**Data Visualization**

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

**with ggplot2**

Cheat Sheet

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  | **Geoms** |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

**One Variable**

**Continuous**

a <- ggplot(mpg, aes(hwy)) **a + geom\_area(stat = "bin")**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

**Continuous X, Continuous Y** f <- ggplot(mpg, aes(cty, hwy))

**f + geom\_blank()**

**Two Variables**

**Continuous Bivariate Distribution**

i <- ggplot(movies, aes(year, rating))

**i + geom\_bin2d(**binwidth = c(5, 0.5)**)**

**Data Visualization**

**- Use a geom to represent data points, use the geom’s aesthetic properties to represent variables**

x, y, alpha, color, fill, linetype, size

b + geom\_area(aes(y = ..density..), stat = "bin")

xmax, xmin, ymax, ymin, alpha, color, fill, linetype, size, weight

**with ggplot2**

**~~One Varia~~ble**

**Two Variables**

**f + geom\_jitter()**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

**i + geom\_density2d()**

**Data Visualization**

|  |  |  |  |  |  | **Geoms** |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | **Continuous** |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  | **One Variable** |  |  |  |  |
|  |  |  |  |  |  |  |  |

**a + geom\_density(**kernel = "gaussian"**)**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

**Continuous X, Continuous Y**

**Continuous Bivariate Distribution**

Cheat Sheet

**- Use a geom to represent data points, use the geom’s aesthetic properties to represent variables** x, y, alpha, color, fill, shape, size

x, y, alpha, colour, linetype, size

**Basics**

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

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

h <- ggplot(movies, aes(year, rating))

x, y, alpha, color, fill, linetype, size, weight

**h + geom\_bin2d(**binwidth = c(5, 0.5)**)**

**with ggplot2**

**a + geom\_area(stat = "bin")**

**f + geom\_blank()**

**Two ~~Variables~~**

**i + geom\_hex()**

**Continuous**

b + geom\_density(aes(y = ..county..)) **Continuous X, Continuous Y**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

xmax, xmin, ymax, ymin, alpha, color, fill, **f + geom\_point()**

**ggplot2** is based on the **grammar of graphics**, the

Cheat Sheet

idea that you can build every graph from the same **Basics** 

few components: a **data** set, a set of **geoms**—visual

x, y, alpha, color, fill, linetype, size

**a + geom\_dotplot()**

b + geom\_area(aes(y = ..density..), stat = "bin")

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

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |
| **geom\_density(** |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

**a +** kernal = "gaussian"**)**

x, y, alpha, color, fill

**a + geom\_area(stat = "bin")**

x, y, alpha, color, fill, linetype, size, weight

x, y, alpha, color, fill, linetype, size

b + geom\_density(aes(y = ..county..))

f <- ggplot(mpg, aes(cty, hwy)) **f + geom\_jitter()**

**f + geom\_blank()**

x, y, alpha, color, fill, shape, size

**Continuous Bivariate Distribution** linetype, size, weight

h <- ggplot(movies, aes(year, rating))

x, y, alpha, color, fill, shape, size

**h + geom\_density2d()**

**h + geom\_bin2d(**binwidth = c(5, 0.5)**)**

x, y, alpha, colour, linetype, size

xmax, xmin, ymax, ymin, alpha, color, fill,

linetype, size, weight

**h + geom\_hex()**

x, y, alpha, colour, fill size **Continuous Function**

marks that represent data points, and a **coordinate system.**

b + geom~~\_area(aes(y =~~ ..density..), stat = "bin") **a+ geom\_dotplot()**

**a +** kernal = "gaussian"**)**

| **geom\_density(** |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| x, y, alpha, color, fill |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
| **geom\_freqpoly()geom\_dotplot()** |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

**f + geom\_point()**

**f + geom\_jitter()**

x, y, alpha, color, fill, shape, size

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

**f + geom\_quantile()**

**h + geom\_density2d()**

x, y, alpha, colour, fill size

x, y, alpha, color, linetype, size, weight

j <- ggplot(economics, aes(date, unemploy))

**Basics**

4

4

**a + geom\_freqpoly()**

x, y, alpha, color, fill, shape, size

x, y, alpha, colour, linetype, size

|  | F M A |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  | F M A |  |
|  |  |  |
|  |  |  |
|  | **data** |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

3

3

x, y, alpha, color, fill, linetype, size, weight

**f + geom\_quantile()**

**Continuous Function**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

**j + geom\_area()**

x, y, alpha, color, linetype, size

b + geom\_density(aes(y = ..county..))

**f + geom\_point()**

**h + geom\_hex()**

g <- ggplot(economics, aes(date, unemploy))

+

2

|  |  |
| --- | --- |
|  |  |

=

2

|  |  |
| --- | --- |
|  |  |

**a +**

**a+**

x, y, alph~~a, color, linet~~ype, size

x, y, alpha, color, linetype, size, weight x, y, alpha, color, fill, shape, size

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

**f + geom\_rug(**sides = "bl"**)**

x, y, alpha, colour, fill size

**g + geom\_area()**

x, y, alpha, color, fill, linetype, size

1

1

x, y, alpha, color, fill

b + geom\_freqpoly(aes(y = ..density..)) **f + geom\_rug(**sides = "bl"**)**

x, y, alpha, color, fill, linetype, size

4

b + geom\_freqpoly(aes(y = ..density..))

4

**f + geom\_quantile()**

alpha, color, linetype, size **Continuous Function**

00 1 2 3 4

00 1 2 3 4

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **geom\_histogram(** |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
| **geom\_freqpoly()** |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

**a + geom\_histogram(**binwidth = 5**)** alpha, color, linetype, size

3

3

**geom ~~coordinate~~**

**a +** binwidth = 5**) a +**

x, y, alpha, color, linetype, size, weight

g <- ggplot(economics, aes(date, unemploy)) **g + geom\_line()**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

**j + geom\_line()**

x = F y = A

+

2

|  | **system** |
| --- | --- |
|  |  |

=

2

| **plot** |  |
| --- | --- |
|  |  |

x, y, alpha, color, fill, linetype, size, weight

x, y, alpha, color, fill, linetype, size, weight

**f + geom\_smooth(**model = lm**)**

x, y, alpha, color, linetype, size

**g + geom\_area()**

x, y, alpha, color, linetype, size

1

1

b + geom\_histogram(aes(y = ..density..))

b + geom\_freqpoly(aes(y = ..density..))

**f + geom\_rug(**sides = "bl"**)**

x, y, alpha, color, fill, linetype, size, weight

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

**f + geom\_smooth(**model = lm**)** x, y, alpha, color, fill, linetype, size

x, y, alpha, color, linetype, size

00 1 2 3 4

00 1 2 3 4

**~~Discret~~e**

b + geom\_histogram(aes(y = ..density..)) alpha, color, linetype, size

To display ~~data va~~lues, map variables in the data set **data geom coordinate**

**system plot**

4

**a + geom\_histogram(**binwidth = 5**)** a <- ggplot(mpg, aes(fl))

**f + geom\_text(**aes(label = cty)**)**

**g + geom\_step(**direction = "hv"**)**

x, y, alpha, color, fill, linetype, size, weight **g + geom\_line()**

x, y, alpha, color, linetype, size

x = F

4

x, y, alpha, color, fill, linetype, size, weight

**Discrete**

to aesthetic properties of the geom like **size**, **color**,

|  | F M A |  |
| --- | --- | --- |
|  |  |  |
|  | **y** l |  |
|  |  |  |
|  |  |  |
|  | F M A |  |
|  |  |  |
|  | **data** |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

y = A

3

3

b + geom\_histogram(aes(y = ..density..))

**f + geom\_smooth(**model = lm**)**

x, y, label, alpha, angle, color, family, fontface,

x, y, alpha, color, linetype, size

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

**j + geom\_step(**direction = "hv"**)**

and **x** and ocations.

|  |  |
| --- | --- |
|  |  |

|  |  |
| --- | --- |
|  |  |

**b + geom\_bar()**

hjust, lineheight, size, vjust

x, y, alpha, color, fill, linetype, size, weight

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

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  | **C** |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  | **A** |  | **B** |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

**f + geom\_text(**aes(label = cty)**) Visualizing error**

+=

2

2

1

1

4

x, alpha, color, fill, linetype, size, weight

**Discrete**

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

**f + geom\_text(**aes(label = cty)**)**

**g + geom\_step(**direction = "hv"**)**

df <- data.frame(grp = c("A", "B"), fit = 4:5, se = 1:2)

x, y, label, alpha, angle, color, family, fontface, x, y, alpha, color, linetype, size

x, y, alpha, color, linetype, size

4

00 1 2 3 4

00 1 2 3 4 3

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **geom\_bar()** |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

**b + geom\_bar()**

**Discrete X, Continuous Y**

x, y, label, alpha, angle, color, family, fontface,

e <- ggplot(df, aes(grp, fit, ymin = fit-se, ymax = fit+se)) hjust, lineheight, size, vjust

3

**geom ~~coordinate~~**

|  | **system** |
| --- | --- |
|  |  |

| **plot** |  |
| --- | --- |
|  |  |

**b +**

**Graphical Primitives**

g <- ggplot(mpg, aes(class, hwy)) hjust, lineheight, size, vjust

**e + geom\_crossbar(**fatten = 2**) Visualizing error**

**Visualizing error**

x = F y = A

+=

2

2

x, alpha, color, fill, linetype, size, weight

x, alpha, color, fill, linetype, size, weight

df <- data.frame(grp = c("A", "B"), fit = 4:5, se = 1:2)

color = F size = A

1

00 1 2 3 4

1

00 1 2 3 4

c <- ggplot(map, aes(long, lat)) **c + geom\_polygon(**aes(group = group)**)**

**g + geom\_bar(stat = "identity")** x, y, alpha, color, fill, linetype, size, weight

**Discrete X, Continuous Y**

x, y, ymax, ymin, alpha, color, fill, linetype,

size

e <- ggplot(df, aes(grp, fit, ymin = fit-se, ymax = fit+se)) **e + geom\_errorbar()**

df <- data.frame(grp = c("A", "B"), fit = 4:5, se = 1:2) k <- ggplot(df, aes(grp, fit, ymin = fit-se, ymax = fit+se))

**data geom coordinate**

**system plot** x = F

**Graphical Primitives** x, y, alpha, color, fill, linetype, size

g <- ggplot(mpg, aes(class, hwy)) **g + geom\_boxplot()**

**Discrete X, Continuous Y e + geom\_crossbar(**fatten = 2**)**

x, ymax, ymin, alpha, color, linetype, size,

y = A

**Graphical Primitives**

**g + geom\_bar(stat = "identity")**

g <- ggplot(mpg, aes(class, hwy)) width (also **geom\_errorbarh()**)

x, y, ymax, ymin, alpha, color, fill, linetype,

color = F

size = A

Build a graph with **qplot()** or **ggplot()**

c <- ggplot(map, aes(long, lat))

**c + geom\_polygon(**aes(group = group)**)** g <- ggplot(economics, aes(date, unemploy)) x, y, alpha, color, fill, linetype, size

lower, middle, upper, x, ymax, ymin, alpha, color, fill, linetype, shape, size, weight

x, y, alpha, color, fill, linetype, size, weight **g + geom\_dotplot(**binaxis = "y", **g + geom\_boxplot()**

stackdir = "center"**)**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

size

**e + geom\_linerange()**

**e + geom\_errorbar()**

x, ymin, ymax, alpha, color, linetype, size

**g + geom\_bar(stat = "identity")** x, ymax, ymin, alpha, color, linetype, size,

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

**k + geom\_crossbar(**fatten = 2**)** x, y, ymax, ymin, alpha, color, fill, linetype,

c <- ggplot(map, aes(long, lat))

width (also **geom\_errorbarh()**)

size

**aesthetic mappings**

**data geom**

**g + geom\_path(**lineend="butt", linejoin="round’, linemitre=1**)**

x, y, alpha, color, fill

lower, middle, upper, x, ymax, ymin, alpha, color, fill, linetype, shape, size, weight

**g + geom\_violin(**scale = "area"**)**

**e + geom\_pointrange()**

x, y, alpha, color, fill, linetype, size, weight **e + geom\_linerange()**

x, y, ymin, ymax, alpha, color, fill, linetype,

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

**k + geom\_errorbar()**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
| **geom\_ribbon(** |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
| ymax=unemploy + 900)**geom\_path(** |  |  |  |  | l | i |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

**c + geom\_polygon(**aes(group = group)**)**

shape, size

x, y, alpha, color, linetype, size

g <- ggplot(economics, aes(date, unemploy))

**qplot(**x = cty, y = hwy, color = cyl, data = mpg, geom = "point"**)**

**g + geom\_dotplot(**binaxis = "y", x, y, alpha, color, fill, linetype, size, weight

x, ymin, ymax, alpha, color, linetype, size

4

4

|  | F M A |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  | F M A |  |
|  |  |  |
|  | **data** |  |
|  |  |  |
|  | data = mpg, |  |
|  |  |  |
|  |  |  |
|  |  |  |

**g +** aes(ymin=unemploy - 900,

stackdir = "center"**)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**g + geom\_boxplot()**

x, ymax, ymin, alpha, color, linetype, size,

x, y, alpha, color, fill, linetype, size x, y, alpha, color, fill

**)**

**Maps**

**e + geom\_pointrange()**

3

**g +** neend="butt",

3

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

Creates a complete plot ~~with give~~n d~~ata, geom~~, and

x, ymax, ymin, alpha, color, fill, linetype, size

x, y, ymin, ymax, alpha, color, fill, linetype,

width (also **geom\_errorbarh()**)

|  |  |
| --- | --- |
|  |  |
| 2 | 3 |
|  | **coordinate system** |
|  |  |

+=

2

2

|  |  |
| --- | --- |
|  |  |
| 2 | 3 |
| **plot))** |  |
|  |  |

linejoin="round’, linemitre=1**)**

**g + geom\_violin(**scale = "area"**) Discrete X, Discrete Y**

lower, middle, upper, x, ymax, ymin, alpha, map <- map\_data("state")

mappings. Supplies many useful defaults.

x, y, alpha, color, linetype, size

shape, size

1

4

1

4

**g + geom\_ribbon(**aes(ymin=unemploy - 900,

x, y, alpha, color, fill, linetype, size, weight h <- ggplot(diamonds, aes(cut, color))

color, fill, linetype, shape, size, weight

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

**Maps**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

**k + geom\_linerange()**

00 1 4 3

00 1 4 3

d<- ggplot(seals, aes(x = long, y = lat)) ymax=unemploy + 900)**)**

**h + geom\_jitter()**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

**e + geom\_map(**aes(map\_id = state), map = map**) + g + geom\_dotplot(**binaxis = "y",

x, ymin, ymax, alpha, color, linetype, size

**geom**

d <- ggplot(economics, aes(date, unemploy))

data <- data.frame(murder = USArrests$Murder, state = tolower(rownames(USArrests))) **expand\_limits(**x = map$long, y = map$lat**)**

+=

2

2

x, ymax, ymin, alpha, color, fill, linetype, size

x, y, alpha, color, fill, shape, size

**ggplot(aes(**x ~~= cty, y =~~ hw~~y~~x = F y = A

**d + geom\_segment(**aes(

**Discrete X, Discrete Y**

map\_id, alpha, color, fill, linetype, size stackdir = "center"**)**

1

1

xend = long + delta\_long,

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

map <- map\_data("state")

Begins a plot that you finish by adding layers to. No

**d + geom\_path(**lineend="butt",

yend = lat + delta\_lat)**)**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  | **geom\_segment(** |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

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

x, y, alpha, color, fill

**e + geom\_map(**aes(map\_id = state), map = map**) +**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

**k + geom\_pointrange()**

00 1 2 3 4

00 1 2 3 4

d<- ggplot(seals, aes(x = long, y = lat))

**h + geom\_jitter()**

**Three ~~Variable~~s**

defaults, but provides more control than qplot().

x, xend, y, yend, alpha, color, linetype, size

**data geom coordinate**

**plot**

linejoin="round’, linemitre=1**)**

x, y, alpha, color, fill, shape, size

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  | **Three Variables** |  |
|  |  |  |  |  |  |  |  |

**expand\_limits(**x = map$long, y = map$lat**)**

**g + geom\_violin(**scale = "area"**)**

x, y, ymin, ymax, alpha, color, fill, linetype,

x = F **system** y = A

**d +** aes(

**d + geom\_rect(**aes(xmin = long, ymin = lat,

seals$z <- with(seals, sqrt(delta\_long^2 + delta\_lat^2)) i <- ggplot(seals, aes(long, lat))

**i + geom\_raster(**aes(fill = z), hjust=0.5, map\_id, alpha, color, fill, linetype, size

shape, size

**data**

**add layers,**

x, y, alpha, color, linetype, size

xend = long + delta\_long,

xmax= long + delta\_long,

yend = ~~lat + delta\_l~~at)**)**

ymax = lat + delta\_lat)**)**

**d + geom\_ribbon(**aes(ymin=unemploy - 900,

vjust=0.5, interpolate=FALSE**)**

x, y, alpha, color, fill, linetype, size, weight x, y, alpha, fill

ggplot(mpg, aes(hwy, cty)) +

**elements with +**

x, xend, y, yend, alpha, color, linetype, size

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  | **geom\_rect(** |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

xmax, xmin, ymax, ymin, alpha, color, fill,

**i + geom\_contour(**aes(z = z)**)**

seals$z <- with(seals, sqrt(delta\_long^2 + delta\_lat^2)) x, y, z, alpha, colour, linetype, size, weight

**i + geom\_tile(**aes(fill = z)**)**

**i + geom\_raster(**aes(fill = z), hjust=0.5, x, y, alpha, color, fill, linetype, size

**Maps**

linetype, size

ymax=unemploy + 900)**)**

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

geom\_point(aes(color = cyl)) +

**d +** aes(xmin = long, ymin = lat,

**layer = geom +**

xmax= long + delta\_long,

i <- ggplot(seals, aes(long, lat))

vjust=0.5, interpolate=FALSE**)**

geom\_smooth(method ="lm") +

**default stat +**

ymax = lat + delta\_lat)**)**

x, ymax, ymin, alpha, color, fill, linetype, size

x, y, alpha, fill

**Discrete X, Discrete Y**

RStudio® is a trademark of RStudio, Inc. • CC BY RStudio • info@rstudio.com • 844-448-1212 • rstudi~~o.com~~ Learn more at docs.ggplot2.org • ggplot2 0.9.3.1 • Updated: 3/15

coord\_cartesian() + scale\_color\_gradient() +

**~~layer sp~~ecific mappings**

xmax, xmin, ymax, ymin, alpha, color, fill, linetype, size

**i + geom\_contour(**aes(z = z)**)**

x, y, z, alpha, colour, linetype, size, weight

**i + geom\_tile(**aes(fill = z)**)**

h <- ggplot(diamonds, aes(cut, color)) x, y, alpha, color, fill, linetype, size

map <- map\_data("state") l <- ggplot(data, aes(fill = murder))

theme\_bw()

**additional**

e <- ggplot(seals, aes(x = long, y = lat))

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

**h + geom\_jitter()**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

**l + geom\_map(**aes(map\_id = state), map = map**) +**

RStudio® is a trademark of RStudio, Inc. • CC BY RStudio • info@rstudio.com • 844-448-1212 • rstudio.com Learn more at docs.ggplot2.org • ggplot2 0.9.3.1 • Updated: 3/15 **~~eleme~~nts**

**expand\_limits(**x = map$long, y = map$lat**)**

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.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

**e + geom\_segment(**aes(

xend = long + delta\_long,

yend = lat + delta\_lat)**)**

x, xend, y, yend, alpha, color, linetype, size

x, y, alpha, color, fill, shape, size

**Three Variables**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

map\_id, alpha, color, fill, linetype, size **m + geom\_raster(**aes(fill = z), hjust=0.5,

**last\_plot()**

Returns the last plot

**ggsave("plot.png", width = 5, height = 5)**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

**e + geom\_rect(**aes(xmin = long, ymin = lat, xmax= long + delta\_long,

ymax = lat + delta\_lat)**)**

seals$z <- with(seals, sqrt(delta\_long^2 + delta\_lat^2)) m <- ggplot(seals, aes(long, lat))

vjust=0.5, interpolate=FALSE**)** x, y, alpha, fill

Saves last plot as 5’ x 5’ file named "plot.png" in working directory. Matches file type to file extension.

xmax, xmin, ymax, ymin, alpha, color, fill, linetype, size

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

**m + geom\_contour(**aes(z = z)**)** x, y, z, alpha, colour, linetype, size, weight

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

**m + geom\_tile(**aes(fill = z)**)** x, y, alpha, color, fill, linetype, size

RStudio® is a trademark of RStudio, Inc. • CC BY RStudio • info@rstudio.com • 844-448-1212 • rstudio.com Learn more at docs.ggplot2.org • ggplot2 0.9.3.1 • Updated: 3/15

**Stats - An alternative way to build a layer Coordinate Systems Scales Faceting**

Some plots visualize a **transformation** of the original data set. Use a **stat** to choose a common transformation to visualize, e.g. a + geom\_bar(stat = "bin")

**Scales** control how a plot maps data values to the visual values of an aesthetic. To change the mapping, add a custom scale.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

r <- b + geom\_bar()

**r + coord\_cartesian(**xlim = c(0, 5)**)** xlim, ylim

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

t <- ggplot(mpg, aes(cty, hwy)) + geom\_point()

| fl |  | cty cyl |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

4

| x | ..count.. |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |

3 

|  |  |
| --- | --- |
|  |  |

4 3

|  |  |
| --- | --- |
|  |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

**n <- b + geom\_bar(**aes(fill = fl)**) n**

The default cartesian coordinate system

**t + facet\_grid(. ~ fl)**

**stat**

+=

2

1

00 1 2 3 4

2

1

00 1 2 3 4

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

**scale\_ aesthetic to adjust**

**prepackaged scale to use**

**scale specific arguments**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |

**r + coord\_fixed(**ratio = 1/2**)** ratio, xlim, ylim

facet into columns based on fl **t + facet\_grid(year ~ .)** facet into rows based on year

**data geom coordinate**

**system plot** x = x

y = ..count..

Each stat creates additional variables to map aesthetics to. These variables use a common **..name..** syntax.

stat functions and geom functions both combine a stat

**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")**)**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

Cartesian coordinates with fixed aspect ratio between x and y units

**r + coord\_flip()**

xlim, ylim

**t + facet\_grid(year ~ fl)**

facet into both rows and columns **t + facet\_wrap(~ fl)**

wrap facets into a rectangular layout

with a geom to make a layer, i.e. stat\_bin(geom="bar") does the same as geom\_bar(stat="bin")

**range of values to include in mapping**

**title to use in legend/axis**

**labels to use in legend/axis**

**breaks to use in legend/axis**

Flipped Cartesian coordinates

**r + coord\_polar(**theta = "x", direction=1 **)**

Set **scales** to let axis limits vary across facets

**stat functionlayer specific mappings**

**variable created by transformation**

**General Purpose scales** Use with any aesthetic:

theta, start, direction Polar coordinates

**t + facet\_grid(y ~ x, scales = "free")** x and y axis limits adjust to individual facets

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

**i + stat\_density2d(**aes(fill = ..level..), geom = "polygon", n = 100**)**

**geom for layer parameters for stat**

alpha, color, fill, linetype, shape, size

**scale\_\*\_continuous()** - map cont’ values to visual values **scale\_\*\_discrete()** - map discrete values to visual values **scale\_\*\_identity()** - use data values **as** visual values

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

**r + coord\_trans(**ytrans = "sqrt"**)** xtrans, ytrans, limx, limy

Transformed cartesian coordinates. Set extras and strains to the name

• **"free\_x"** - x axis limits adjust • **"free\_y"** - y axis limits adjust

Set **labeller** to adjust facet labels

**a + stat\_bin(**binwidth = 1, origin = 10**)**

1D distributions

of a window function.

60

**t + facet\_grid(. ~ fl, labeller = label\_both)**

x, y | ..count.., ..ncount.., ..density.., ..ndensity.. **a + stat\_bindot(**binwidth = 1, binaxis = "x"**)** x, y, | ..count.., ..ncount..

**a + stat\_density(**adjust = 1, kernel = "gaussian"**)**

**scale\_\*\_manual(**values = c()**)** - map discrete values to manually chosen visual values

**X and Y location scales**

|  |
| --- |
|  |

**z + coord\_map(**projection = "ortho", lat

orientation=c(41, -74, 0)**)**

projection, orientation, xlim, ylim

**fl: c fl: d fl: e fl: p fl: r**

**t + facet\_grid(. ~ fl, labeller = label\_bquote(**alpha ^ .(x**)))** *↵c ↵d ↵e ↵p ↵r*

x, y, | ..count.., ..density.., ..scaled..

Use with x or y aesthetics (x shown here)

long

Map projections from the mapproj package

**t + facet\_grid(. ~ fl, labeller = label\_parsed)**

**f + stat\_bin2d(**bins = 30, drop = TRUE**)** x, y, fill | ..count.., ..density..

**f + stat\_binhex(**bins = 30**)**

x, y, fill | ..count.., ..density..

**f + stat\_density2d(**contour = TRUE, n = 100**)** x, y, color, size | ..level..

2D distributions

**scale\_x\_date(**labels = date\_format("%m/%d"), breaks = date\_breaks("2 weeks")**)** - treat x

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

values as dates. See ?strptime for label formats.

**scale\_x\_datetime()** - treat x values as date times. Use same arguments as scale\_x\_date().

**scale\_x\_log10()** - Plot x on log10 scale

(mercator (default), azequalarea, lagrange, etc.)

**Position Adjustments**

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

**c d e p r**

**Labels**

**t + ggtitle(**"New Plot Title"**)**

Add a main title above the plot

**m + stat\_contour(**aes(z = z)**)**

3 Variables

**scale\_x\_reverse()** - Reverse direction of x axis

s <- ggplot(mpg, aes(fl, fill = drv))

**t + xlab(**"New X label"**)**

**Use scale functions**

x, y, z, order | ..level..

**m+ stat\_spoke(**aes(radius= z, angle = z)**)**

angle, radius, x, xend, y, yend | ..x.., ..xend.., ..y.., ..yend.. **m + stat\_summary\_hex(**aes(z = z), bins = 30, fun = mean**)**

**scale\_x\_sqrt()** - Plot x on square root scale **Color and fill scales**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

**s + geom\_bar(position = "dodge")** Arrange elements side by side

**s + geom\_bar(position = "fill")**

Change the label on the X axis **t + ylab(**"New Y label"**)** Change the label on the Y axis

**to update legend labels**

x, y, z, fill | ..value..

Discrete Continuous

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

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

**m + stat\_summary2d(**aes(z = z), bins = 30, fun = mean**)**

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**n <- b + geom\_bar(**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Stack elements on top of one another,

All of the above

x, y, z, fill | ..value..

aes(fill = fl)**)o <- a + geom\_dotplot(** aes(fill = ..x..)**)**

normalize height

**g + stat\_boxplot(**coef = 1.5**)**

x, y | ..lower.., ..middle.., ..upper.., ..outliers..

Comparisons

**n + scale\_fill\_brewer(** palette = "Blues"**)**

**o + scale\_fill\_gradient(** low = "red",

**s + geom\_bar(position = "stack")** Stack elements on top of one another

**Legends**

**t + theme(**legend.position = "bottom"**)**

**g + stat\_ydensity(**adjust = 1, kernel = "gaussian", scale = "area"**)** x, y | ..density.., ..scaled.., ..count.., ..n.., ..violinwidth.., ..width..

For palette choices: library(RcolorBrewer) display.brewer.all()

**n + scale\_fill\_grey(**

high = "yellow"**)**

**o + scale\_fill\_gradient2(** low = "red", hight = "blue", mid = "white", midpoint = 25**)**

**f + geom\_point(position = "jitter")** Add random noise to X and Y position

Place legend at "bottom", "top", "lef", or "right" **t + guides(**color = "none"**)**

**f + stat\_ecdf(**n = 40**)** x, y | ..x.., ..y..

Functions

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

start = 0.2, end = 0.8, na.value = "red"**)**

**o + scale\_fill\_gradientn(** colo**u**rs = terrain.colors(6)**)**

of each element to avoid overplotting

Set legend type for each aesthetic: colorbar, legend,

**f + stat\_quantile(**quantiles = c(0.25, 0.5, 0.75), formula = y ~ log(x), method = "rq"**)**

x, y | ..quantile.., ..x.., ..y..

Also: rainbow(), heat.colors(), topo.colors(), cm.colors(), RColorBrewer::brewer.pal()

Each position adjustment can be recast as a function with manual **width** and **height** arguments

or none (no legend)

**t + scale\_fill\_discrete(**name = "Title", labels = c("A", "B", "C")**)**

**f + stat\_smooth(**method = "auto", formula = y ~ x, se = TRUE, n = 80,

**Shape scales**

Manual Shape values

**s + geom\_bar(position = position\_dodge(width = 1))**

Set legend title and labels with a scale function.

fullrange = FALSE, level = 0.95**)**

Manual shape values

x, y | ..se.., ..x.., ..y.., ..ymin.., ..ymax.. **ggplot() + stat\_function(**aes(x = -3:3),

General Purpose

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**p <- f + geom\_point(**

~~0~~

~~18~~

~~6~~

~~12~~

~~24~~

aes(shape = fl)**)**

~~19~~

~~13~~

~~25~~

~~7~~

~~1~~

**p + scale\_shape(**

~~0~~ ~~0~~

+ ~~+~~~~-~~~~-~~~~|~~ ~~|~~

**Themes Zooming**

**Without clipping** (preferred)

fun = dnorm, n = 101, args = list(sd=0.5)**)**

x | ..y..

**f + stat\_identity()**

**ggplot() + stat\_qq(**aes(sample=1:100), distribution = qt, dparams = list(df=5)**)**

sample, x, y | ..x.., ..y..

~~2~~

solid = FALSE**)**

~~3~~

**p + scale\_shape\_manual(** ~~4~~

values = c(3:7)**)**

Shape values shown in

~~5~~

chart on right

~~8~~

~~9~~

~~10 11~~

~~14 15 16 17~~

~~20 21 22 23~~

~~\*~~~~\*~~.

o ~~o~~ ~~O~~~~O~~

~~%~~~~%~~ ~~#~~ ~~#~~

coun~~t~~ count

150 10050

0

150100 50

cdep rfl

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

150

**r + theme\_bw()** count

100

50

White background 0

with grid lines

**r + theme\_grey()**

150

count

100

Grey background 50

cdep rfl

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**r + theme\_classic()** White background no gridlines

**r + theme\_minimal()** Minimal theme

**t + coord\_cartesian(**

xlim = c(0, 100), ylim = c(10, 20)**)**

**With clipping** (removes unseen data points) **t + xlim(**0, 100**) + ylim(**10, 20**)**

**f + stat\_sum()**

x, y, size | ..size..

0

**Size scales**

cdep rfl

(default theme) 0

cdep rfl

**t + scale\_x\_continuous(**limits = c(0, 100)**) +**

**f + stat\_summary(**fun.data = "mean\_cl\_boot"**) f + stat\_unique()**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**q <- f + geom\_point(** aes(size = cyl)**)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**q + scale\_size\_area(**max = 6**)** Value mapped to area of circle (not radius)

**ggthemes -** Package with additional ggplot2 themes

**scale\_y\_continuous(**limits = c(0, 100)**)**

RStudio® is a trademark of RStudio, Inc. • CC BY RStudio • info@rstudio.com • 844-448-1212 • rstudio.com Learn more at docs.ggplot2.org • ggplot2 0.9.3.1 • Updated: 3/15