Hexamaps for Visualizing Age-Period-Cohort Data Trends

Authors: Hawre Jalal^{1,*} and Donald S. Burke²

Affiliations

1. Department of Health Policy and Management, Graduate School of Public Health, University of Pittsburgh.
2. Department of Epidemiology, Graduate School of Public Health, University of Pittsburgh.

***Correspondence to:** hjalal@pitt.edu

Keywords: graphical, age-period-cohort, visualization, analysis

Word Count: 1322

Abstract:

Age-period-cohort (APC) analyses often reveal important insights into patterns of disease incidence and mortality such as cancer. Both visual and analytical APC tools are available to reveal patterns by age, period and cohort. While developing new analytical methods of APC data is an active area of research, the choices of visual tools specific to APC data has been limited. In this study, we propose a “hexamap” as a new APC specific data visualization tool. This tool uses hexagons to address the unique challenge of APC data visualization, that is cohort = period - age. The approach is further illustrated using alcohol related mortality. Flexible Open-Source functions for implementation in R is also provided.

Background

Age-period-cohort (APC) analyses is currently the mainstay to disentangle the effect of age, period and cohort on disease outcomes, such as cancer incidence and death. Because cohort = period – age, typical APC analysis involves an identification issue. Significant progress in addressing this issue has been achieved with many APC-specific analytical tools, such as ¹⁻⁶ However, there is a lack of APC-specific data visualization tools. Most of APC data visualization rely on simple line charts or heatmaps on a regular grid that can only clearly present data on two of the APC axes.

In this paper, we propose hexagonal heatmaps as APC-specific data visualization tool that can overcome some of the challenges in current data display. We illustrate the hexagonal heatmaps with two examples. We also provide the R-code implementation of the tool to facilitate wider adoption and implementation.

Hexagonal grids

Square grids are often used in Lexis diagrams to display APC data, where the data is arranged in a rectangular field and the rates or counts in each square is shown using different colors or shades (Figure 1A). [For simplicity, only an empty grid is shown here]. In this setup, age is often shown perpendicular to period while cohorts are being placed on the diagonal between age and period. Because of this uneven alignment, this display is less effective in revealing cohort patterns relative to age or period because (1) the square pixels only share a corner making it visually challenging to see the patterns along the diagonals relative to the pixels along the age and period axes that share a full side, and (2) this display creates a visual distortion where the spacing among cohorts seem to be wider than age or period. For example, if the data is arranged at one-year intervals, successive cohorts seem to be separated at $\sqrt{2} = 1.4$ years, causing cohort patterns to appear diluted relative to those for age or period.

Figure 1. Age-period-cohort isoline orientation in a square vs. hexagonal data grid.

We propose a hexagonal grid (Figure 1B) to overcome the limitations of the traditional square grid. Hexagons are the most rounded shape that can tessellate (i.e., be repeated edge-to-edge without gaps to create an evenly spaced grid)⁷. Each cell in a hexagonal grid can have six neighbors. Thus, for each cell, one can easily navigate to how the patterns are changing with increasing or decreasing age, period or cohort. Specifically, because the hexagonal grid places the APC axes at equal angles from each other, the hexagonal pixels along any of the three dimensions share a side, facilitating visual pattern recognition along these axes. In addition, successive ages, periods and cohorts are equally spaced, thereby facilitating the visual display and interpretation of APC data along these axes.

Plotting APC hexagons on the XY coordinate system:

Using a hexagonal grid on a two dimensional medium or display requires computing the coordinates for the center of each hexagon on the Cartesian XY coordinate system. Here we follow the age, period and cohort isoline placements as shown in Figure 1 (B). We use these equations to compute the center for each hexagon based on age and period setting the distance between any two neighbouring hexagonal pixel centers to equal to one unit of time (e.g., one year):

$$x = \frac{p\sqrt{3}}{2} \quad (1)$$

$$y = a - \frac{p}{2} \quad (2)$$

where x and y are the x and y coordinates on the Cartesian coordinate system, and a = age, and p = period. Because of our setup with period as vertical isolines, the x -coordinate of the hexagonal pixel is only a function of p , while the y -coordinate is a function of both p and a as they both determine the vertical position of the hexagonal pixels. Then, each hexagon can be plotted around its computed center using these XY coordinates for the hexagonal vertices (corners):

$$\left\{ \left(\frac{-1}{\sqrt{3}}, 0 \right), \left(\frac{-1}{2\sqrt{3}}, \frac{1}{2} \right), \left(\frac{1}{2\sqrt{3}}, \frac{1}{2} \right), \left(\frac{1}{\sqrt{3}}, 0 \right), \left(\frac{1}{2\sqrt{3}}, -\frac{1}{2} \right), \left(\frac{-1}{2\sqrt{3}}, -\frac{1}{2} \right) \right\}$$

Case study – alcohol related mortality in the United States from 1999 through 2017:

To illustrate the Hexamap, we use alcohol-related mortality rates in the US from 1999 through 2017 as an example. The data was obtained from CDC-Wonder.⁸ Alcohol related poisoning is identified as alcohol overdose poisonings (International Classification of Disease 10th Revision (ICD-10) codes of X45, X65 or Y15. The starting period is 1999, starting age is 0, and the interval is 1 year.

Figure 2 shows the hexamap as produced in R for alcohol-related mortality rates. Each data point at the intersection of age, period and cohort is represented as a hexagon. The APC isolines are placed at 5-year intervals. Alcohol related deaths are increasing among most age groups,

especially after 2008. In 2017, the highest mortality rates were between ages 53-63. In addition to the increase of deaths by period, mortality rates seem especially higher among cohorts born after 1947.

Figure 2: A hexamap showing the birth-year cohort patterns and the recent increase in alcohol-related mortality rates in the US from 1999 through 2017.

Discussion:

In this paper we propose a hexamap as a new way to visualize APC data. We illustrate how a hexagonal grid can overcome challenges of traditional visualization of APC data. We illustrate that using a hexagonal grid, one can compare patterns by changing age, period and cohort. We provide the functionality using R code in the supplement.

Researchers have often presented APC data using generic non-APC specific visualizations ranging from simple line charts to 3d plots ⁹. Line charts often show the incidence or mortality rates by age or period on the x-axis with individual lines presenting various cohorts. These line charts are challenging to interpret as they can only provide data readily on two of the three dimensions without a clear link between the remaining dimension. In addition, 3-d plots have been proposed where two of the dimensions are often age and period, and the third dimension is the incidence rate or mortality. The display of APC data in these visualizations is not necessarily clear given that one still needs to rotate the 3-d plot to 45 degrees to visualize the cohort patterns.

Hexagons are especially attractive for data visualization, because they are the most rounded object that creates a continuous grid. Hexagons has been recommended for visualizing data in many fields, such as ecological modeling and spatial analysis ⁷, showing bivariate statistical density distribution ¹⁰, and traffic pattern studies ¹¹. In these fields, hexagons are often preferred because each data point can be compared readily to six neighboring data points compared to the four data points in traditional square grid presentation. In this paper, we illustrate how a hexagonal grid can be especially useful to visualize APC data because the data structure is naturally suited for a hexagonal.

It is important to note that our hexagonal grid places the APC axes at 60 degree angles. Thus, successive age, period and cohort isolines produce an equilateral triangular mesh. This equilateral mesh is equivalent to the Lexis's equilateral diagram recommended in Lexis in 1875 ¹² and then by Weinkam and Sterling in 1991 ¹³ as a superior presentation of APC data to the standard square grid that produces right-angled triangular mesh. In addition, Weinkam and Sterling propose to overlay level curves that represent continuous regions with similar rate over the Lexis' equilateral triangular mesh. However, this display is complicated as it involves multiple intersecting isolines of the various levels with the isolines of the APC axes. However, the use of the triangular mesh and the level-curves have been limited because of their complexity. Unlike the level-curves, the hexagons do not interfere with the isolines. Thus, we hope that researchers will find the hexamaps and the associated R code as a more practical and intuitive way to reveal patterns across the APC dimensions.

References:

1. Lee WC, Lin RS. Modelling the Age-period-cohort Trend Surface. *Biometrical J.* 1996;38(1):97-106. doi:10.1002/bimj.4710380109
2. Rosenberg PS, Check DP, Anderson WF. A Web Tool for Age – Period – Cohort Analysis of Cancer Incidence and Mortality Rates. 2014;23(November):2296-2303. doi:10.1158/1055-9965.EPI-14-0300
3. Chernyavskiy P, Little MP, Rosenberg PS. A unified approach for assessing heterogeneity in age–period–cohort model parameters using random effects. *Stat Methods Med Res.* 2019;28(1):20-34. doi:10.1177/0962280217713033
4. Rosenberg PS. A new age-period-cohort model for cancer surveillance research. *Stat Methods Med Res.* 2018;28:3363-3391. doi:10.1177/0962280218801121
5. Robertson C, Boyle P. Age-period-cohort analysis of chronic disease rates I: Modelling approach. *Stat Med.* 1998;17(12):1305-1323. doi:10.1002/(SICI)1097-0258(19980630)17:12<1305::AID-SIM853>3.0.CO;2-W
6. Yang Y. Social inequalities in happiness in the United States, 1972 to 2004: An age-period-cohort analysis. *Am Sociol Rev.* 2008;73(2):204-226. doi:10.1177/000312240807300202
7. Birch CPD, Oom SP, Beecham JA. Rectangular and hexagonal grids used for observation, experiment and simulation in ecology. *Ecol Modell.* 2007;206(3-4):347-359. doi:10.1016/j.ecolmodel.2007.03.041
8. Centers for Disease Control and Prevention National Center for Health Statistics. Multiple Cause of Death 1999-2017 on CDC WONDER Online Database, released 2018. *Data are from Mult cause death files.* 2018;2018.
9. Robertson C, Boyle P. Age-period-cohort models of chronic disease rates. II: Graphical approaches. *Stat Med.* 1998;17(12):1325-1339. doi:10.1002/(SICI)1097-0258(19980630)17:12<1325::AID-SIM854>3.0.CO;2-R
10. Scott DW. A note on choice of bivariate histogram bin shapes. *J Off Stat.* 1988;4(1):47-51.
11. Maniccam S. Traffic jamming on hexagonal lattice. *Phys A Stat Mech its Appl.* 2003;321(3-4):653-664. doi:10.1016/S0378-4371(02)01549-2
12. Lexis W. *Einleitung in Die Theorie Der Bevölkerungsstatistik.*; 1875.
13. Weinkam JJ, Sterling TD. A graphical approach to the interpretation of age-period-cohort data. *Epidemiology.* 1991;2(2):133-137. <http://dx.doi.org/>.
14. Rosenberg PS, Check DP, Anderson WF. A web tool for age-period-cohort analysis of cancer incidence and mortality rates. *Cancer Epidemiol Biomarkers Prev.* 2014;23(11):2296-2302. doi:10.1158/1055-9965.EPI-14-0300

Supplementary material

R code

In this section we provide the R functions that can create hexamaps for any matrix of counts or rates by age and period. In summary, the hexamaps can be produced by running the `create_hexamap()` function which requires four parameters:

```
create_hexamap(data, first_age, first_period, interval)
```

where `data` is a matrix containing the values to be plotted. The function expects the values to be arranged by age as rows and periods as columns. This pattern is consistent with other tools, such as the National Cancer Institute (NCI)'s online APC tool¹⁴ and the Surveillance, Epidemiology, and End Results (SEER) program that generally produces outputs in this structure.

Similarly, `first_age` is the first age group in the data, `first_period` is the first period in the data, and `interval` defines the interval that the data is spaced. The code requires the data to be equally spaced for both period and age. The `create_hexamap()` then creates the hexagons, colors them based on their relative values and then overlays the age, period and cohort isolines. Optionally, the user can modify how the starting values for the isolines, and how the isolines are spaced. In addition, the user can also modify some of the aesthetics, such as the color scales used to show the intensity of the values, the color and width of the lines, and the color and size of the fonts for the isolines. Other properties can also be modified by varying the underlying code which only requires base R.

For example, the code below reads the data and generates the hexamap for alcohol mortality rates in the US from 1979 through 2017 shown in Figure 2:

```
# =====
# Hexamap for Alcohol-related deaths 1979-2017

data.df <- read.csv("Alcohol_Mortality.csv")
data <- as.matrix(data.df) # convert to a matrix and exclude the row names

create_hexamap(data = data, #matrix: age as rows, period as columns
               first_age = 0,
               first_period = 1999,
               interval = 1,
               first_age_isoline = 0,
               first_period_isoline = 2000,
               isoline_interval = 5,
               color_scale = c(0,30))
```

The code below shows the main hexamap function which is dependent on two functions to transform the X and Y coordinates to the hexagonal APC grid:

```
create_hexamap <- function(data, #matrix: age as rows, period as columns
                          first_age,
                          first_period,
                          interval,
                          first_age_isoline = NULL,
                          first_period_isoline = NULL,
                          isoline_interval = NULL,
                          color_scale = NULL,
                          color_map = NULL,
                          line_width = .5,
                          line_color = "grey",
                          label_size = .5,
                          label_color = "black",
                          scale_units = "Rate",
                          wrap_cohort_labels = TRUE) {
```



```
# setting default values for missing parameters
if(is.null(first_age_isoline)){
  first_age_isoline = first_age
}
if(is.null(first_period_isoline)){
  first_period_isoline = first_period
}
if(is.null(isoline_interval)){
  isoline_interval = 2 * interval
}
if(is.null(color_scale)){ #if color scale is missing use the min and max of data
  color_scale[1] <- min(data)
  color_scale[2] <- max(data)
}
if(is.null(color_map)){
  # define jet colormap
  jet.colors <- colorRampPalette(c("black", "#00007F", "blue", "#007FFF", "cyan", "#7FFF7F", "yellow", "#FF7F00", "red",
"#7F0000"))
  color_map = jet.colors(100)
}
# end of default values

m <- dim(data)[1]
n <- dim(data)[2]

last_age = first_age + (m - 1) * interval
last_period = first_period + (n - 1) * interval
first_cohort = first_period - last_age
last_cohort = last_period - first_age

age_isolines = seq(from = first_age_isoline, to = last_age, by = isoline_interval)
period_isolines = seq(from = first_period_isoline, to = last_period, by = isoline_interval)
last_age_isoline = tail(age_isolines,1)
first_cohort_isoline = first_period_isoline - last_age_isoline
cohort_isolines = seq(from = first_cohort_isoline, to = last_cohort, by = isoline_interval)

periods <- seq(from = first_period, to = last_period, by = interval)
ages <- seq(from = first_age, to = last_age, by = interval)
cohorts <- seq(from = first_cohort, to = last_cohort, by = interval)
n_ages <- length(ages)
n_periods <- length(periods)
n_cohorts <- length(cohorts)

n_age_isolines <- length(age_isolines)
n_period_isolines <- length(period_isolines)
n_cohort_isolines <- length(cohort_isolines)

# apply the limits to the data by truncating it
data[data<color_scale[1]] = color_scale[1]
data[data>color_scale[2]] = color_scale[2]

# === plotting
ncol <- length(color_map)
not_nan_data <- !is.nan(data)

v_data <- as.vector(data[not_nan_data])
datac = cut(data[not_nan_data], #discretize the data
  seq(from = color_scale[1], to = color_scale[2], length.out = ncol),
  include.lowest = T,
  labels = F)

a <- interval / sqrt(3) # radius of the hexagon (distance from center to a vertex).
b <- sqrt(3)/2 * a # half height of the hexagon (distance from the center perpendicular to the middle of the top edge)
yv <- c(0, b, b, 0, -b, -b, 0)
xv <- c(-a, -a/2, a/2, a, a/2, -a/2, -a)

# compute the center of each hexagon by creating an a*p grid for each age-period combination
P0 <- matrix(periods, nrow = n_ages, ncol=n_periods, byrow = TRUE)
A0 <- t(matrix(ages, nrow = n_periods, ncol = n_ages, byrow = TRUE))

# convert the grid to the X-Y coordinate
X <- compute_xcoordinate(P0)
Y <- compute_ycoordinate(P0, A0)

# only keep those that have non-NA values
X <- X[not_nan_data]
Y <- Y[not_nan_data]

# get the color for each level
color_map2 <- color_map[datac]

Xvec <- as.vector(X)
Yvec <- as.vector(Y)
n_hexagons <- length(Xvec)

# compute the X and Y coordinate for each hexagon - each hexagon is a row and each point is a column
Xhex <- outer(Xvec, xv, '+')
Yhex <- outer(Yvec, yv, '+')

minX <- min(Xhex) - interval
maxX <- max(Xhex) + interval
if (wrap_cohort_labels){
  minY <- min(Yhex) - interval
```

```

} else {
  minY <- compute_ycoordinate(p=first_period, a=first_age - (last_period-first_period)) - interval
}
maxY <- max(Yhex) + interval

layout(t(1:2),widths=c(4,1)) # two columns - one for the plot, the other for the colorbar

par(mar=c(.5,.5,.5,.5))

plot(x = NULL, y = NULL,
      xlim = c(minX,maxX),
      ylim = c(minY,maxY), axes=FALSE, frame.plot=FALSE, xaxt = 'n', yaxt = 'n', type = 'n', asp = 1)
for (i in 1:n_hexagons){
  polygon(x = Xhex[i,], # X-Coordinates of polygon
          y = Yhex[i,], # Y-Coordinates of polygon
          col = color_map2[i], # Color of polygon
          border = NA, # Color of polygon border
          lwd = 1)
}

#age-isolines
y1 <- compute_ycoordinate(first_period,age_isolines)
y2 <- compute_ycoordinate(last_period+ interval,age_isolines)
x1 <- compute_xcoordinate(first_period)
x2 <- compute_xcoordinate(last_period + interval)
for (i in 1:n_age_isolines){
  lines(x=c(x1,x2), y=c(y1[i],y2[i]), col = line_color, lwd = line_width)
  text(x=x2, y=y2[i], labels = paste("A:",age_isolines[i]),
        col = label_color, cex = label_size, srt = -30,
        adj = c(0, 0.5))
}

# period-isolines
x <- compute_xcoordinate(period_isolines)
y1 <- compute_ycoordinate(period_isolines, first_age)
y2 <- compute_ycoordinate(period_isolines, last_age+interval)

for (i in 1:n_period_isolines){
  lines(x=c(x[i], x[i]), y=c(y1[i],y2[i]), col = line_color, lwd = line_width)
  text(x=x[i], y=y2[i], labels = paste("P:",period_isolines[i]),
        col = label_color, cex = label_size, srt = 90,
        adj = c(0, .5)) #pos = 4)
}

# cohort-isolines (need some more processing!)
# determine the periods where the cohort isolines cross the last age
p_top <- cohort_isolines + last_age
p_top <- p_top[p_top < last_period]
n_top <- length(p_top)
# and the periods where they cross the first age
p_bottom <- cohort_isolines + first_age
p_bottom <- p_bottom[p_bottom > first_period]
n_bottom <- length(p_bottom)
# and the ages where they cross the first period
a_left <- first_period - cohort_isolines
if (wrap_cohort_labels){
  a_left <- a_left[a_left >= first_age]
}
n_left <- length(a_left)
# and the ages where they cross the last period
a_right <- last_period - cohort_isolines
a_right <- a_right[a_right <= last_age]
n_right <- length(a_right)

# combine the periods and ages initial and final points on the a*p coordinates
# first the left-bottom edge
if (wrap_cohort_labels){
  p1 <- c(rep(first_period, n_left), p_bottom)
  a1 <- c(a_left, rep(first_age, n_bottom))
} else {
  p1 <- c(rep(first_period, n_left))
  a1 <- c(a_left)
}
# then the top-right edge
p2 <- c(p_top, rep(last_period, n_right))
a2 <- c(rep(last_age, n_top), a_right)

# convert the a*p coordinates to x-y coordinates
x1 <- compute_xcoordinate(p1-interval) #,a1-1)
x2 <- compute_xcoordinate(p2) #,a2)
y1 <- compute_ycoordinate(p1-interval, a1-interval)
y2 <- compute_ycoordinate(p2, a2)
# finally draw the lines.
for (i in 1:n_cohort_isolines){
  lines(x=c(x1[i], x2[i]), y=c(y1[i],y2[i]), col = line_color, lwd = line_width)
  text(x=x1[i], y=y1[i], labels = paste("C:",cohort_isolines[i]),
        col = label_color, cex = label_size, srt = 30,
        adj = c(1,.5))
}

# create the colorbar
par(las=2)
par(mar=c(10,2,10,2.5))
cb_range <- seq(from = color_scale[1], to = color_scale[2], length.out = ncol)
image(y=cb_range,z=t(cb_range), col=color_map, axes=FALSE, main=scale_units, cex.main=.8)
axis(4,cex.axis=label_size,mgp=c(0,.5,0))

```

```
}  
  
compute_xcoordinate <- function(p) {  
  x <- p * sqrt(3) / 2  
  return(x)  
}  
  
compute_ycoordinate <- function(p, a){  
  y <- a - p / 2  
  return(y)  
}
```