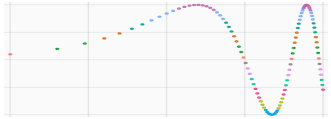
ggplot Reference Card v0.1

(c) 2014 John Lynch – john.lynch@iname.com



1 Introduction

ggplot was developed by Yhat to provide a system for plotting and visualization based on ggplot2 and the book "The Grammar of Graphics"
Details and examples can be found in the manual. https://github.com/yhat/ggplot/

Grammar represents an abstraction of objects aiming to shorten the distance from mind to page by mapping data to aesthetic attributes and geometric objects. Think of looking at how y changes with x across levels of z.

`pip install ggplot` to install
Load with `import ggplot` or `from ggplot import *`

2 Invoking Plot .

This form is the standard "grammar of graphics" method of plotting. Data is supplied in the form of a dataframe, columns of the data are bound to *aesthetics*, and plot elements including *scales, coordinates, statistics, guides, and geometries* are added to the plot.

```
ggplot(data::AbstractDataFrame, elements::Element...; mapping...)
```

```
p = ggplot(dfname, aes('c1','c2'))          generates plot object of c2 vs c1
p = ggplot(aes(x="c1", y="c2"), data=name)  same

p + geom_line(color='red') + ...            add geom to plot as a red line

p = (ggplot(df2, aes('x', 'diffs')) +       plot df2[diffs] with blue points
  geom_point(color='blue') +                  and also df[dy] as red lines
geom_line(df, aes('x', 'dy'), color='red')    without using pandas melt
```

3 Modifying Aesthetics

```
plot(..., color ="c3", ...)                 color based on categories in c3
plot('x', 'value', color = 'variable')      color based on melted dataframe
```

4 Geometries do the actual drawing

```
geom_point(color='blue')                    use points and optionally specify color
geom_point(alpha=0.05, color='red')           change transparency and color
geom_lines() geom_step()                    use lines
geom_bar()   geom_histogram(binwidth=x)
geom_density(color=, fill=True, alpha=0.5)
geom_jitter()   geom_text(label='this')
geom_tile(fill=x)
geom_area(ymin=l, ymax=h)
geom_abline()
geom_hline(y=y1) + geom_vline(x=x1)
geom_now_its_art()
```

4 Statistics transform one or more aesthetics and draw

```
stat_smooth()                               add loess smoothing of chart points
stat_smooth(span=.15, se=True)              adjust smoothing (.66 default)
                                            and turn standard error shading on
```

5 Guides draw graphics to support the visualization such as axis ticks, labels and keys

```
xlab("Time")   ylab("Info")                 set x label or y label
labs('Time', 'Ratio')                       set x and y labels
ggtitle('Amazing ggplot Example')
```

6 Scales transform the data to aid visualization

```
scale_x_continuous()
scale_x_discrete
scale_x_log   scale_y_log                   log10 axis
scale_x_reverse  scale_y_reverse            reverse axis
scale_color_manual(values=[color1,...])
scale_color_gradient(low=co1, high=co2)     with aes color set to values
scale_color_brewer()                        coming
scale_x_date
```

7 Showing Facets and Layers

```
facet_wrap('col_for_facetting')
facet_grid(col1, col2, scales='free_y')
```

8 Output to other Formats

```
ggsave(plotname, "filename.png")            Coming: PDF
```

9 Themes

Many parameters controlling the appearance of plots can be overridden by passing a Theme object to the plot function.

```
theme_gray()                default theme
theme()
theme_bw()
theme_seaborn()
theme_xkcd()
theme_matplotlib()
```

10 Pandas

stack on c1 vals:
```
df2 = pd.melt(df[['c1', 'c2', 'c3', 'c5']], id_vars='c1')
ggplot(df2, aes(x='c1', y='value', colour='variable'))
```