Data Visualization with ggplot2 :: CHEAT SHEET

Basics

ggplot2 is based on the grammar of graphics, the idea that you can build every graph from the same components: a data set, a coordinate system, and geoms—visual marks that represent data points.

to display values, map variables in the data to visual properties of the geom (aesthetics) like size, color, and x and y locations.

Geoms

Use a geom function to represent data points, use the geom’s aesthetic properties to represent variables. Each function returns a layer.

Graphical Primitives

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>a &lt;- ggplot(economics, aes(date, unemploy!))</td>
<td>Create a plot with the data and aesthetics</td>
</tr>
<tr>
<td>b &lt;- ggplot(seas, aes(x = long, y = lat))</td>
<td>Use the geom function</td>
</tr>
<tr>
<td>c &lt;- ggplot(mpg, aes(hwy))</td>
<td>Add one geom to the plot</td>
</tr>
</tbody>
</table>

One Variable Continuous

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>c + geom_area(aes(x = long, y = lat))</td>
<td>Add the geom to the plot</td>
</tr>
</tbody>
</table>

Two Variables Continuous

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>d + geom_point(aes(x = long, y = lat))</td>
<td>Add a point geom to the plot</td>
</tr>
</tbody>
</table>

Continuous Bivariate Distribution

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>d + geom_bin2d(aes(x = x, y = y))</td>
<td>Create a bivariate distribution plot</td>
</tr>
</tbody>
</table>

Continuous Function

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>e + geom_line(aes(x = long, y = lat))</td>
<td>Add a line to the plot</td>
</tr>
</tbody>
</table>

Visualizing Error

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>f + geom_point(aes(x = long, y = lat))</td>
<td>Add a point to the plot</td>
</tr>
</tbody>
</table>

Maps

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>k &lt;- ggplot(data, aes(x = long, y = lat))</td>
<td>Add a map to the plot</td>
</tr>
</tbody>
</table>

Three Variables

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>l + geom_contour(aes(x = long, y = lat))</td>
<td>Add a contour to the plot</td>
</tr>
</tbody>
</table>

Seals

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>m + geom_bar(aes(x = long, y = lat))</td>
<td>Add a bar to the plot</td>
</tr>
</tbody>
</table>

Cheat Sheet

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>n + geom_point(aes(x = long, y = lat))</td>
<td>Add a point to the plot</td>
</tr>
</tbody>
</table>

RStudio

RStudio® is a trademark of RStudio, Inc. • CC BY SA RStudio • info@rstudio.com • 844-448-1212 • rstudio.com • Learn more at http://ggplot2.tidyverse.org • ggplot2 2.1.0 • Updated: 2016-11
Stats

A stat builds new variables to plot (e.g., count, prop).

```r
stats + stat_unique()
```

Visualize a stat by changing the default stat of a geom function, `geom_bar(stat="count")` or by using a stat function, `stat_count(~bar)`, which calls a default geom to make a layer (equivalent to a geom function). Use `..name..` syntax to map stat variables to aesthetics.

```r
data + stat_density(binwidth = 1, origin = 10)
```

Colors and Fill Scales (discrete)

Different colors and fills are available for different discrete mappings.

```r
n + scale_fill_manual(values = c("blue", "red", "green"))
```

Coordinate Systems

Each coordinate system defines how x and y are transformed. The default is Cartesian, but new ones can be created.

```r
r + coord_fixed(ratio = 1/2)
```

Position Adjustments

Position adjustments determine how to arrange geoms that would otherwise occupy the same space.

```r
s + geom_bar(position = "dodge")
```

Faceting

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

```r
t <- ggplot(mpg, aes(cty, hwy)) + geom_point(
  facet_grid(fl ~ .)
)
```

Themes

Themes are a collection of settings that modify the overall appearance of a plot.

```r
r + theme_bw()
```

Labels

Labels are used to provide titles and legends for plots.

```r
n + labs(title = "Minimal theme")
```

Zooming

Without clipping, zooming with `zoom` is preferred.

```r
t + coord_cartesian(xlim = c(0, 100), ylim = c(0, 100))
```

Legends

Legends provide a visual representation of the mappings used in a plot.

```r
n + scale_fill_discrete(name = "Legends")
```

Scales

Scales map data values to the visual values of an aesthetic. To change a mapping, add a new scale.

```r
n + scale_fill_manual(values = c("blue", "red", "green"))
```

GENERAL PURPOSE SCALES

Use with most aesthetics

```r
scale_*_continuous() - map cont. values to visual ones
```

COLOR AND FILL SCALES (DISCRETE)

Let scales map discrete values to visually distinct colors.

```r
n + scale_fill_manual(values = c("blue", "red", "green"))
```

Faceting

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

```r
t + facet_grid(ratio = 1/2)
```

Facet into both rows and columns with

```r
r + facet_wrap(~ fl)
```

Wrap facets into a rectangular layout with

```r
r + facet_wrap(~ fl)
```

Set scales to let axis limits vary across facets

```r
t + facet_grid(dv ~ fl, scales = "free")
```

Position labelers to adjust facet labels

```r
n + guides(fill = "free")
```

Position labels to adjust facet labels

```r
n + guides(fill = "free")
```

Legend place arguments

```r
n + labs(title = "Minimal theme")
```

Legend title and labels

```r
n + scale_fill_manual(values = c("blue", "red", "green"))
```

Minimal themes

```r
r + theme_linedraw()
```

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

```r
n + scale_fill_discrete(name = "Legends")
```

Legend place arguments

```r
n + labs(title = "Minimal theme")
```

Legend title and labels

```r
n + scale_fill_manual(values = c("blue", "red", "green"))
```

Themes

Themes are a collection of settings that modify the overall appearance of a plot.

```r
r + theme_bw()
```

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

```r
n + scale_fill_manual(values = c("blue", "red", "green"))
```

Legend place arguments

```r
n + labs(title = "Minimal theme")
```

Legend title and labels

```r
n + scale_fill_manual(values = c("blue", "red", "green"))
```