Stat 5810, Section 005 &
Stat 6910, Section 003

Statistical Visualization II
Spring 2019

Dr. Jürgen Symanzik

Utah State University

Department of Mathematics and Statistics

3900 Old Main Hill

Logan, UT 84322–3900

Tel.: (435) 797–0696

FAX: (435) 797–1822

e-mail: `symanzik@math.usu.edu`

Web: `http://www.math.usu.edu/~symanzik/`

Scans//PhdcomicsCom_Plotting.jpg

Figure 1: http://www.phdcomics.com/comics/archive.php?comicid=1541, Cartoon.

Contents

Acknowledgements

This course uses some of the course materials provided by Dr. Mike Minnotte (formerly USU, now with the University of North Dakota) as held in the Fall 2006 semester. Additional materials have been taken from other Statistical Graphics courses, such as the ones offered by Dr. Di Cook (formerly Iowa State University; now Monash University: `http://dicook.org/`) and Dr. Dan Carr (George Mason University: `http://mason.gmu.edu/~dcarr/`). Other examples and R code originate from David Kahle, Heike Hofmann, Carson Sievert, Martin Theus, Antony Unwin, Simon Urbanek, Hadley Wickham, Lee Wilkinson, and others. We are likely to include parts from additional authors and sources that will be specified later during the semester.

Thanks are also due to 80+ students and guests who took the former "Stat 6560: Graphical Methods" and the current "Statistical Visualization I & II" courses with me since the Spring 2009 semester for their valuable comments that helped to improve, correct, and extend these lecture notes.

Jürgen Symanzik, January 21, 2019.

`Scans//Zelazny_px_Fig.jpg`

Figure 2: **?**, p. x, Cartoon.

1 Bivariate Plots

1.1 Scatterplots

?, p. 46, state:

> "Scatterplots are used to show relationship (causal relationship or covariance) between two *quantitative* variables. The data consists of a number of pairs of coordinates (x, y). Each pair of coordinates is indicated by a dot or circle in a system of coordinates."

In the next example, we resume work with the iris data introduced in the previous chapter.

Example 1:

```
> head(iris)

  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1         3.5          1.4         0.2  setosa
2          4.9         3.0          1.4         0.2  setosa
3          4.7         3.2          1.3         0.2  setosa
4          4.6         3.1          1.5         0.2  setosa
5          5.0         3.6          1.4         0.2  setosa
6          5.4         3.9          1.7         0.4  setosa

> slength <- iris[, 1]
> swidth <- iris[, 2]
> species <- iris[, 5]
> set.seed(1234)
> par(mfrow = c(2, 2))
> plot(slength, swidth,
+   xlim = c(4.0, 8.0), ylim = c(2.0, 5.0))
> plot(slength, swidth, pch = 21, bg = unclass(species),
+   xlim = c(4.0, 8.0), ylim = c(2.0, 5.0))
> legend("topright", levels(species), pch = 21, pt.bg = 1:3)
> plot(jitter(slength), jitter(swidth), pch = 22, bg = unclass(species),
```

```
+   xlim = c(4.0, 8.0), ylim = c(2.0, 5.0))
> legend("topright", levels(species), pch = 22, pt.bg = 1:3)
> plot(jitter(slength, amount = 0.05), jitter(swidth, amount = 0.05), pch = unclass(sp
+   xlim = c(4.0, 8.0), ylim = c(2.0, 5.0))
> legend("topright", levels(species), pch = 1:3)
```

Scatterplots in ggplot2:

```
> library(ggplot2)
> library(gridExtra)
> g1 <- ggplot(iris, aes(x = Sepal.Length, y = Sepal.Width)) +
+   geom_point()
> g2 <- ggplot(iris, aes(x = Sepal.Length, y = Sepal.Width, color = Species)) +
+   geom_point() +
+   theme(legend.position = "bottom")
> g3 <- ggplot(iris, aes(x = Sepal.Length, y = Sepal.Width, color = Species, shape = S
+   geom_point() +
+   theme(legend.position = "bottom")
> g4 <- ggplot(iris, aes(x = Sepal.Length, y = Sepal.Width, color = Species)) +
+   geom_point(size = 2, shape = 24) +
+   geom_jitter() +
+   theme(legend.position = "bottom")
> grid.arrange(g1, g2, g3, g4, nrow = 2)
```

Sepal.Width
Sepal.Length
4.5
4.0
3.5
3.0
2.5
2.0
5
6
7
8
Species
setosa
versicolor
virg

In the next example, we look at samples of sizes 10, 100, 1,000, 10,000, 100,000, and 1,000,000. from a bivariate normal distribution with correlation $\rho = 0$. Up to 1,000 observations, we can still identify regions of higher and lower density, but for 10,000 (and more) observations, we just have a black center in the plot. We will need some different graphical representation when we have too many observations (and thus too much overplotting) for an ordinary scatterplot.

Example 2:

```
> set.seed(1234)
> par(mfrow = c(2, 3), pty = "s", pch = ".")
> x10 <- rnorm(10)
> y10 <- rnorm(10)
> plot(x10, y10,
+   xlim = c(-4.0, 4.0), ylim = c(-4.0, 4.0))
> x100 <- rnorm(100)
> y100 <- rnorm(100)
> plot(x100, y100,
+   xlim = c(-4.0, 4.0), ylim = c(-4.0, 4.0))
> x1000 <- rnorm(1000)
> y1000 <- rnorm(1000)
> plot(x1000, y1000,
+   xlim = c(-4.0, 4.0), ylim = c(-4.0, 4.0))
> x10000 <- rnorm(10000)
> y10000 <- rnorm(10000)
> plot(x10000, y10000,
+   xlim = c(-4.0, 4.0), ylim = c(-4.0, 4.0))
> x100000 <- rnorm(100000)
> y100000 <- rnorm(100000)
> plot(x100000, y100000,
+   xlim = c(-4.0, 4.0), ylim = c(-4.0, 4.0))
> x1000000 <- rnorm(1000000)
> y1000000 <- rnorm(1000000)
> plot(x1000000, y1000000,
+   xlim = c(-4.0, 4.0), ylim = c(-4.0, 4.0))
```

y10
x10
y100
x100
y1000
x1000
y10000
x10000
y100000
x100000
y1000000
x1000000
−4 −2 0 2 4

1.2 Contour Plots and Bivariate Density Plots

Several suggestions for baseR and ggplot were obtained at this web page: `https://stats.stackexchange.com/questions/31726/scatterplot-with-contour-heat-overlay`

```
> library(MASS)
> # some pretty colors
> library(RColorBrewer)
> my.cols <- brewer.pal(9, "YlOrRd")
> par(mfrow = c(1, 2))
> # compute 2D kernel density, see MASS book, pp. 130-131
> z10000 <- kde2d(x10000, y10000, n = 50)
> z100000 <- kde2d(x100000, y100000, n = 50)
> plot(x10000, y10000, xlim = c(-4.0, 4.0), ylim = c(-4.0, 4.0),
+      xlab = "X label", ylab = "Y label", pch = 19, cex = 0.4)
> contour(z10000, drawlabels = FALSE, nlevels = 9, col = my.cols, add = TRUE, lwd = 5)
> plot(x100000, y100000, xlim = c(-4.0, 4.0), ylim = c(-4.0, 4.0),
+      xlab = "X label", ylab = "Y label", pch = 19, cex = 0.4)
> contour(z100000, drawlabels = FALSE, nlevels = 9, col = my.cols, add = TRUE, lwd = 5
```

```
> library(ggplot2)
> library(gridExtra)
> df10000 <- data.frame(x = x10000, y = y10000)
> commonTheme <- list(labs(color = "Density",
+                          fill = "Density",
+                          x = "X label",
+                          y = "Y label"),
+                     theme_bw(),
+                     theme(legend.position = c(0.02, 0.98),
+                           legend.justification = c(0, 1)))
> g1 <- ggplot(data = df10000, aes(x, y)) +
+   xlim(-4, 4) +
+   ylim(-4, 4) +
+   geom_point() +
+   geom_density2d(aes(colour = ..level..)) +
```

```
+   scale_colour_gradient(low = "yellow", high = "red") +
+   commonTheme
> g1

> g2 <- ggplot(data = df10000, aes(x, y)) +
+   xlim(-4, 4) +
+   ylim(-4, 4) +
+   geom_point() +
+   scale_fill_continuous(low = "yellow", high = "red") +
+   stat_density2d(aes(fill = ..level.., alpha = ..level..), geom = "polygon", colour =
+   guides(alpha = "none") +
+   commonTheme
> g2

> g3 <- ggplot(data = df10000, aes(x, y)) +
+   xlim(-4, 4) +
+   ylim(-4, 4) +
+   stat_density_2d(geom = "point", aes(size = ..density..), n = 20, contour = FALSE) +
+   guides(alpha = "none") +
+   commonTheme
> g3

> grid.arrange(g1, g2, g3, nrow = 2)
```

Density
0.16
0.12
0.08
0.04
Y label
X label
Density
0.16
0.12
0.08
0.04
Y label
X label
density
0.04
0.08
0.12
0.16
Y label
X label

1.3 Hexagon Binning

(Based on ?)

The R help page for hexbin indicates:

> "Creates a "hexbin" object. Basic components are a cell id and a count of points falling in each occupied cell."

Let us apply hexbin to data from our bivariate normal distribution for 10,000 and 1,000,000 observations.

Example 3:

```
> library(hexbin)
> library(colorspace)
> plot(hexbin(x10000, y10000))
```

```
> plot(hexbin(x10000, y10000), colramp = heat.colors)
```

```
> plot(hexbin(x10000, y10000), style = "lattice")
```

```
> plot(hexbin(x10000, y10000), style = "centroid")
```

```
> plot(hexbin(x10000, y10000), colramp = terrain.colors)
```

```
> plot(hexbin(x10000, y10000, xbins = 16), colramp = topo.colors)
```

Hexagonal binning in ggplot2:

```
> library(ggplot2)
> library(gridExtra)
> g1 <- ggplot(data = df10000, aes(x, y)) +
+   xlim(-4, 4) +
+   ylim(-4, 4) +
+   geom_hex()
> g1

> g2 <- ggplot(data = df10000, aes(x, y)) +
+   xlim(-4, 4) +
+   ylim(-4, 4) +
+   geom_hex(binwidth = c(0.5, 0.5))
> g2
```

```
> grid.arrange(g1, g2, nrow = 2)
```

1.4 Bivariate Histograms

Execute this code once to create the bivhist3d function, written by Mike Minnotte:

```
> bivhist3d <- function(x, y, nbins = nclass.Sturges(x),
+                       col = c(heat.colors(19)), xlab = "x", ylab = "y")
+
+   # Bivariate histogram plot.  Uses the gplots and rgl libraries.
+   # Written by Mike Minnotte
+ {
+   h2d <- hist2d(x, y, nbins, show = FALSE)
+   xb <- h2d$x
+   yb <- h2d$y
+   count <- h2d$counts
+   xs <- (xb[2] - xb[1]) / 2
+   ys <- (yb[2] - yb[1]) / 2
+   xb <- xb + xs
+   yb <- yb + ys
+
+   if (length(col) > 1)
+     zcol <- matrix(cut(count, length(col), labels = FALSE), length(xb))
+   else
+     zcol <- matrix(1, length(xb), length(yb))
+
+   cube <- cube3d()
+   clear3d()
+   bg3d(col = "grey")
+
+   xr <- max(xb) - min(xb)
+   yr <- max(yb) - min(yb)
+   zr <- max(count)
+   mr <- max(xr, yr, zr) * 0.8
+
+   aspect3d(mr / xr, mr / yr, mr / zr)
+
+   for (i in 1:length(xb))
+     for (j in 1:length(yb))
+     {
+       scube <- scale3d(cube, xs, ys, count[i, j] / 2)
+       tcube <- translate3d(scube, xb[i], yb[j], count[i, j] / 2)
+       if (count[i, j] == 0)
+         shade3d(tcube, color = "black")
+       else
+         shade3d(tcube, color = col[zcol[i, j]])
+     }
+
+   axes3d()
+   title3d(xlab = xlab, ylab = ylab, zlab = "count")
+   return()
+ }
```

Let us apply bivhist3d to data from our bivariate normal distribution for 10,000 observations.

```
> library(gplots)
> library(rgl)
> bivhist3d(x10000, y10000)

NULL

> bivhist3d(x10000, y10000, col = "red")

NULL

> bivhist3d(x10000, y10000, col = terrain.colors(15), nbins = 20)

NULL
```

If you want to create a snapshot of an interesting 3D view, use the command:

```
> #rgl.snapshot("bivhist1.png")
```

Finally, let us do some automatic animation, based on `https://stat.ethz.ch/pipermail/r-devel/2014-April/068817.html`:

```
> bivhist3d(x10000, y10000, col = terrain.colors(15), nbins = 20)

NULL

> U <- par3d("userMatrix")
> for (theta in seq(0, 2 * pi, len = 100)) {
+    par3d(userMatrix = rotationMatrix(theta, 0, 1, 0) %*% U ) # Rotate about the z-ax
+    Sys.sleep(0.1)
+ }
```

1.5 Further Reading

Additional sources for bivariate plots are:

- To be announced.

Scans//Cartoonstock_RoomBelow.png

Figure 3: http://www.cartoonstock.com/cartoonview.asp?start=4&search=main&catref=bven597&MA_Artist=&MA_Category=&ANDkeyword=statistics&ORkeyword=&TITLEkeyword=&NEGATIVEkeyword=, Cartoon.

— THE END —

Scans//Cartoonstock_FallingArrow.jpg

Figure 4: http://www.cartoonstock.com/blowup_stock.asp?imageref=vsh0184&artist=Shirvanian,+Vahan&topic=statistics+, Cartoon.