Descriptive Epidemiology using epiR

Mark Stevenson and Evan Sergeant

12 Nov 2026

Contents

1 Individual 1
2 Time 4
3 Place 11
References 14

Epidemiology is the study of the occurrence and distribution of health-related events, states, and processes in specified populations, including the study of the determinants influencing such processes, and the application of this knowledge to control relevant health problems (Centers for Disease Control and Prevention 2006; Porta, Greenland, and Last 2014).

This vignette provides instruction on the use of R and `epiR` for descriptive epidemiological analyses, that is, to describe how the frequency of disease varies by individual, place and time.

1 Individual

The frequency of disease can be reported in terms of either **prevalence** or **incidence**.

Two types of prevalence are reported in the literature: (1) **point prevalence** equals the proportion of a population in a diseased state at a single point in time; (2) **period prevalence** equals the proportion of a population with a given disease or condition over a specific period of time (i.e., the number of existing cases at the start of a follow-up period plus the number of incident cases that occur during the follow-up period).

Incidence provides a measure of how frequently susceptible individuals become disease cases as they are observed over time. An incident case occurs when an individual changes from being susceptible to being diseased. The count of incident cases is the number of such events that occur in a population over a defined follow-up period. There are two ways to express incidence:

Incidence risk (also known as cumulative incidence) is the proportion of initially susceptible individuals in a population that become new cases over a defined follow-up period.

Incidence rate (also known as incidence density) is the number of new cases of disease that occur per unit of individual time at risk over a defined follow-up period.

In addition to reporting the point estimate of disease frequency, it is important to provide an indication of the uncertainty around that point estimate. The `epi.conf` function in the `epiR` package allows you to calculate confidence intervals for prevalence, incidence risks and incidence rates.

Let's say we're interested in the prevalence of disease X in a population comprised of 1000 individuals. Two hundred are tested and four returned a positive result. Assuming 100% diagnostic test sensitivity and specificity, what is the estimated prevalence of disease X in this population?

```
library(epiR); library(ggplot2); library(scales); library(zoo)

ncas <- 4; npop <- 200
tmp <- as.matrix(cbind(ncas, npop))
epi.conf(tmp, ctype = "prevalence", method = "exact", N = 1000, design = 1,
   conf.level = 0.95) * 100
#>   est     lower    upper
#> 1   2 0.5475566 5.041361
```

The estimated prevalence of disease X in this population is 2.0 (95% confidence interval [CI] 0.55 to 5.0) cases per 100 individuals at risk.

Another example. A study was conducted by Feychting, Osterlund, and Ahlbom (1998) to report the frequency of cancer among the blind. A total of 136 diagnoses of cancer were made from 22,050 person-years at risk. What was the incidence rate of cancer in this population?

```
ncas <- 136; ntar <- 22050
tmp <- as.matrix(cbind(ncas, ntar))
epi.conf(tmp, ctype = "inc.rate", method = "exact", N = 1000, design = 1,
   conf.level = 0.95) * 1000
#>         est    lower    upper
#> ncas 6.1678 5.174806 7.295817
```

The incidence rate of cancer in this population was 6.2 (95% CI 5.2 to 7.3) cases per 1000 person-years at risk.

We now want to compare the frequency of disease across several populations. An effective way to do this is to use a ranked error bar plot. With a ranked error bar plot the points represent the point estimate of the measure of disease frequency and the error bars indicate the 95% confidence interval around each estimate. With a ranked error bar plot the disease frequency estimates are plotted from lowest to highest. Generate some data:

```
ncas <- c(347,444,145,156,56,618,203,113,10,30,663,447,213,52,256,216,745,97,
   31,250,430,494,96,544,352)
npop <- c(477,515,1114,625,69,1301,309,840,68,100,1375,1290,1289,95,307,354,
   1393,307,35,364,494,1097,261,615,508)
rname <- paste("Region ", 1:length(npop), sep = "")
dat.df <- data.frame(rname,ncas,npop)
```

Calculate the prevalence of disease in each region and its 95% confidence interval. The function `epi.conf` provides several options for confidence interval calculation methods for prevalence. For this example we'll use the exact method:

```
tmp <- as.matrix(cbind(dat.df$ncas, dat.df$npop))
tmp <- epi.conf(tmp, ctype = "prevalence", method = "exact", N = 1000,
   design = 1, conf.level = 0.95) * 100
dat.df <- cbind(dat.df, tmp)
head(dat.df)
#>      rname ncas npop      est    lower    upper
#> 1 Region 1  347  477 72.74633 68.51271 76.69532
#> 2 Region 2  444  515 86.21359 82.93082 89.07325
#> 3 Region 3  145 1114 13.01616 11.09506 15.13489
#> 4 Region 4  156  625 24.96000 21.61207 28.54645
#> 5 Region 5   56   69 81.15942 69.93958 89.56878
#> 6 Region 6  618 1301 47.50192 44.75821 50.25695
```

Sort the data in order of variable `est`, assign a 1 to `n` identifier as variable `rank` and make a copy of `dat.df$rname`:

```
dat.df <- dat.df[sort.list(dat.df$est),]
dat.df$rank <- 1:nrow(dat.df)
dat.df$labels <- dat.df$rname
```

Create a ranked error bar plot, rotating the horizontal axis labels by 90 degrees as shown in Figure 1.

```
ggplot(data = dat.df, aes(x = rank, y = est)) +
  theme_bw() +
  geom_errorbar(aes(ymin = lower, ymax = upper), width = 0.1) +
  geom_point() +
  scale_x_continuous(limits = c(0,25), breaks = dat.df$rank,
     labels = dat.df$labels, name = "Region") +
  scale_y_continuous(limits = c(0,100),
     name = "Cases per 100 individuals at risk") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```

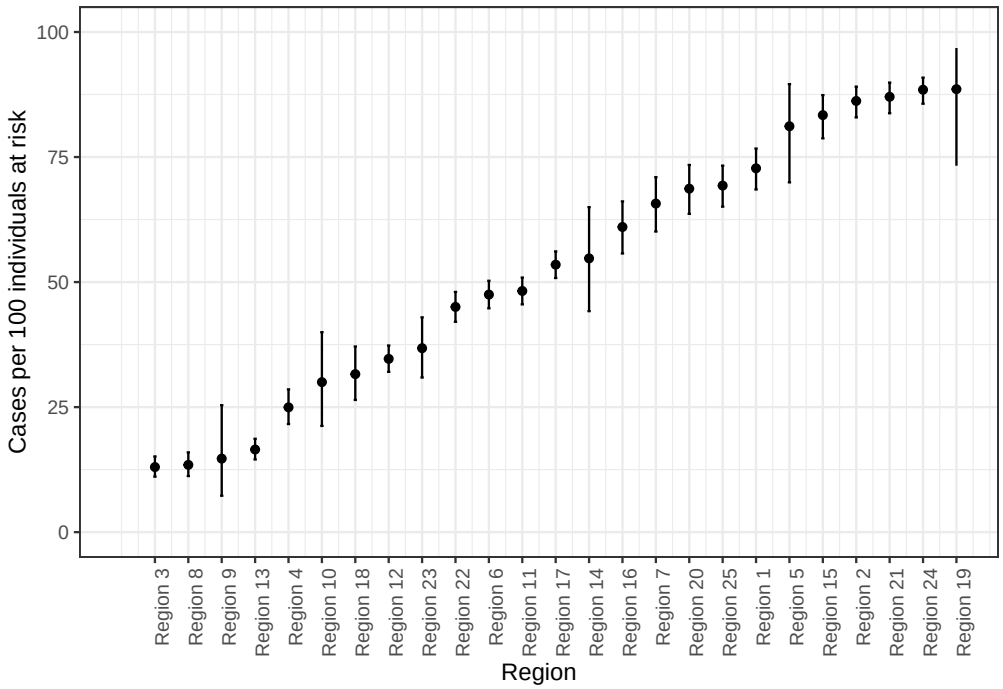


Figure 1: Ranked error bar plot showing the number of prevalent cases of disease (and its 95% confidence interval) for every 100 population units.

If you have a large number of regions the horizontal axis labels can become crowded and difficult to read. Use the `ndelete` function to drop every `n`th region name.

```
ndelete <- function(x, n){
  id <- seq(from = 1, to = length(x), by = n)
  rval <- rep("", times = length(x))
  rval[id] <- x[id]
  rval
}

dat.df$labels <- ndelete(x = dat.df$rname, n = 2)
```

Re-draw the ranked error bar plot (Figure 2):

```r
ggplot(data = dat.df, aes(x = rank, y = est)) +
  theme_bw() +
  geom_errorbar(aes(ymin = lower, ymax = upper), width = 0.1) +
  geom_point() +
  scale_x_continuous(limits = c(0,25), breaks = dat.df$rank,
     labels = dat.df$labels, name = "Region") +
  scale_y_continuous(limits = c(0,100),
     name = "Cases per 100 individuals at risk") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```

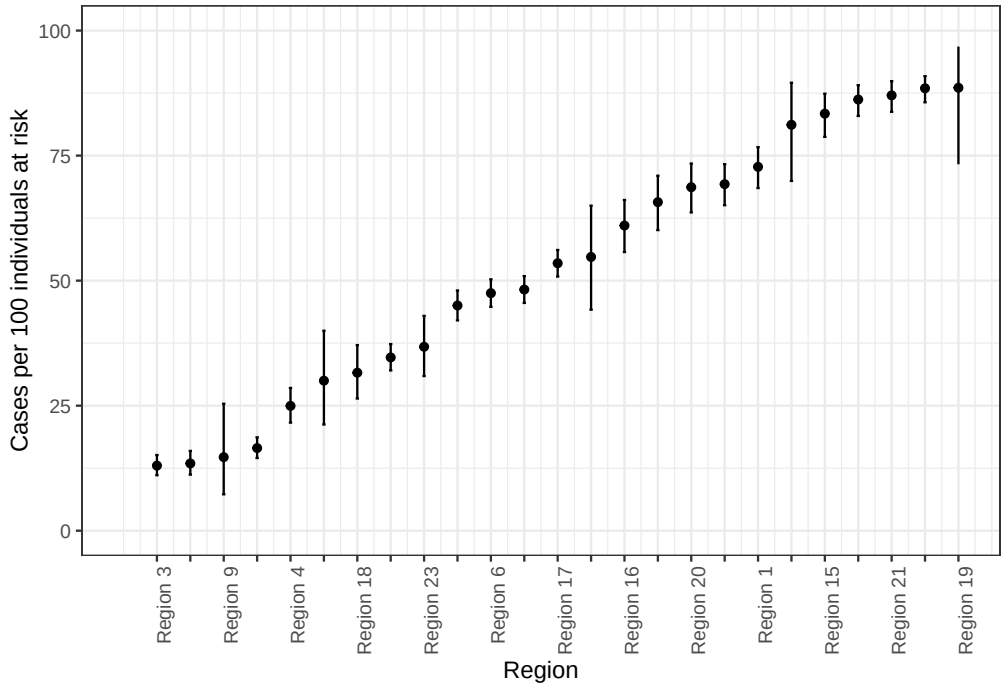


Figure 2: Ranked error bar plot showing the number of prevalent cases of disease (and its 95% CI) per 100 population units.

2 Time

Epidemic curves are used to describe patterns of disease over time. Epidemic curve data are often presented in one of two formats:

1. One row for each individual identified as a case with an event date assigned to each.
2. One row for every event date with an integer representing the number of cases identified on that date.

In the notes that follow we provide details on how to produce an epidemic curve when you're data are in these formats.

One row of data for each case

Generate some data, with one row for every individual identified as a case:

```r
n.males <- 100; n.females <- 50
odate <- seq(from = as.Date("2024-07-26"), to = as.Date("2024-12-13"), by = 1)
prob <- c(1:100, 41:1); prob <- prob / sum(prob)
modate <- sample(x = odate, size = n.males, replace = TRUE, p = prob)
fodate <- sample(x = odate, size = n.females, replace = TRUE)
```

```
dat.df <- data.frame(sex = c(rep("Male", n.males), rep("Female", n.females)),
   odate = c(modate, fodate))
dat.df$odate <- as.Date(dat.df$odate, format = "%Y-%m-%d")

# Sort the data in order of odate:
dat.df <- dat.df[sort.list(dat.df$odate),]
```

Plot the epidemic curve using the `ggplot2` and `scales` packages (Figure 3):

```
ggplot(data = dat.df, aes(x = odate)) +
  theme_bw() +
  geom_histogram(binwidth = 7, colour = "gray", fill = "dark blue",
     linewidth = 0.1) +
  scale_x_date(breaks = date_breaks("7 days"), labels = date_format("%d %b"),
     name = "Date") +
  scale_y_continuous(breaks = seq(from = 0, to = 30, by = 5), limits = c(0,30),
     name = "Number of cases") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```

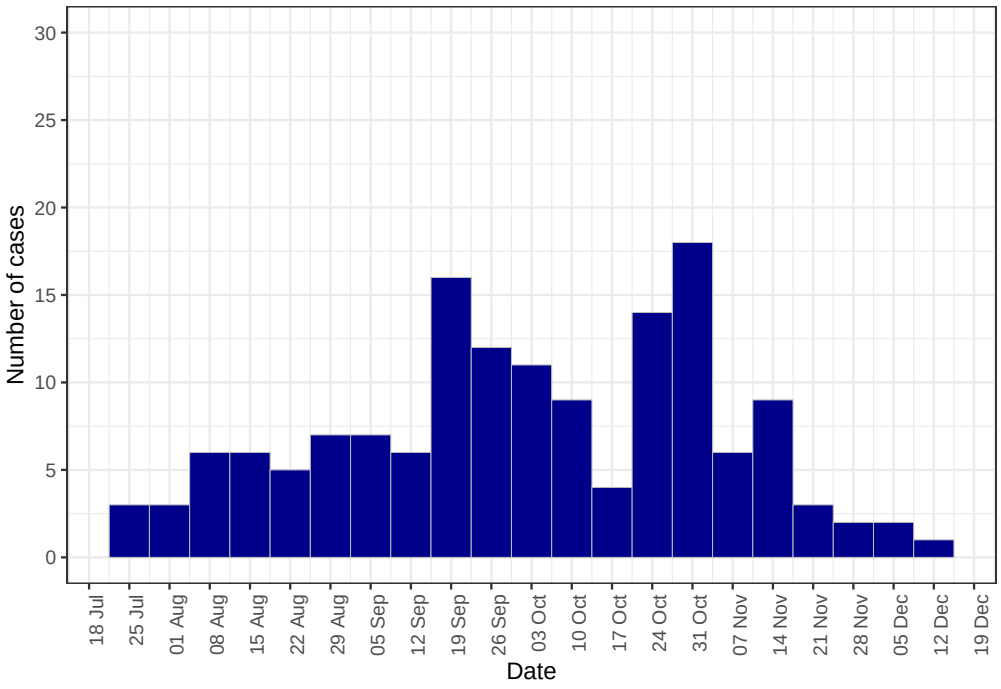


Figure 3: Frequency histogram showing counts of incident cases of disease as a function of calendar date, 26 July to 13 December 2024.

You may want to superimpose a smoothed line to better appreciate trend (Figure 4). Do this using the `geom_density` function in `ggplot2`:

```
ggplot(data = dat.df, aes(x = odate)) +
  theme_bw() +
  geom_histogram(binwidth = 7, colour = "gray", fill = "dark blue",
     linewidth = 0.1) +
  geom_density(aes(y = after_stat(density) * (nrow(dat.df) * 7)),
     colour = "red") +
  scale_x_date(breaks = date_breaks("7 days"), labels = date_format("%d %b"),
     name = "Date") +
  scale_y_continuous(breaks = seq(from = 0, to = 30, by = 5), limits = c(0,30),
```

```
    name = "Number of cases") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```

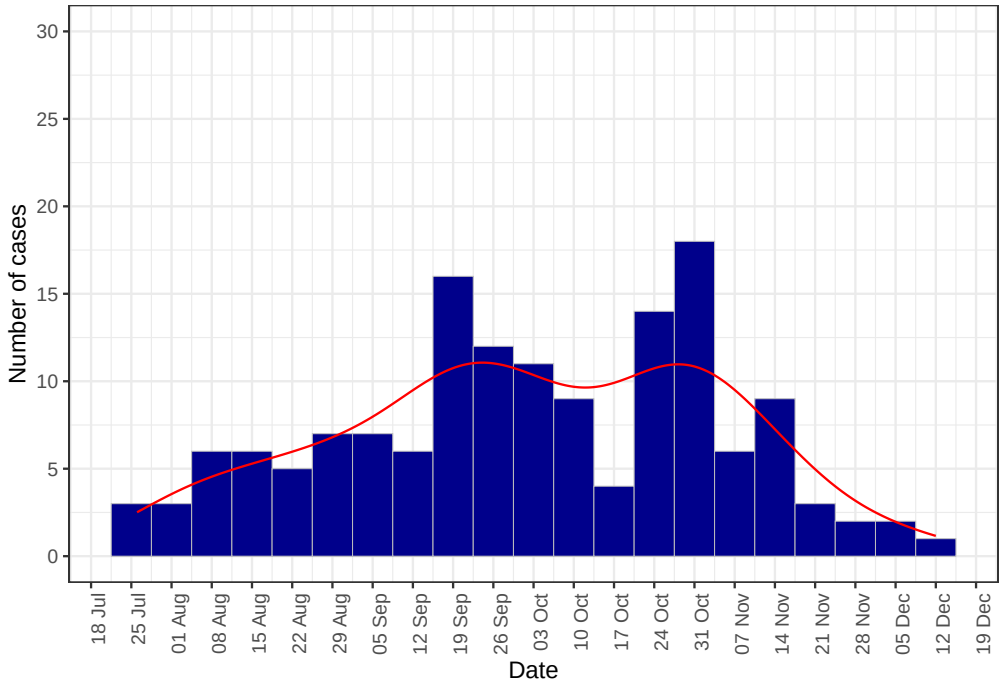


Figure 4: Frequency histogram showing counts of incident cases of disease as a function of calendar date, 26 July to 13 December 2024. Superimposed on this plot is a smoothed estimate of case density.

Condition the epidemic curve by sex using the `facet_grid` function (Figure 5):

```
ggplot(data = dat.df, aes(x = odate)) +
  theme_bw() +
  geom_histogram(binwidth = 7, colour = "gray", fill = "dark blue",
     linewidth = 0.1) +
  facet_grid( ~ sex) +
  scale_x_date(breaks = date_breaks("1 week"), labels = date_format("%d %b"),
     name = "Date") +
  scale_y_continuous(breaks = seq(from = 0, to = 30, by = 5), limits = c(0,30),
     name = "Number of cases") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```

Let's say an event occurred on 31 October 2024. Mark this date on the epidemic curve using `geom_vline` (Figure 6):

```
ggplot(data = dat.df, aes(x = odate)) +
  theme_bw() +
  geom_histogram(binwidth = 7, colour = "gray", fill = "dark blue",
     linewidth = 0.1) +
  facet_grid( ~ sex) +
  scale_x_date(breaks = date_breaks("1 week"), labels = date_format("%d %b"),
     name = "Date") +
  scale_y_continuous(breaks = seq(from = 0, to = 30, by = 5), limits = c(0,30),
     name = "Number of cases") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  geom_vline(aes(xintercept = as.Date("31/10/2024", format = "%d/%m/%Y")),
```

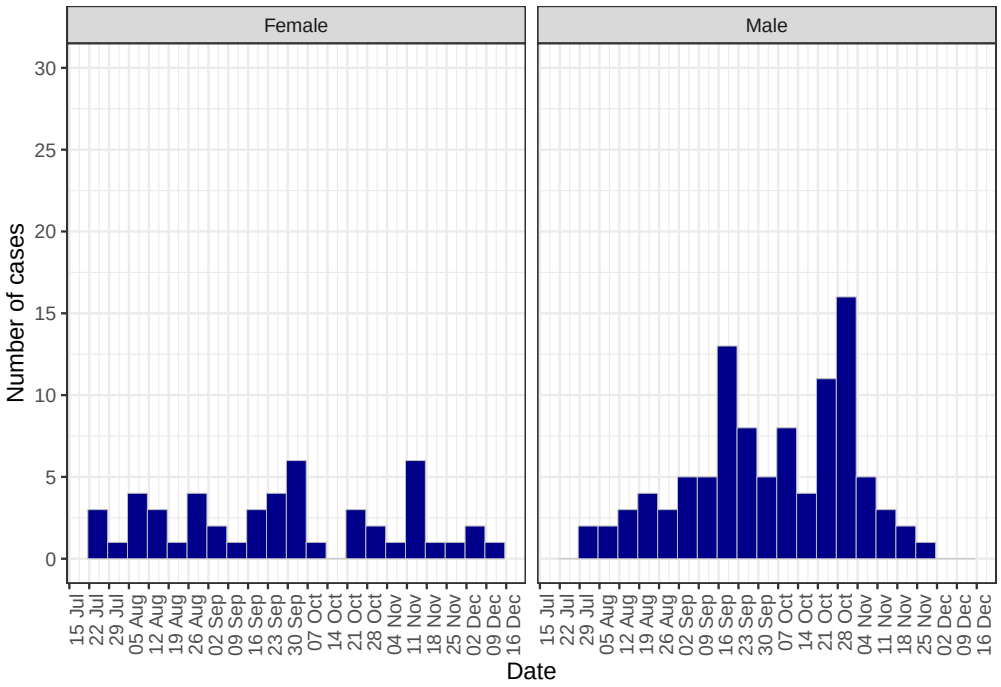


Figure 5: Frequency histogram showing counts of incident cases of disease as a function of calendar date, 26 July to 13 December 2024, conditioned by sex.

```
   linetype = "dashed")
```

Plot the total number of disease events by day, coloured by sex (Figure 7):

```
ggplot(data = dat.df, aes(x = odate, group = sex, fill = sex)) +
  theme_bw() +
  geom_histogram(binwidth = 7, colour = "gray", linewidth = 0.1) +
  scale_x_date(breaks = date_breaks("1 week"), labels = date_format("%d %b"),
     name = "Date") +
  scale_y_continuous(breaks = seq(from = 0, to = 30, by = 5), limits = c(0,30),
     name = "Number of cases") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  geom_vline(aes(xintercept = as.Date("31/10/2024", format = "%d/%m/%Y")),
   linetype = "dashed") +
  scale_fill_manual(values = c("#d46a6a", "#738ca6"), name = "Sex") +
  theme(legend.position = "bottom")
```

It can be difficult to appreciate differences in male and female disease counts as a function of date with the above plot format so dodge the data instead (Figure 8):

```
ggplot(data = dat.df, aes(x = odate, group = sex, fill = sex)) +
  theme_bw() +
  geom_histogram(binwidth = 7, colour = "gray", linewidth = 0.1,
     position = "dodge") +
  scale_x_date(breaks = date_breaks("1 week"), labels = date_format("%d %b"),
     name = "Date") +
  scale_y_continuous(breaks = seq(from = 0, to = 30, by = 5), limits = c(0,30),
     name = "Number of cases") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
```

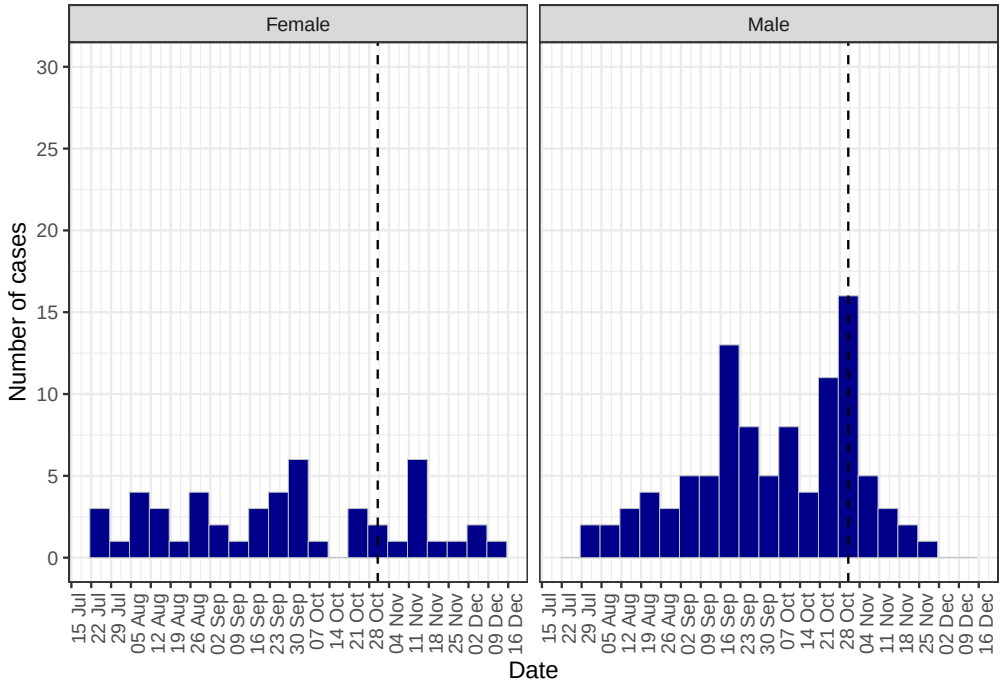


Figure 6: Frequency histogram showing counts of incident cases of disease as a function of calendar date, 26 July to 13 December 2024, conditioned by sex. An event that occurred on 31 October 2024 is indicated by the vertical dashed line.

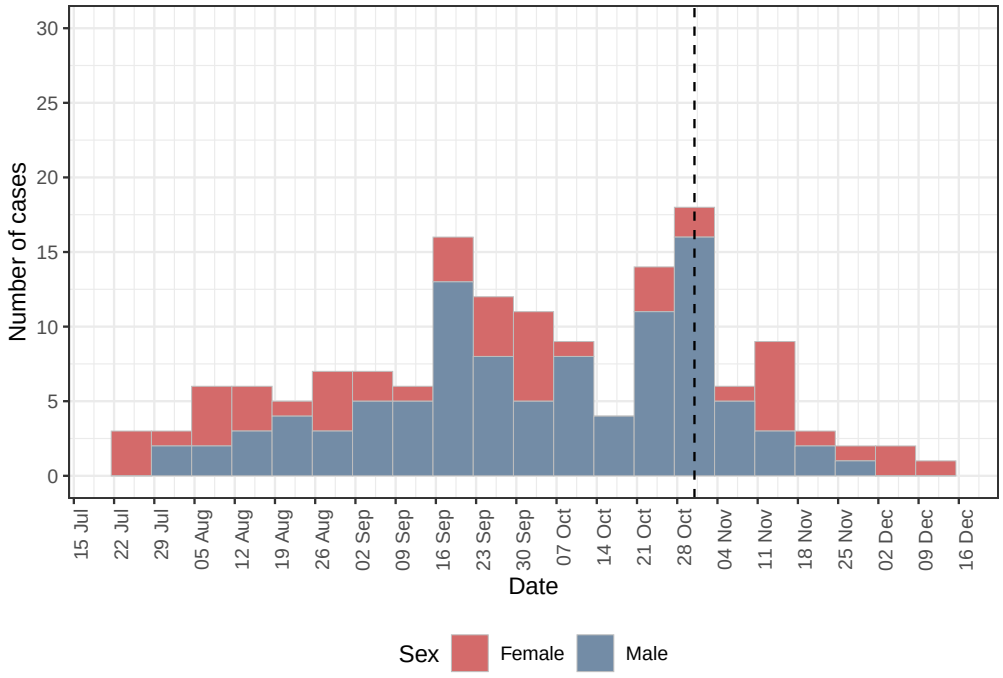


Figure 7: Frequency histogram showing counts of incident cases of disease as a function of calendar date, 26 July to 13 December 2024, grouped by sex.

```
  geom_vline(aes(xintercept = as.Date("31/10/2024", format = "%d/%m/%Y")),
   linetype = "dashed") +
  scale_fill_manual(values = c("#d46a6a", "#738ca6"), name = "Sex") +
  theme(legend.position = "bottom")
```

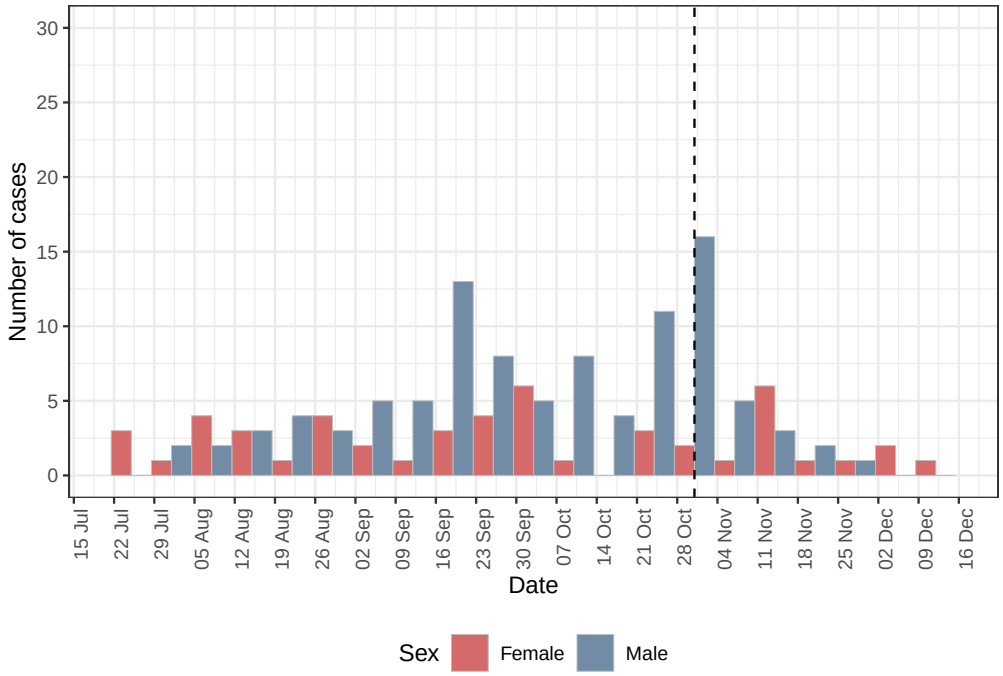


Figure 8: Frequency histogram showing counts of incident cases of disease as a function of calendar date, 26 July to 13 December 2024, grouped by sex.

Integers representing counts of cases on each date

We now provide code to deal with the situation where the data are presented with one row for every date during an outbreak and an integer representing the number of cases identified on each date.

Actual outbreak data will be used for this example. In the code below `edate` represents the event date (i.e., the date of case detection) and `ncas` represents the number of cases identified on each `edate`.

```
edate <- seq(from = as.Date("2024-02-24"), to = as.Date("2024-07-20"), by = 1)
ncas <- c(1,0,0,1,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,0,0,1,0,0,
   0,0,2,0,0,1,0,1,1,2,3,2,5,10,15,5,7,17,37,31,34,42,46,73,58,67,57,54,104,77,
   52,90,59,64,61,21,26,25,32,24,14,11,23,6,8,9,4,5,7,14,14,1,5,1,1,5,3,3,1,3,3,
   7,5,10,11,21,14,16,15,13,13,8,5,16,7,9,19,13,5,6,6,5,5,10,9,2,2,5,8,10,6,
   8,8,4,9,7,8,3,1,4,2,0,4,8,5,8,10,12,8,20,16,11,25,19)

dat.df <- data.frame(edate, ncas)
dat.df$edate <- as.Date(dat.df$edate, format = "%Y-%m-%d")
head(dat.df)
#>        edate ncas
#> 1 2024-02-24    1
#> 2 2024-02-25    0
#> 3 2024-02-26    0
#> 4 2024-02-27    1
#> 5 2024-02-28    0
#> 6 2024-02-29    1
```

Generate an epidemic curve (Figure 9). Note `weight = ncas` in the aesthetics argument for `ggplot2`:

```
ggplot() +
  theme_bw() +
  geom_histogram(dat.df, mapping = aes(x = edate, weight = ncas), binwidth = 1,
     fill = "#738ca6", colour = "grey", linewidth = 0.1) +
  scale_x_date(breaks = date_breaks("2 weeks"), labels = date_format("%b %Y"),
     name = "Date") +
  scale_y_continuous(limits = c(0,125), name = "Number of cases") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```

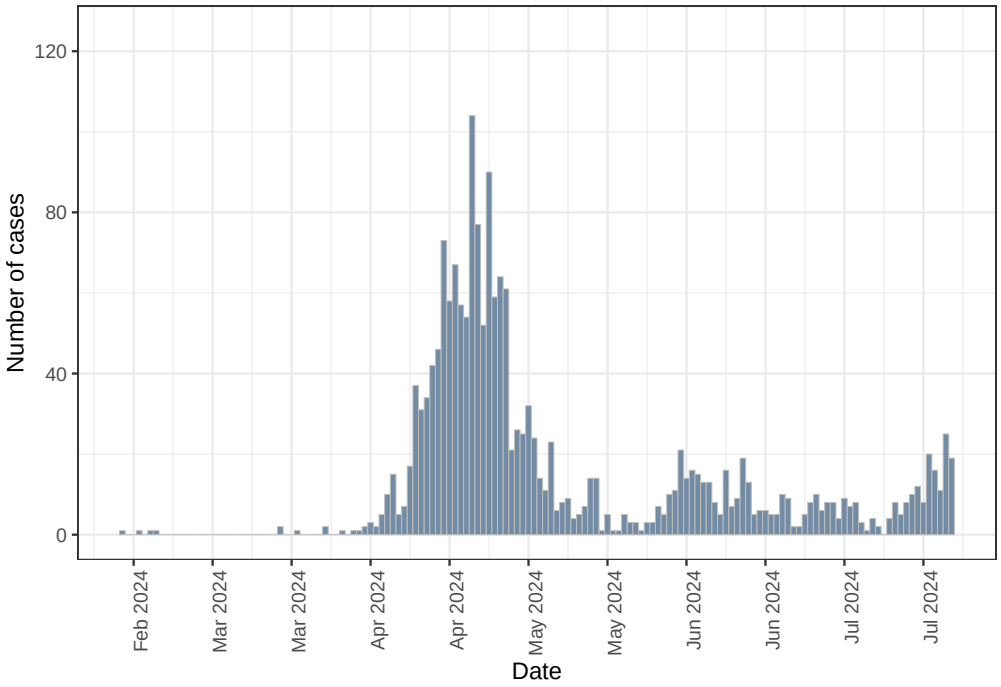


Figure 9: Frequency histogram showing counts of incident cases of disease as a function of calendar date, 24 February 2024 to 20 July 2024.

This plot has features of a common point source epidemic for the period April 2024 to May 2024. After May 2024 the plot shows feature of a propagated epidemic pattern.

Add a line to the plot to show the cumulative number of cases detected as a function of calendar date (Figure 10). The coding here requires some thought. First question: What was the cumulative number of cases at the end of the follow-up period? Use the `cumsum` (cumulative sum) function in base R:

```
max(cumsum(dat.df$ncas))
#> [1] 1834
```

At the end of the follow-up period the cumulative number of cases was 1834. What we need to do is to get our 0 to 1834 cumulative case numbers to 'fit' into the 0 to 125 vertical axis limits of the epidemic curve. A reasonable approach would be to: (1) divide cumulative case numbers by a number so that the maximum cumulative case number divided by our selected number roughly equals the maximum number of cases identified per day; for this example, 15 would be a good choice (1834 / 15 = 122); and (2) set `sec.axis = sec_axis(~ . * 15)` to multiply the values that appear on the primary vertical axis by 15 for the labels that appear on the secondary vertical axis:

```
ggplot() +
  theme_bw() +
  geom_histogram(data = dat.df, mapping = aes(x = edate, weight = ncas),
```

```
    binwidth = 1, fill = "#738ca6", colour = "grey", linewidth = 0.1) +
  geom_line(data = dat.df, mapping = aes(x = edate, y = cumsum(ncas) / 15)) +
  scale_x_date(breaks = date_breaks("2 weeks"), labels = date_format("%b %Y"),
    name = "Date") +
  scale_y_continuous(limits = c(0,125), name = "Number of cases",
     sec.axis = sec_axis(~ . * 15, name = "Cumulative number of cases")) +
  guides(fill = "none") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```

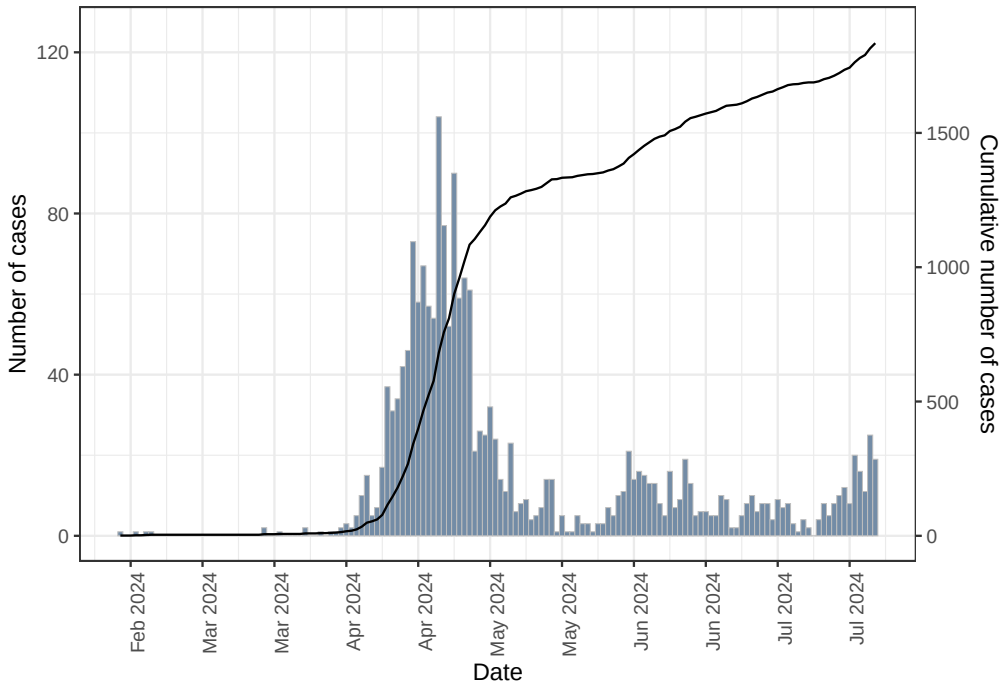


Figure 10: Frequency histogram showing counts of incident cases of disease as a function of calendar date, 24 February 2024 to 20 July 2024. Superimposed on this plot is a line showing cumulative case numbers.

Finally, we might want to superimpose a line representing the rolling average of case numbers (Figure 11). Calculate the 5-day rolling mean use the `rollmean` function in the contributed `zoo` package:

```
dat.df$rncas <- rollmean(x = dat.df$ncas, k = 5, fill = NA)

ggplot() +
  theme_bw() +
  geom_histogram(data = dat.df, mapping = aes(x = edate, weight = ncas),
    binwidth = 1, fill = "#738ca6", colour = "grey", linewidth = 0.1) +
  geom_line(data = dat.df, mapping = aes(x = edate, y = rncas),
    colour = "red") +
  scale_x_date(breaks = date_breaks("2 weeks"), labels = date_format("%b %Y"),
    name = "Date") +
  scale_y_continuous(limits = c(0,125), name = "Number of cases") +
  guides(fill = "none") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```

3 Place

Two types of maps are often used when describing patterns of disease by place:

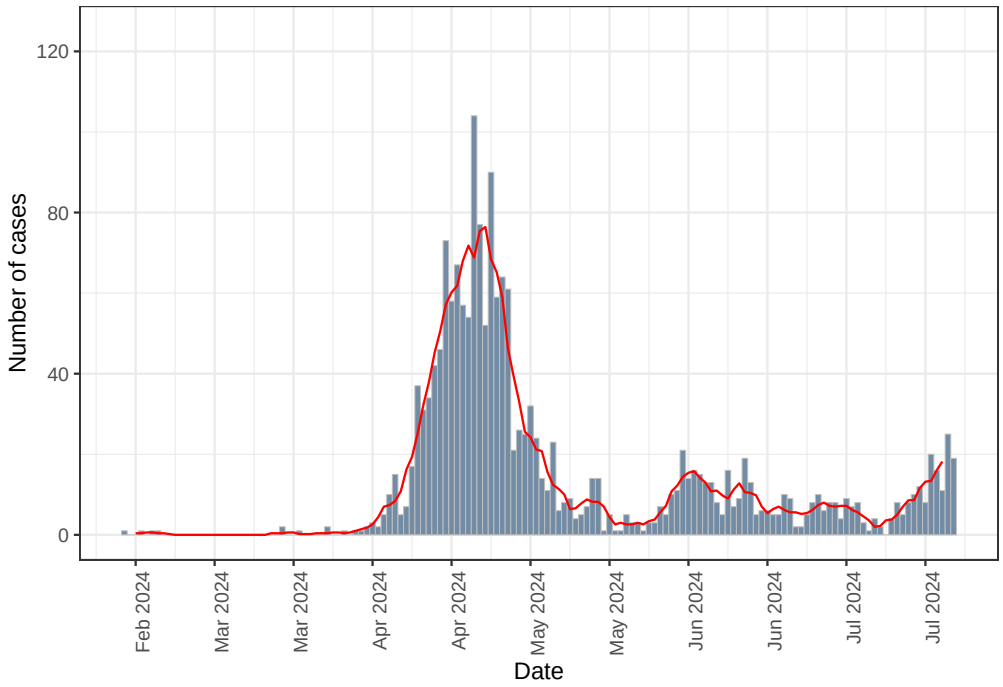


Figure 11: Frequency histogram showing counts of incident cases of disease as a function of calendar date, 24 February 2024 to 20 July 2024. Superimposed on this plot is the 5-day rolling mean number of cases per day.

1. Choropleth maps. Choropleth mapping involves producing a summary statistic of the outcome of interest (e.g., count of disease events, prevalence, incidence) for each component area within a study region. A map is created by 'filling' (i.e. colouring) each component area with colour, providing an indication of the magnitude of the variable of interest and how it varies geographically.
2. Point maps.

Choropleth maps

For illustration we make a choropleth map of the percentage of individuals aged greater than 65 years in an area of New York, USA. These data are taken from the data sets supporting Waller and Gotway (2004). In the code that follows `ny` refers to New York, `age65` refers to individuals aged greater than 65 years and `utm` refers to the spatial projection of the `sf` object (Universal Transverse Mercator). The object name suffix `.sf` reminds you that this is a simple features (spatial) object.

```
library(sf); library(spData); library(plyr); library(RColorBrewer);
library(sp); library(spatstat)

nyage65utm.sf <- st_read(dsn = system.file("shapes/NY8_bna_utm18.gpkg",
   package = "spData")[1], quiet = TRUE)
```

The `nyage65utm.sf` simple features object lists for each census tract the percentage of individuals aged greater than 65 years. Here we make a choropleth map and position the legend at the bottom of the plot and control the width of the legend using `legend.key.width` in the `theme` argument (Figure 12):

```
ggplot() +
   theme_bw() +
   geom_sf(data = nyage65utm.sf, aes(fill = PCTAGE65P), colour = "dark grey") +
   scale_fill_gradientn(limits = c(0,0.5), colours = brewer.pal(n = 5, "Reds"),
      guide = "colourbar") +
   scale_x_continuous(name = "Longitude") +
```

```
   scale_y_continuous(name = "Latitude") +
   labs(fill = "Age >65 years") +
   theme(legend.position = "bottom", legend.key.width = unit(1.0, "cm"))
```

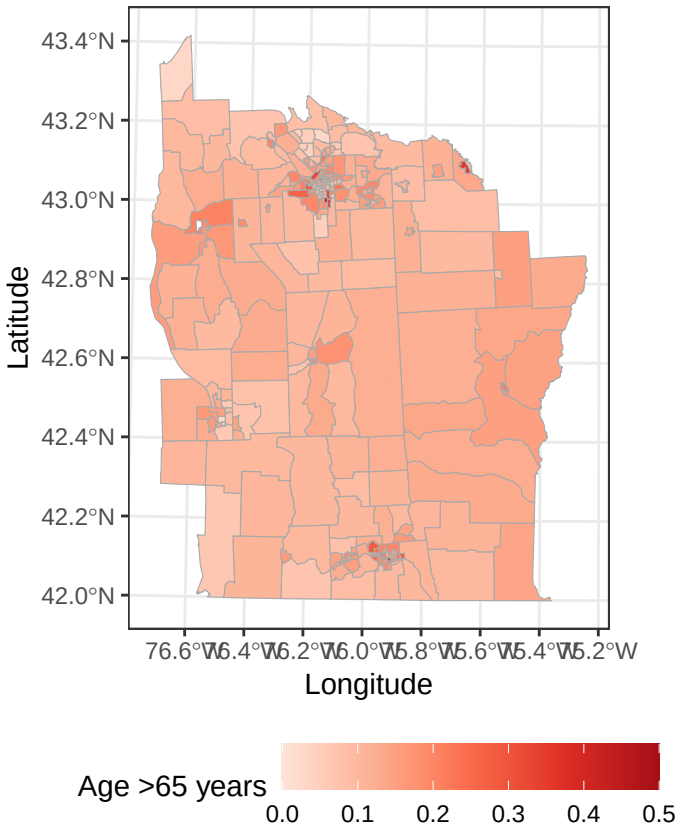


Figure 12: Map of an area of New York, USA showing for each census tract the percentage of individuals aged greater than 65 years.

Point maps

Between 1972 and 1980 an industrial waste incinerator operated at a site about 2 kilometres southwest of the town of Coppull in Lancashire, England. Addressing community concerns that there were greater than expected numbers of laryngeal cancer cases in close proximity to the incinerator Diggle (1990) conducted a study investigating risks for laryngeal cancer, using recorded cases of lung cancer as controls. The study area is 20 km $\times$ 20 km in size and includes location of residence of patients diagnosed with each cancer type from 1974 to 1983.

Load the `chorley` data set from the `spatstat` package. The point locations in this data are projected using the British National Grid coordinate reference system (EPSG code 27700). Create an observation window for the data as `coppull.ow` and a `ppp` object for plotting:

```
data(chorley)
chorley.df <- data.frame(xcoord = chorley$x * 1000, ycoord = chorley$y * 1000,
   status = chorley$marks)
chorley.df$status <- factor(chorley.df$status, levels = c("lung","larynx"),
   labels = c("Lung","Larynx"))

chlarynxbng.sf <- st_as_sf(chorley.df, coords = c("xcoord","ycoord"),
   remove = FALSE)
st_crs(chlarynxbng.sf) <- 27700
```

```
chlarynxbng.ow <- chorley$window
```

Create a simple features polygon object from `coppull.ow`. First we convert `chlarynxbng.ow` to a `SpatialPolygonsDataFrame` object:

```
coords <- matrix(c(chlarynxbng.ow$bdry[[1]]$x * 1000,
   chlarynxbng.ow$bdry[[1]]$y * 1000), ncol = 2, byrow = FALSE)
pol <- Polygon(coords, hole = FALSE)
pol <- Polygons(list(pol),1)
pol <- SpatialPolygons(list(pol))
chpolbng.spdf <- SpatialPolygonsDataFrame(Sr = pol, data = data.frame(id = 1),
   match.ID = TRUE)
```

Convert the `SpatialPolygonsDataFrame` to an `sf` object and set the coordinate reference system to the British National Grid, EPSG 27700:

```
chpolbng.sf <- as(chpolbng.spdf, "sf")
st_crs(chpolbng.sf) <- 27700
```

The `mformat` function is used to plot the axis labels in kilometres (instead of metres):

```
mformat <- function(){
   function(x) format(x / 1000, digits = 2)
}
```

And make the plot (Figure 13):

```
ggplot() +
  theme_bw() +
  geom_sf(data = chlarynxbng.sf, aes(colour = status, shape = status)) +
  geom_sf(data = chpolbng.sf, fill = "transparent", colour = "black") +
  coord_sf(datum = st_crs(chpolbng.sf)) +
  scale_colour_manual(name = "Type", values = c("grey","red")) +
  scale_shape_manual(name = "Type", values = c(1,16)) +
  scale_x_continuous(name = "Easting (km)", labels = mformat()) +
  scale_y_continuous(name = "Northing (km)", labels = mformat()) +
  theme(legend.position = "bottom")
```

References

Centers for Disease Control and Prevention. 2006. *Principles of Epidemiology in Public Health Practice: An Introduction to Applied Epidemiology and Biostatistics*. Atlanta, Georgia: Centers for Disease Control; Prevention.

Diggle, PJ. 1990. "A point process modeling approach to raised incidence of a rare phenomenon in the vicinity of a prespecified point." *Journal of the Royal Statistical Society Series A* 153: 349–62.

Feychting, M, B Osterlund, and A Ahlbom. 1998. "Reduced cancer incidence among the blind." *Epidemiology* 9: 490–94.

Porta, M, S Greenland, and JM Last. 2014. *A Dictionary of Epidemiology*. London: Oxford University Press.

Waller, L, and C Gotway. 2004. *Applied Spatial Statistics for Public Health Data*. New York, USA: John Wiley and Sons Ltd.

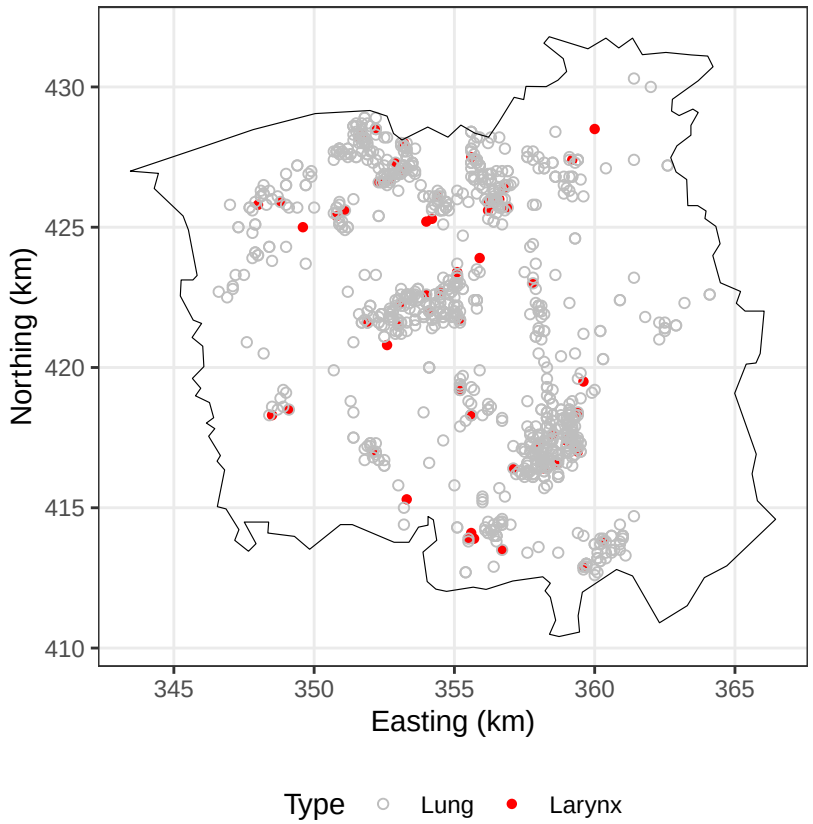


Figure 13: Point map showing the place of residence of individuals diagnosed with laryngeal cancer and lung cancer, Copull Lancashire, UK, 1972 to 1980.