline code

If you'd like to put the results of your analysis directly into your discussion, add inline code like this:

> The `cos` of 2π is 1.

Another example would be the direct calculation of the standard deviation:

> The standard deviation of `speed` in `cars` is 5.2876444.

One last neat feature is the use of the `ifelse` conditional statement which can be used to output text depending on the result of an **R** calculation:

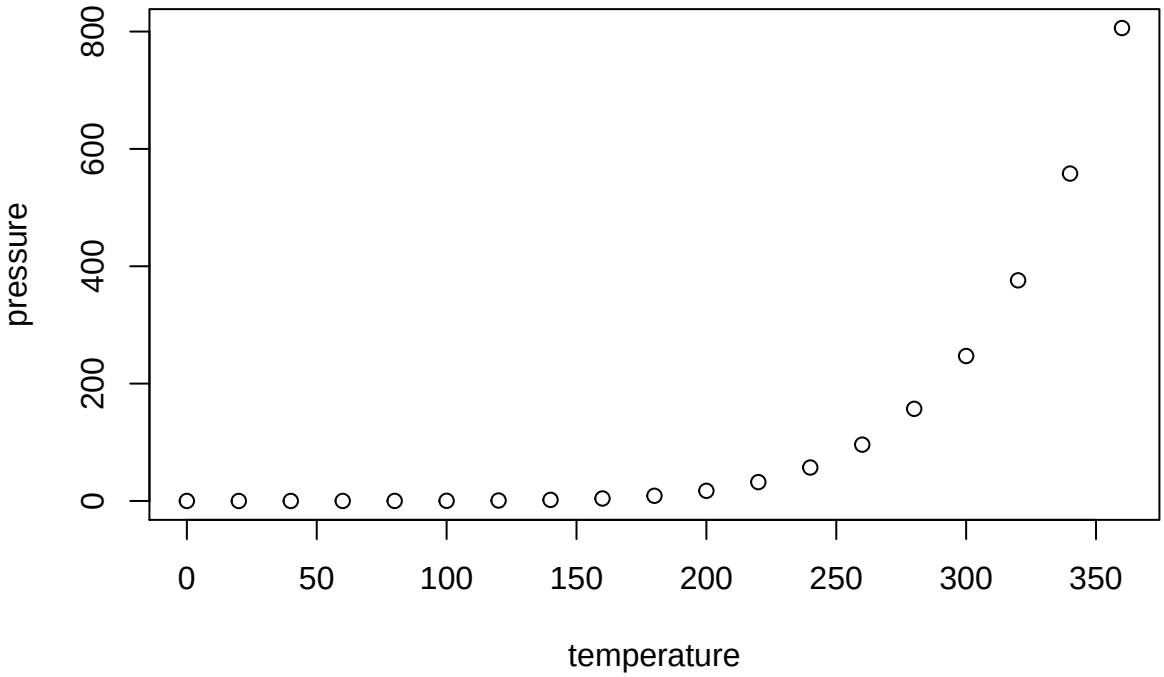
> The standard deviation is less than 6.

Note the use of `>` here, which signifies a quotation environment that will be indented.

As you see with `$2 \pi$` above, mathematics can be added by surrounding the mathematical text with dollar signs. More examples of this are in [Mathematics and Science] if you uncomment the code in [Math].

Including plots

You can also embed plots. For example, here is a way to use the base **R** graphics package to produce a plot using the built-in `pressure` dataset:



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the **R** code that generated the plot. There are plenty of other ways to add chunk options. More information is available at http://yihui.name/knitr/options/.

Another useful chunk option is the setting of `cache = TRUE` as you see here. If document rendering becomes time consuming due to long computations or plots that are expensive to generate you can use knitr caching to improve performance. Later in this file, you'll see a way to reference plots created in **R** or external figures.

Loading and exploring data

Included in this template is a file called `flights.csv`. This file includes a subset of the larger dataset of information about all flights that departed from Seattle and Portland in 2014. More information about this dataset and its **R** package is available at http://github.com/ismayc/pnwflights14. This subset includes only Portland flights and only rows that were complete with no missing values. Merges were also done with the `airports` and `airlines` data sets in the `pnwflights14` package to get more descriptive airport and airline names.

We can load in this data set using the following command:

```
flights <- read.csv("data/flights.csv")
```

The data is now stored in the data frame called `flights` in **R**. To get a better feel for the variables included in this dataset we can use a variety of functions. Here we can see the dimensions (rows by columns) and also the names of the columns.

```
dim(flights)
```

```
## [1] 52808    16
```

```
names(flights)
```

```
##  [1] "month"        "day"          "dep_time"     "dep_delay"
##  [5] "arr_time"     "arr_delay"    "carrier"      "tailnum"
##  [9] "flight"       "dest"         "air_time"     "distance"
## [13] "hour"         "minute"       "carrier_name" "dest_name"
```

Another good idea is to take a look at the dataset in table form. With this dataset having more than 50,000 rows, we won't explicitly show the results of the command here. I recommend you enter the command into the Console ***after*** you have run the **R** chunks above to load the data into **R**.

```
View(flights)
```

While not required, it is highly recommended you use the `dplyr` package to manipulate and summarize your data set as needed. It uses a syntax that is easy to understand using chaining operations. Below I've created a few examples of using `dplyr` to get information about the Portland flights in 2014. You will also see the use of the `ggplot2` package, which produces beautiful, high-quality academic visuals.

We begin by checking to ensure that needed packages are installed and then we load them into our current working environment:

```
# List of packages required for this analysis
pkg <- c("dplyr", "ggplot2", "knitr", "devtools")
# Check if packages are not installed and assign the
# names of the packages not installed to the variable new.pkg
new.pkg <- pkg[!(pkg %in% installed.packages())]
# If there are any packages in the list that aren't installed,
# install them
if (length(new.pkg))
  install.packages(new.pkg, repos = "http://cran.rstudio.com")
# Load packages
library(dplyr)
library(ggplot2)
library(knitr)
```

The example we show here does the following:

- Selects only the `carrier_name` and `arr_delay` from the `flights` dataset and then assigns this subset to a new variable called `flights2`.
- Using `flights2`, we determine the largest arrival delay for each of the carriers.

```
flights2 <- flights %>% select(carrier_name, arr_delay)
max_delays <- flights2 %>% group_by(carrier_name) %>%
  summarize(max_arr_delay = max(arr_delay, na.rm = TRUE))
```

We next introduce a useful function in the `knitr` package for making nice tables in *R Markdown* called `kable`. It produces the LaTeX code required to make the table and is much easier to use than manually entering values into a table by copying and pasting values into Excel or LaTeX. This again goes to show how nice reproducible documents can be! There is no need to copy-and-paste values to create a table. (Note the use of `results = "asis"` here which will produce the table instead of the code to create the table. You'll learn more about the `label` later.)

```
kable(max_delays, col.names = c("Airline", "Max Arrival Delay"),
      caption = "Max Delays by Airline label{tab:max_delay}")
```

Table 1: Max Delays by Airline label{tab:max_delay}

Airline	Max Arrival Delay
Alaska Airlines Inc.	338
American Airlines Inc.	1539
Delta Air Lines Inc.	651
Frontier Airlines Inc.	575
Hawaiian Airlines Inc.	407
JetBlue Airways	273
SkyWest Airlines Inc.	421
Southwest Airlines Co.	694
United Air Lines Inc.	472
US Airways Inc.	347
Virgin America	366

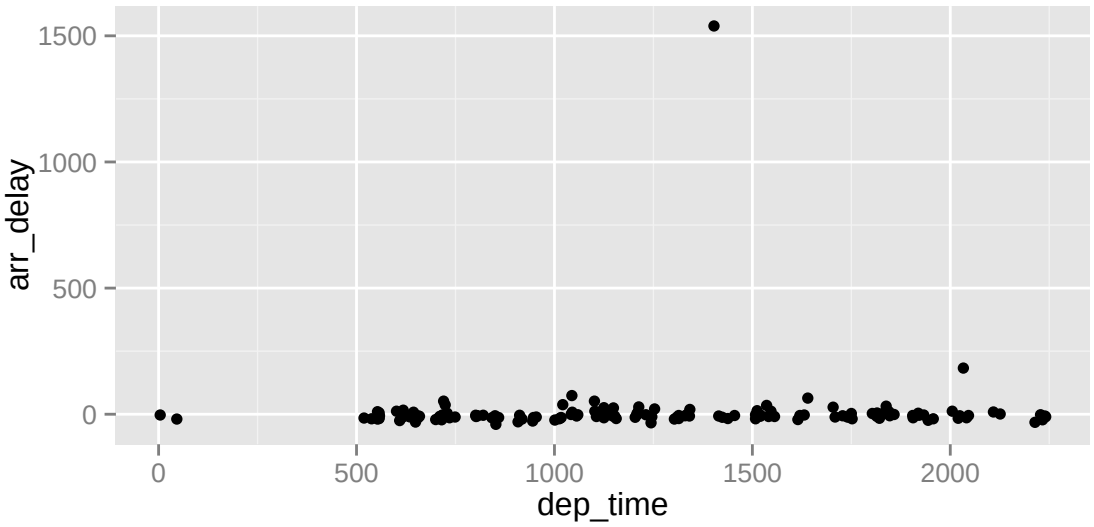
We can further look into the properties of the largest value here for American Airlines Inc. To do so, we can isolate the row corresponding to the arrival delay of 1539 minutes for American in our original `flights` dataset.

```
flights %>% filter(arr_delay == 1539,
                   carrier_name == "American Airlines Inc.") %>%
  select(-c(month, day, carrier, dest_name, hour,
            minute, carrier_name, arr_delay))
```

```
##   dep_time dep_delay arr_time tailnum flight dest air_time distance
## 1     1403      1553     1934  N595AA   1568  DFW      182     1616
```

We see that the flight occurred on March 3rd and departed a little after 2 PM on its way to Dallas/Fort Worth. Lastly, we show how we can visualize the arrival delay of all departing flights from Portland on March 3rd against time of departure.

```
flights %>% filter(month == 3, day == 3) %>%
  ggplot(aes(x = dep_time, y = arr_delay)) +
  geom_point()
```



Additional resources

- *Markdown* Cheatsheet - https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet
- *R Markdown* Reference Guide - https://www.rstudio.com/wp-content/uploads/2015/03/rmarkdown-reference.pdf
- Introduction to `dplyr` - https://cran.rstudio.com/web/packages/dplyr/vignettes/introduction.html
- `ggplot2` Documentation - http://docs.ggplot2.org/current/ F