R for Public Health

Monday, February 17, 2014

ggplot2: Cheatsheet for Visualizing Distributions

In the third and last of the ggplot series, this post will go over interesting ways to visualize the distribution of your data. I will make up some data, and make sure to set the seed.

```
library(ggplot2)
library(gridExtra)
set.seed(10005)

xvar <- c(rnorm(1500, mean = -1), rnorm(1500, mean = 1.5))
yvar <- c(rnorm(1500, mean = 1), rnorm(1500, mean = 1.5))
zvar <- as.factor(c(rep(1, 1500), rep(2, 1500)))
xy <- data.frame(xvar, yvar, zvar)
```

>> Histograms

I've already done a post on histograms using base R, so I won't spend too much time on them. Here are the basics of doing them in ggplot. More on all options for histograms here.

The R cookbook has a nice page about it too: **http://www.cookbook-r.com/Graphs/Plotting_distributions_(ggplot2)/**

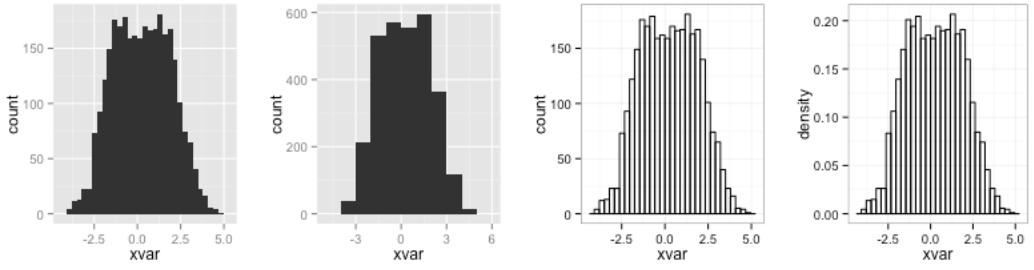
Also, I found this really great aggregation of all of the possible geom layers and options you can add to a plot. In general the site is a great reference for all things ggplot.

```
#counts on y-axis
g1<-ggplot(xy, aes(xvar)) + geom_histogram()
#horribly ugly default
g2<-ggplot(xy, aes(xvar)) + geom_histogram(binwidth=1)
#change binwidth
g3<-ggplot(xy, aes(xvar)) + geom_histogram(fill=NA, color="black") +
theme_bw()   #nicer looking

#density on y-axis
g4<-ggplot(xy, aes(x=xvar)) + geom_histogram(aes(y = ..density..),
color="black", fill=NA) + theme_bw()

grid.arrange(g1, g2, g3, g4, nrow=1)
```

```
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust
## this. stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to
## adjust this. stat_bin: binwidth defaulted to range/30. Use 'binwidth = x'
## to adjust this.
```



Notice the warnings about the default binwidth that always is reported unless you specify it yourself. I will remove the warnings from all plots that follow to conserve space.

>> Density plots

We can do basic density plots as well. Note that the default for the smoothing kernel is gaussian, and you can change it to a number of different options, including **kernel="epanechnikov"** and **kernel="rectangular"** or whatever you want. You can find all of those options here.

```
#basic density
p1<-ggplot(xy, aes(xvar)) + geom_density()

#histogram with density line overlaid
p2<-ggplot(xy, aes(x=xvar)) +
```

Data and Code Download

All data and code for this blog can be downloaded here:

Data and Code Download Site

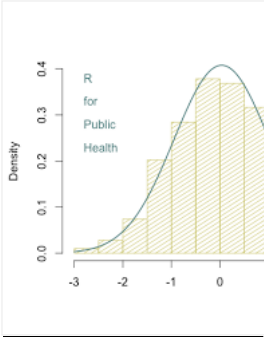
NB: It's been pointed out to me that s images don't show up on IE, so you'l switch to Chrome or Firefox if you are IE. Thanks!

Why R for public health?

I created this blog to help public hea researchers that are used to Stata or begin using R. I find that public healt unique and this blog is meant to add specific data management and analy needs of the world of public health.

R is a very powerful tool for program can have a steep learning curve. In r experience, people find it easier to d long way with another programming language, rather than try R, because takes longer to learn. I think all statis packages are useful and have their p the public health world. However, I a strong proponent of R and I hope this can help you move toward using it w makes sense for you.

Please email me with posts you wou see or R questions, and I'll try my bes answer them. Thanks for following!



Blog Archive

Search This Blog

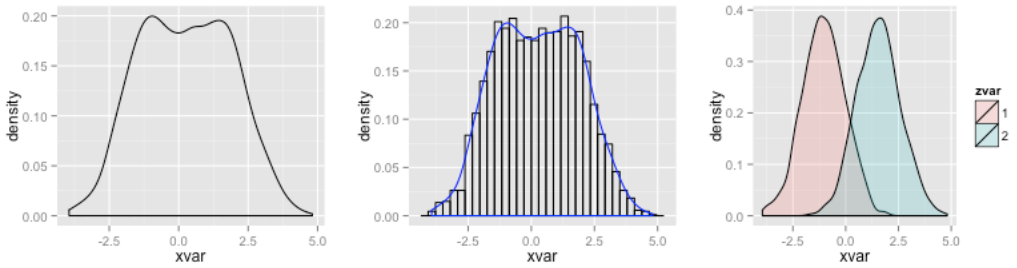
Labels

aggregate (2) animation (1) apply (6) categorical (2) cbind (3) character (3)

```
  geom_histogram(aes(y = ..density..), color="black", fill=NA) +
  geom_density(color="blue")

#split and color by third variable, alpha fades the color a bit
p3<-ggplot(xy, aes(xvar, fill = zvar)) + geom_density(alpha = 0.2)

grid.arrange(p1, p2, p3, nrow=1)
```



>> Boxplots and more

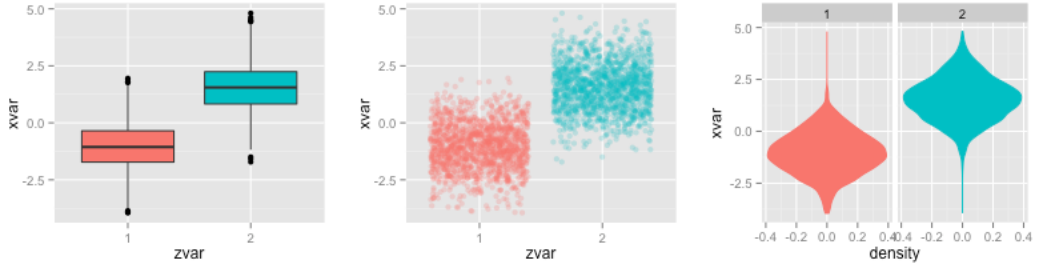
We can also look at other ways to visualize our distributions. Boxplots are probably the most useful in order to describe the statistics of a distribution, but sometimes other visualizations are nice. I show a jitter plot and a violin plot. More on boxplots here. Note that I removed the legend from each one because it is redundant.

```
#boxplot
b1<-ggplot(xy, aes(zvar, xvar)) +
  geom_boxplot(aes(fill = zvar)) +
  theme(legend.position = "none")

#jitter plot
b2<-ggplot(xy, aes(zvar, xvar)) +
  geom_jitter(alpha=I(1/4), aes(color=zvar)) +
  theme(legend.position = "none")

#violin plot
b3<-ggplot(xy, aes(x = xvar)) +
  stat_density(aes(ymax = ..density..,  ymin = -..density..,
               fill = zvar, color = zvar),
               geom = "ribbon", position = "identity") +
  facet_grid(. ~ zvar) +
  coord_flip() +
  theme(legend.position = "none")

grid.arrange(b1, b2, b3, nrow=1)
```

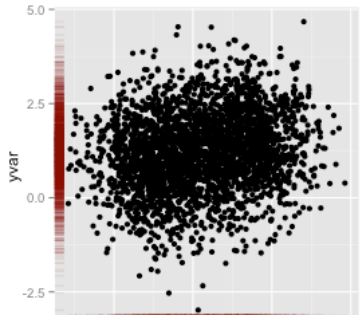


>> Putting multiple plots together

Finally, it's nice to put different plots together to get a real sense of the data. We can make a scatterplot of the data, and add marginal density plots to each side. Most of the code below I adapted from this StackOverflow page.

One way to do this is to add distribution information to a scatterplot as a “rug plot”. It adds a little tick mark for every point in your data projected onto the axis.

```
#rug plot
ggplot(xy,aes(xvar,yvar))  + geom_point() + geom_rug(col="darkred",alpha=.1)
```



clustered (1) continuous (2) CSV (1) cut (1) (7) date (1) density (1) DHS (1) distributior message (2) Export (2) factor (1) functi ggplot (4) ggplot2 (4) graph (4) his ifelse (2) Import (1) knitr (1) label (1) latex logical (1) loop (1) matrix (6) names (2) numeric (4) plot (7) power (1) R regression (5) reshape (1) sample size scatterplot (4) standard errors (1) stargaze (4) subset (5) summary (5) table (3) time (1) vector (5) xtable (1)

About Me

Slawa Rokicki

I am a doctoral student i Policy at Harvard Univer interested in research in maternal health, and esp reproductive health.

View my complete profile

Follow @slawarokicki

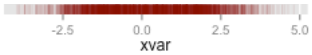
Other great R sites and blogs

- R Bloggers: aggregate of many R
- DiffusePrioR - more advanced sta
- R Graph Gallery
- Statmethods - Data mgmt, graphs
- Stattler - Useful for graphs

Follow by Email

Email address...

g+1 21



Another way to do this is to add histograms or density plots or boxplots to the sides of a scatterplot. I followed the stackoverflow page, but let me know if you have suggestions on a better way to do this, especially without the use of the empty plot as a place-holder.

I do the density plots by the zvar variable to highlight the differences in the two groups.

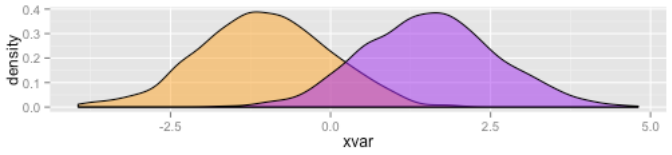
```
#placeholder plot - prints nothing at all
empty <- ggplot()+geom_point(aes(1,1), colour="white") +
     theme(
       plot.background = element_blank(),
       panel.grid.major = element_blank(),
       panel.grid.minor = element_blank(),
       panel.border = element_blank(),
       panel.background = element_blank(),
       axis.title.x = element_blank(),
       axis.title.y = element_blank(),
       axis.text.x = element_blank(),
       axis.text.y = element_blank(),
       axis.ticks = element_blank()
     )

#scatterplot of x and y variables
scatter <- ggplot(xy,aes(xvar, yvar)) +
  geom_point(aes(color=zvar)) +
  scale_color_manual(values = c("orange", "purple")) +
  theme(legend.position=c(1,1),legend.justification=c(1,1))

#marginal density of x - plot on top
plot_top <- ggplot(xy, aes(xvar, fill=zvar)) +
  geom_density(alpha=.5) +
  scale_fill_manual(values = c("orange", "purple")) +
  theme(legend.position = "none")

#marginal density of y - plot on the right
plot_right <- ggplot(xy, aes(yvar, fill=zvar)) +
  geom_density(alpha=.5) +
  coord_flip() +
  scale_fill_manual(values = c("orange", "purple")) +
  theme(legend.position = "none")

#arrange the plots together, with appropriate height and width for each row
and column
grid.arrange(plot_top, empty, scatter, plot_right, ncol=2, nrow=2,
widths=c(4, 1), heights=c(1, 4))
```



Posted by Slawa Rokicki at 7:21 PM

Labels: boxplot, density, distributions, ggplot, ggplot2, graph, histogram, plot, R, scatterplot

9 comments:

Myles February 19, 2014 at 8:25 AM

Wow, Slawa this is a great resource! Thanks so much for putting this together.

Also I think 3rd plot under 'Boxplots' is a not a volcano plot, but "violin plot":
http://en.wikipedia.org/wiki/Violin_plot

Great stuff.

Reply

Replies

Slawa Rokicki February 19, 2014 at 8:29 AM

Thanks! Thanks for the comment; I changed it.

Reply

Raphael February 19, 2014 at 10:39 AM

You should check out beanplots, which are basically violin plots, with superimposed boxplots and dot plots. There is a beanplot package for R, but ggplot2 does not include a geom specifically for this. You can easily create one by using geom_violin, geom_boxplot, and geom_point.

Reply

Replies

Slawa Rokicki February 19, 2014 at 12:34 PM

Yes, I think that's really the beauty of ggplot2 and what I've tried to convey over three posts about it is the idea of layering. You can superimpose layers of points, boxplots, and whatever else you want very easily once you know how to build the different components.

Reply

baptiste auguie February 19, 2014 at 5:36 PM

> "It's really nice that grid.arrange() clips the plots together so that the scales are automatically the same. "

That's not the case, and for this very reason I wouldn't recommend using grid.arrange when the axes ought to be aligned. Consider using gtable instead, e.g http://stackoverflow.com/a/21531303/471093

Reply

Hichem Fenghour February 20, 2014 at 9:01 AM

I like this blog
Thank you for sharing.
hichem

Reply

Pavani Smiley March 21, 2014 at 10:41 AM

its really nice ehealthy

Reply

fahim hassan August 15, 2014 at 10:40 AM

Such a great post, Slawa!
Tips for the readers - if you are interested in customizing your graphs in ggplot, checkout this blog post in R bloggers - http://www.r-bloggers.com/how-to-customize-ggplot2-graphics/

Reply

Rabbi Sardar March 10, 2015 at 3:12 PM

That is an extremely smart written article. I will be sure to bookmark it and return to learn extra of your useful information. Thank you for the post. I will certainly return.

Personalized Kids Scrubs
Toddler scrubs

Reply

Enter your comment...

Comment as: Google Accou

Publish Preview

Newer Post Home Older Post

Subscribe to: Post Comments (Atom)