GGPLOT Cheat Sheet

If you are looking for a very fast and efficient way to produce good-looking data visualizations that you can use to derive and communicate insights from your data sets, then look no further than R's ggplot2 package. This package was designed to help you create all different types of data graphics in R, including histograms, scatterplots, bar charts, box plots, and density plots.

Installation: Install.packages("ggplot2") **Calling the package**: library(ggplot2)
We are ready to unleash the visualization potential of ggplot2. We will be using inbuilt data sets available in R for creating graphs and plots.

ggplot(data = mpg, aes(x = cty, y = hwy)) Begins a plot that you finish by adding layers to.

Add a new layer to a plot with a geom_*() or stat_*() function.
Each provides a geom, a set of aesthetic mappings, and a default stat and position adjustment.
To Returns the last plot : last_plot()
To save the image: ggsave("plot.png", width = 5, height = 5)

ggplot2 is based on the grammar of graphics, the idea that you can build every graph from the same
few components: a data set, a set of geoms—visual marks that represent data points, and a coordinate. Use a geom to represent data points, use the geom's aesthetic properties to represent variables. Each function returns a layer.

Data to be used: **mpg and airquality,diamonds** that has few continuous and few categorical variables. **Str() and summary()** will give summary of data along with column type

One variable

Continuous: a <- ggplot(mpg, aes(hwy))

Discrete: b<-ggplot(mpg,aes(fl))

Two variables

Continuous X, Continuous Y

f <- ggplot(mpg, aes(cty, hwy))

Discrete X, Continuous Y

f <- ggplot(mpg, aes(cty, hwy))

Discrete X, Discrete Y

h <- ggplot(diamonds, aes(cut, color))

Labels

t + labs(title =" New title",x = "New x", y = "New y")

Faceting:

Facets divide a plot into subplots based on the values of one or more discrete variables

t <- ggplot(mpg, aes(cty, hwy)) + geom_point()

facet into both rows and columns:

t + facet_grid(year ~ fl)

Legends

Place legend at "bottom", "top", "lef", or "right":

Continuous Bivariate Distribution

l <- ggplot(airquality, aes(Month, Ozone))

Themes

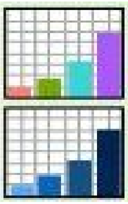
Scales

Scales control how a plot maps data values to the visual values of an aesthetic.

b<-ggplot(mpg,aes(fl))

n <- b + geom_bar(aes(fill = fl))

n + scale_fill_manual(values = c("skyblue", "royalblue", "blue", "navy"), limits = c("d", "e", "p", "r"), breaks =c("d", "e", "p", "r"),name = "fuel", labels = c("D", "E", "P", "R"))

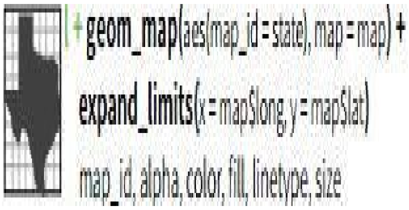


Maps

data <- data.frame(murder = USArrests$Murder, state = tolower(rownames(USArrests)))

map <- map_data("state")

l <- ggplot(data, aes(fill = murder))



```
+ geom_map(aes(map_id = state), map = map) +
expand_limits(x = map$long, y = map$lat)
map_id, alpha, color, fill, linetype, size
```