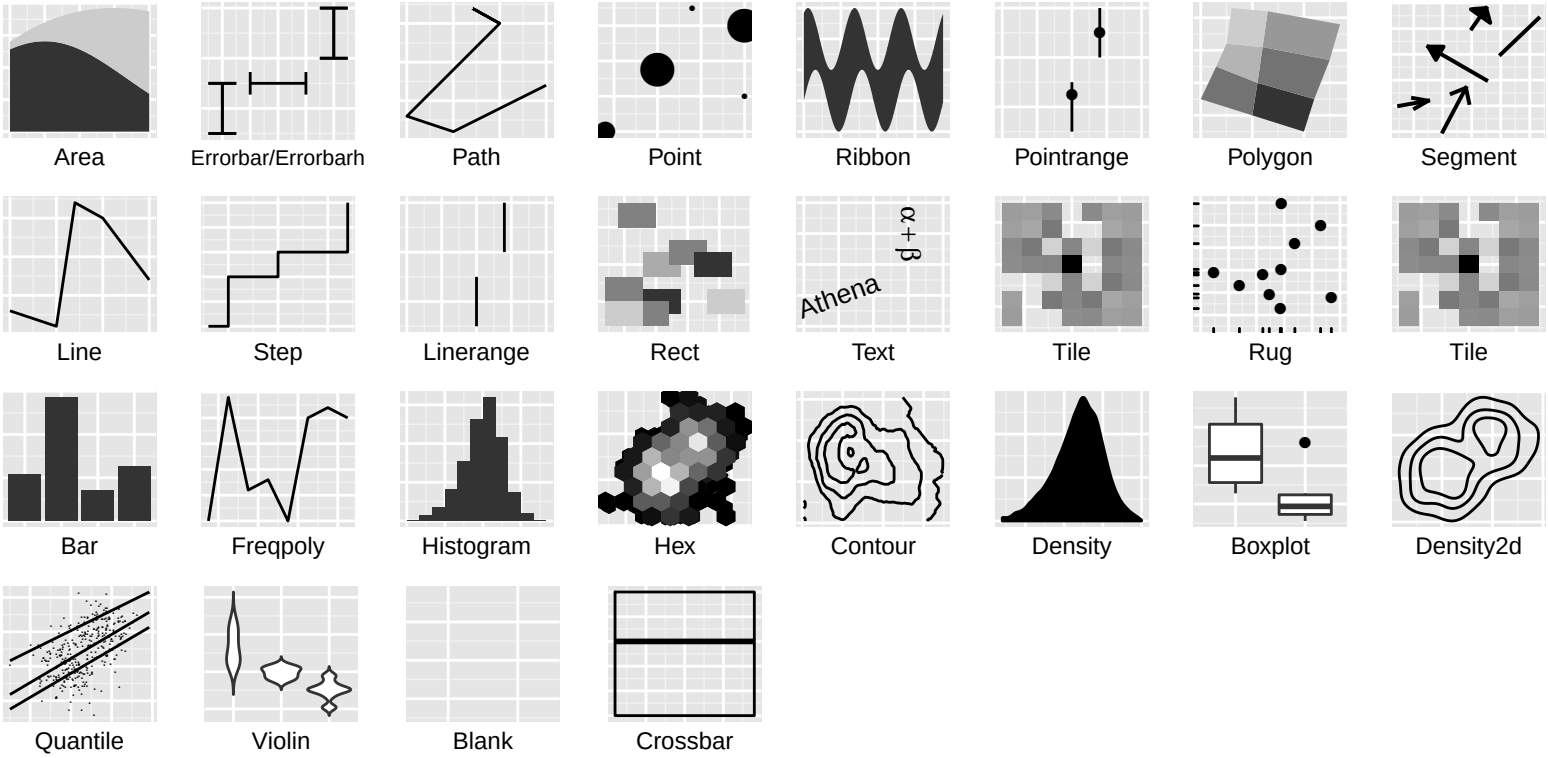
library('ggplot2')
library('grid') # has arrow, unit

ggplot(*data=dataframe, mapping=aes()*)
+geom_XXX(*mapping = aes( ), data = dataframe*, stat = , position =, other options)
+stat_XXX(*mapping = aes( ), data = dataframe*, geom = , position =, other options)
+facet_grid(.∼X or X∼. or X∼Y)+facet_wrap(∼X)
+scale_XXX_TYPE()
+theme(item=element_XXX() or other)
+labs(title=,x=,y=, color=)+xlab()+ylab()
+theme_XXX: bw, grey, classic, minimal
+xlim()+ylim+coord_cartesian(xlim=,ylim=)

qplot(x, y = NULL, ..., data, facets = NULL, margins = FALSE, geom = "auto", stat = list(NULL), position = list(NULL), xlim = c(NA, NA), ylim = c(NA, NA), log = "", main = NULL, xlab = deparse(substitute(x)), ylab = deparse(substitute(y)), asp = NA)

GEOMS

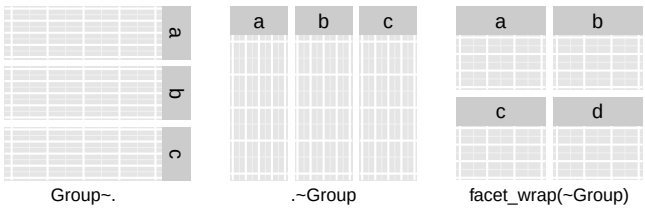


Other Geoms: abline, hline, vline, jitter, raster, smooth

Positions: dodge, fill, identity, jitter, stack

STATS

bin	bin2d	bindot	binhex	boxplot	contour	density	density2d	ecdf	
function	hline	identity	qq	quantile	smooth	spoke	sum	summary	summary_hex
summary2d	unique	vline	ydensity						



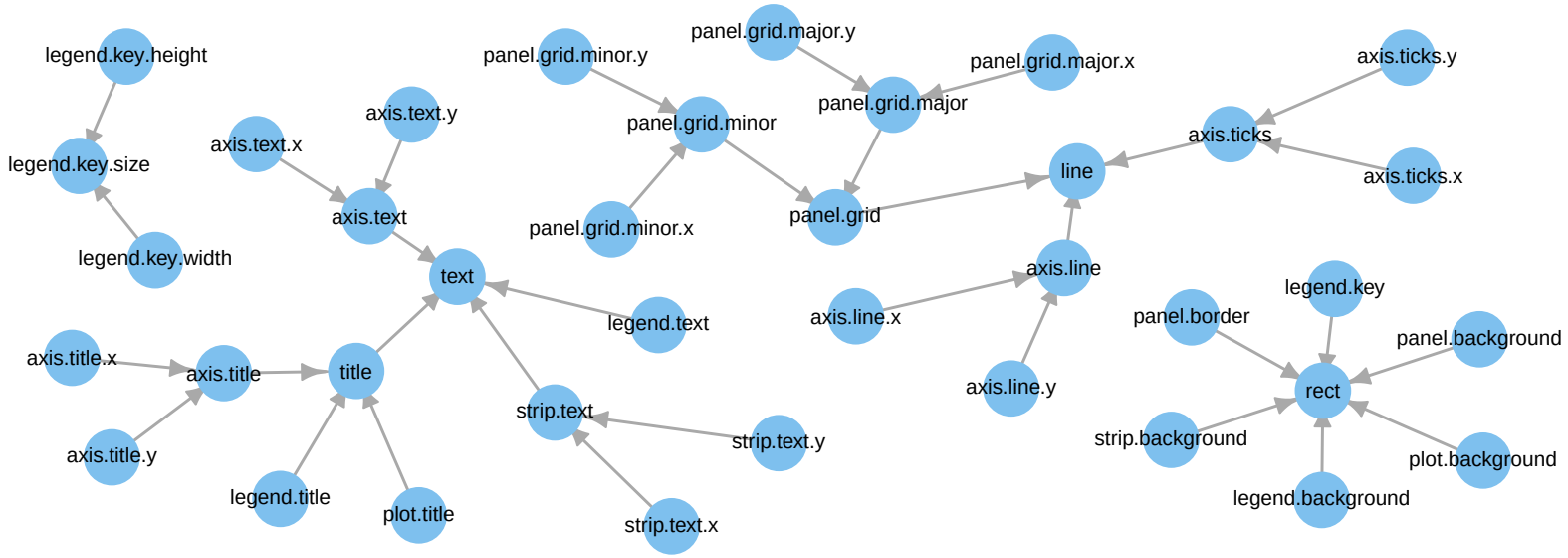
FACETS

SCALES: AXES and LEGENDS

		Discrete	Continuous
	fill,color	hue, brewer, grey, identity, manual	gradient, gradient2, gradientn
	x, y	discrete	continuous, date
scale	shape	shape, identity, manual	
	linetype	linetype, identity, manual	
	size	identity, manual	size

name, breaks, labels, limits, trans, pal, values

APPEARANCE +theme(item=element_XXX() or other)



element_line(colour, size, linetype, lineend)
element_rect(fill, colour, size, linetype)
element_text(family, face, colour, size, hjust, vjust, angle, lineheight)
element_blank()

axis.ticks.length, axis.ticks.margin, legend.margin, legend.key.size, panel.margin, plot.margin, legend.key.height, legend.key.width= unit(x,"unit")
units: npc (default. width/height=1, center=(.5,.5)), cm, in, points, lines, native, grobheight/grobwidth,...

legend.title.align= (number from 0 (left) to 1 (right)) **legend.position**=("left", "right", "bottom", "top", "none",or two-element numeric vector) **legend.direction**=("horizontal" or "vertical") **legend.justification**=("center" or two-element numeric vector) **legend.box** - arrangement of multiple legends =("horizontal" or "vertical") **legend.box.just** -justification of each legend box, when there are multiple legends =("top", "bottom", "left", or "right")

OTHER
ggsave(filename=,plot, height=,width=)
ggplot_build(p) # produces a list with data frame for each layer and a panel object
coord_cartesian, coord_equal, coord_flip, coord_trans, coord_polar(theta=,start=,direction=)
plot %+% new.df # Use a different data frame
+theme(legend.position="none") # supresses legend
viewport(x,y,width,height,just,angle), plot first then, print(p,vp=)
pie chart = polar+bar