Plots!

Maria Kamenetsky
Friday, October 30, 2015

Graphs!

```
#1) Regular boxplots
q <- ggplot(df, aes(factor(Pal), SA))</pre>
q + geom_boxplot(outlier.colour="orange", outlier.size=3, aes(fill=factor(Pal))) + coord_flip() + face
#2) Boxplots with jittering
r <- ggplot(data=sub, aes(Conc,DevA))
r + geom_boxplot(outlier.colour="orange", outlier.size=6, aes(fill=Conc)) +
    geom_jitter() + coord_flip() + facet_grid(Pop~.)
  • GeomPoint
q <- ggplot(BA, aes(SA, DD))
q + geom_point() + coord_flip() + facet_grid(Dec ~ Pal) #+ stat_smooth(method=lm, fullrange = TRUE)
  • Histogram
#1) histogram on log scale
ggplot(df, aes(LtoSM)) + geom_histogram() + scale_x_log10()
#2) Facetted Histogram
ggplot(svN, aes(sv, fill = reg)) + geom_histogram(binwidth=1) + facet_grid(rg ~ reg + soft)
#3) Histogram
p <- qplot(mathkind, data=classroom, geom="histogram",</pre>
           xlab="Mathematics Score in Kindergarten", main="Kindergarten Mathematics Score",
           binwidth = 10)
p + geom_histogram(binwidth=10)
  \bullet qplot
qplot(R1,it, data=z)
  • statsmooth
p <- ggplot(z, aes(R2, it))
p + scale_y_continuous(limits=c(0,15)) +stat_smooth()
\#smoothing method (function) to use, eg. lm, glm, gam, loess, rlm. For datasets with n < 1000 default i
```

• bar chart

```
#Classic Barchart
qplot(factor(md), data=z, geom="bar", fill=factor(iG))

#Barchart with x-axis labels on 45 degree angle
q <-qplot(factor(md), data=z, geom="bar", fill=factor(reg))
q + theme(axis.text.x = element_text(angle = 45, hjust = 1))

#Facetted Bar Charts
ggplot(z, aes(md, fill=factor(reg))) + geom_bar() + facet_grid(rg ~ cf) + coord_flip()</pre>
```

• Histograms

```
#Facetted Histograms
ggplot(z, aes(md, fill=factor(reg))) + geom_bar() + facet_grid(rg ~ cf) + coord_flip()
```

• Density Plots

• ScatterPlots

```
#Scatter plot with point-wise confidence intervals
p <- qplot(mathgain, mathkind, data=classroom, geom=c("smooth"),</pre>
           method="loess", xlab="Mathematics Gain for Students in Sample",
           ylab="Mathematics Score in Kindergart.",
           main="Scatterplot of 'Mathgain' vs. 'Mathkind'")
p + geom_point() + stat_smooth(se=TRUE) +
  geom smooth(method="loess") +
  labs(title="'Mathgain' vs. 'Mathkind'")
#ScatterPlot with Regression Line (and no confidence bands)
p <- ggplot(classroom, aes(x=mathgain,y=mathkind),</pre>
            geom_smooth(method="gam", formula = mathkind~mathgain),
            stat_smooth(se=FALSE),
            xlab="Mathematics Gain for Students in Sample",
            ylab="Mathematics Score in Kindergarten")
p + geom_point() + stat_smooth(se=FALSE) +
  labs(title="Scatterplot of 'Mathgain' vs. 'Mathkind'") +
  geom_smooth(method= "gam")
#Log Scatter Plot with GLM Smoother
v <- ggplot(ufc, aes(x=log(Height), y = log(Dbh))) + geom_point()</pre>
v + stat smooth(se=TRUE) + geom smooth(method="glm", formula = log(Dbh)~log(Height))
```

• Dot Plot

GGMAP

• Getting the Map

```
map <- get_map(location='united states', zoom=4, maptype='toner')
ggmap(map) + geom_point(aes(x=longitude, y= latitude, size=(it)), data=z, alpha=0.5)
#I supplied the long/lat here from my dataset</pre>
```

• Plotting points

```
#1) Basic Map
ggmap(map) + geom_point(aes(x=longitude, y= latitude, color=densityR1, fill=densityR1), data=z, alpha=0
#2) Facetted Map (with colours specified)
ggmap(map) + geom_point(aes(x=longitude, y= latitude, colour=it), size = 3, data=z, alpha=0.5) + facet
```

• Keep Unit Aspect Ratio

```
+ theme(aspect.ratio = 1)
```

• Change Color Gradient and Point Size

```
#1) Change color gradient Scale and specify which will be low
+ scale_colour_gradient(limits=c(3,4), low="red")

#2) Change with color brewer
+ scale_colour_brewer(type="div",palette="Set1")

#3) Point size (size=3)
ggmap(map) + geom_point(aes(x=longitude, y= latitude, color=sv), data=svN, colour="red", size=4)
```

• density map

Maps with Spatial Polygons

```
stP$name <- tolower(stP$stL)
stP <- stP[order(stP$name),] # reorder alpha
stP <- subset(stP, (name!="hawaii" & name!="alaska"))
us <- map("state", plot=FALSE, fill=TRUE)
us.ids <- sapply(strsplit(us$names,":"), function(x) x[1])
us.sp <- map2SpatialPolygons(us, us.ids, CRS("+proj=longlat + datum=wgs84"))

IDs <- match(stP$name, row.names(us.sp))
us.spNew <- us.sp[IDs]
row.names(stP) <- row.names(us.spNew) #these need to match
us.spdf <- SpatialPolygonsDataFrame(us.spNew, data=as.data.frame(stP)) # rownames of state need to be u

#Plot
spplot(us.spdf, "reg", main="Title")

#Exclude Obs and Plot
exclude <- us.spdf[!(us.spdf$stsB==1888 | us.spdf$stsB==1734),]
sp5 <-spplot(exclude, "stsB",cuts=99, main="Title")</pre>
```

• Plotting Multiple ggplots into a Grid

```
library(grid)
library(automap)
vplayout <- function(x, y) viewport(layout.pos.row = x, layout.pos.col = y)

plot1 <- ggplot(z, aes(phR,it)) + geom_boxplot(outlier.colour="orange", outlier.size=3, aes(fill=factor plot2 <- ggplot(z, aes(trR,it)) + geom_boxplot(outlier.colour="orange", outlier.size=3, aes(fill=factor plot3 <- ggplot(z, aes(rrR,it)) + geom_boxplot(outlier.colour="orange", outlier.size=3, aes(fill=factor grid.newpage()
pushViewport(viewport(layout = grid.layout(2,2)))
print(plot1, vp=vplayout(1,1))
print(plot2, vp=vplayout(1,2))
print(plot3, vp=vplayout(2,1))</pre>
```

ggplot2 - General Tips and Tricks

• Suppress Plotting NA's

```
#outlier.shape=NA
r + geom_boxplot(outlier.shape=NA, aes(fill=factor(phR))) + coord_flip() + scale_y_continuous(limits=c
```

• Setting Limits on X/Y Coords in ggplot

```
#for y (or x) limits
+ scale_y_continuous(limits=c(0,50))
```

• Dealing with NA's in DF to Plot

```
#Solution - create sub data frame without NA's

#Example: use ggplot(na.omit(subset))
subz <- z[c("it", "densityR1", "iG")]
b <- ggplot(na.omit(subz), aes(factor(densityR1), it))
b + geom_boxplot(aes(fill=factor(densityR1))) + scale_y_continuous(limits=c(0,100)) + coord_flip() + fa</pre>
```

Neat Random Tricks

• Using texreg Package for Publication-Quality md Output Comparisons

```
screenreg(list(mod11F, mod11IR, mod11P, mod11Ss)) #this makes output pretty on the screen
texreg(list(mod11F, mod11IR, mod11P, mod11Ss)) # this makes output pretty in latex
```

• Other Tables

```
xtabs(disease/total ~ sex+food, babyfood)
or xtable
or
table
```

Sweet R Packages

- {tidyr} = some data wrangling things; has some SQL style things
- $\{ggplot2\} = literally the best$
- {gridExtra} = need this if you want to create panels with ggplot that aren't automated (where you can just facet-grid/wrap)
- {ggmap} = mapping with ggmpa
- {mapproj} = needed to get maps using spatial polygons
- {stats} = used for function aggregate
- {texreg} = used for fancy tex output
- reshape = this one makes more sense and is stata-like
- reshape2 = this one contains 'melt'
- xtable