DATA VISUALIZATION WITH GGPLOT2

Statistics with Geoms

ggplot2, course 2

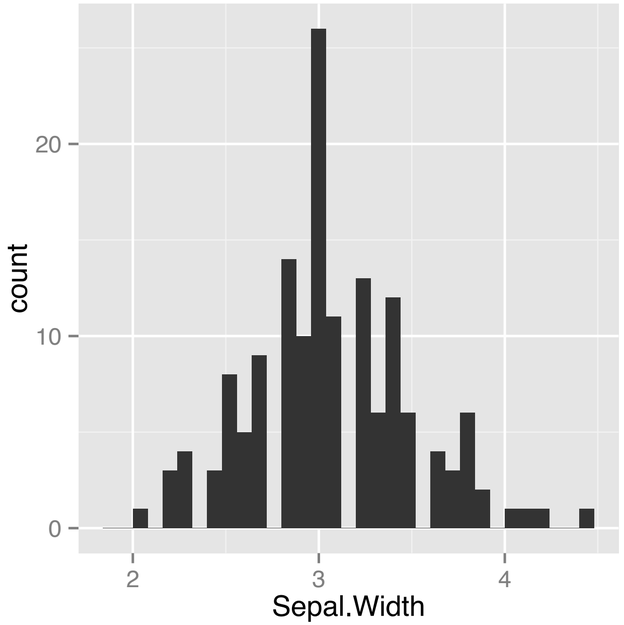
- Statistics
- Coordinates
- Facets
- Themes
- Data Visualization Best Practices
- Case Study: California Health Information Survey

Statistics Layer

- Two categories of functions
  - Called from within a geom
  - Called independently
- `stat_`

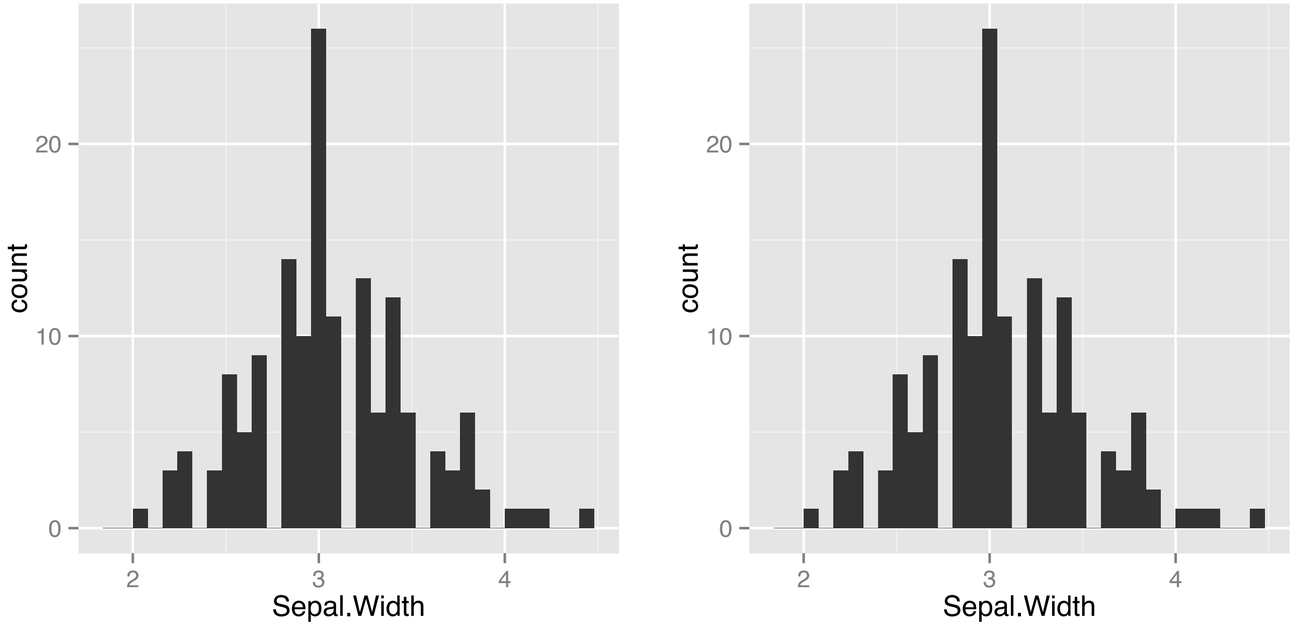
geom_ <-> stat_

```
> p <- ggplot(iris, aes(x = Sepal.Width))
> p + geom_histogram()
```



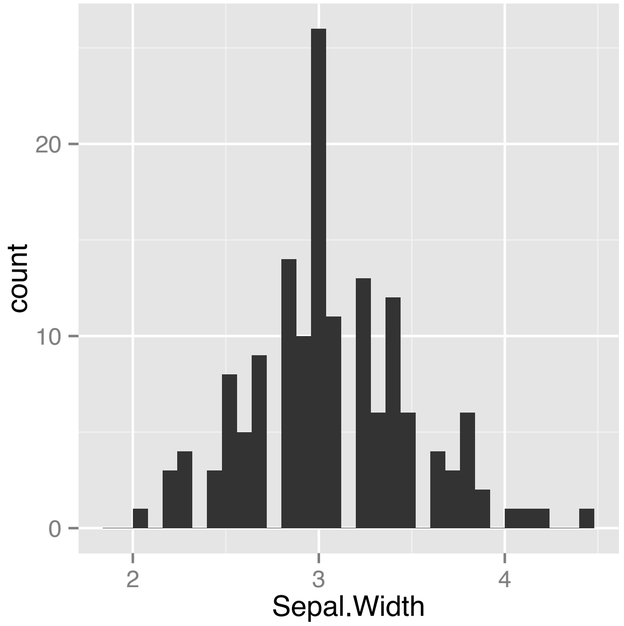
geom_ <-> stat_

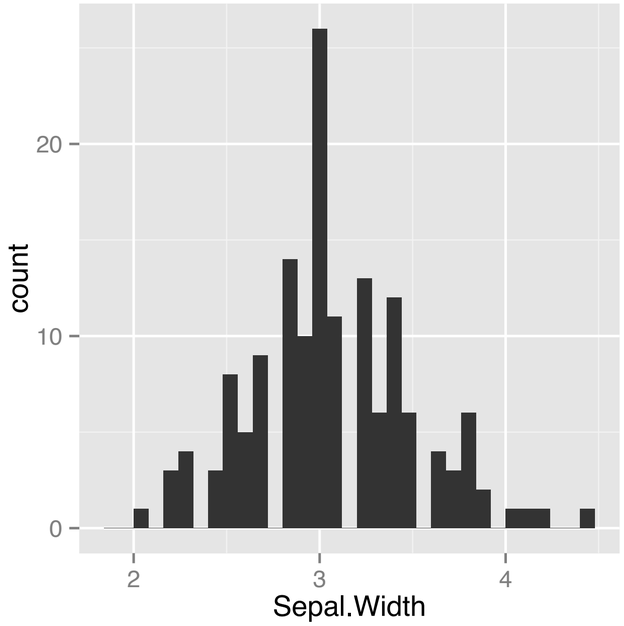
```
> p <- ggplot(iris, aes(x = Sepal.Width))
> p + geom_histogram()
> p + geom_bar()
```

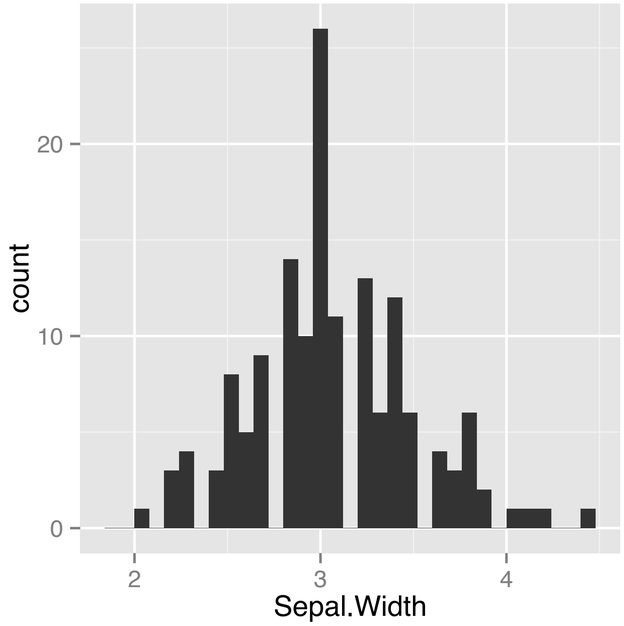


geom_ <-> stat_

```
> p <- ggplot(iris, aes(x = Sepal.Width))
> p + geom_histogram()
> p + geom_bar()
> p + stat_bin()
```

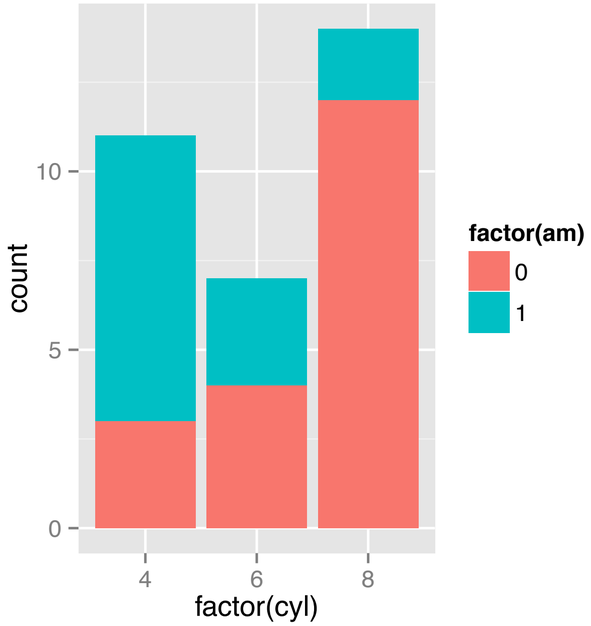


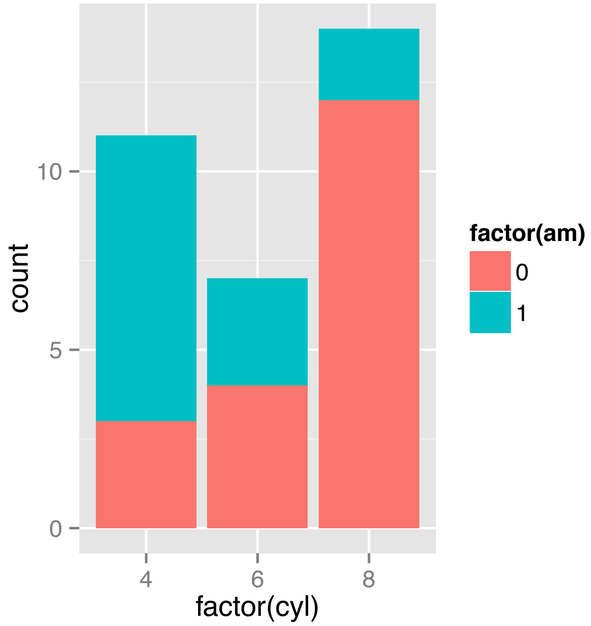




grouping by fill

```
> ggplot(mtcars, aes(x = factor(cyl), fill = factor(am))) +
    geom_bar()
> ggplot(mtcars, aes(x = factor(cyl), fill = factor(am))) +
    stat_bin()
```

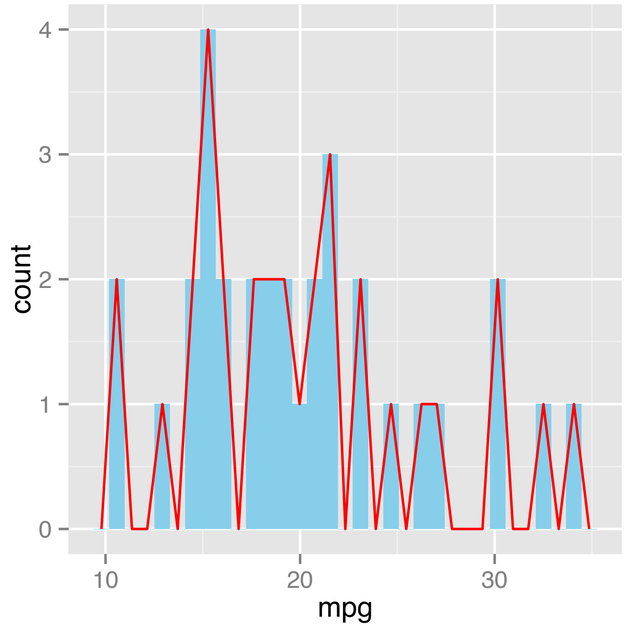




geom_ <-> stat_

stat_	geom_
stat_bin()	geom_histogram()
stat_bin()	geom_bar()
stat_bin()	geom_freqpoly()

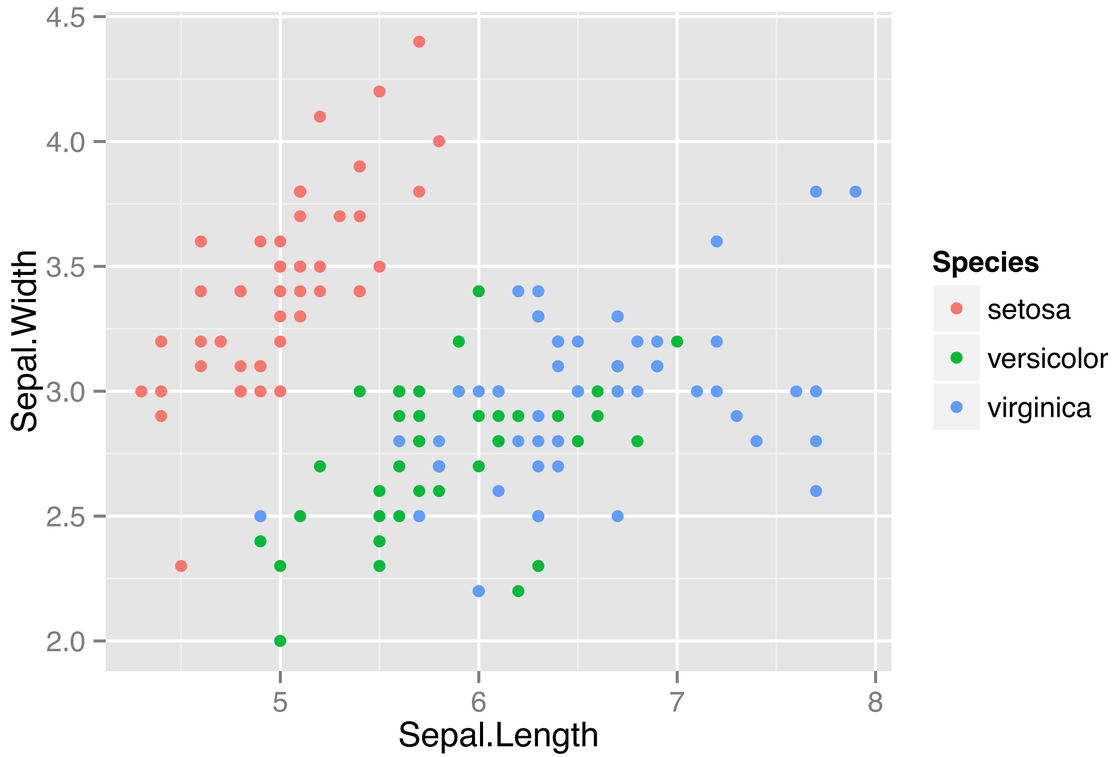
```
> ggplot(mtcars, aes(x = mpg)) +
    geom_histogram(fill = "skyblue") +
    geom_freqpoly(col = "red")
stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
```



smooth

stat_	geom_
stat_smooth()	geom_smooth()

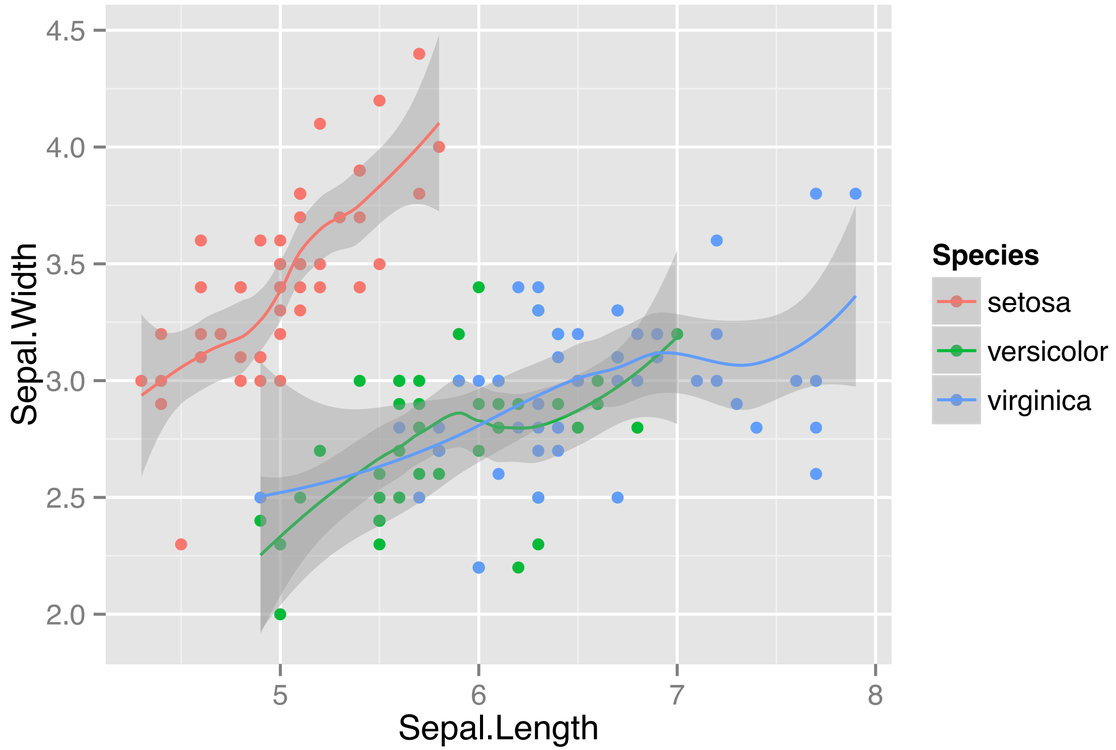
```
> ggplot(iris, aes(x = Sepal.Length, y = Sepal.Width, col = Species)) +
    geom_point()
```



smooth

stat_	geom_
stat_smooth()	geom_smooth()

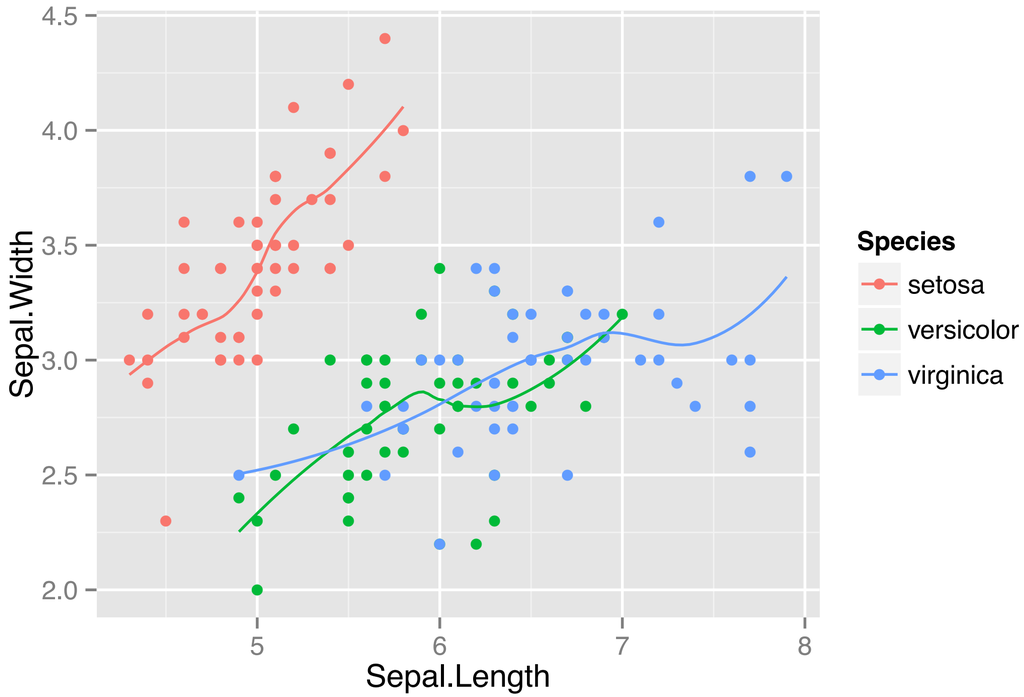
```
> ggplot(iris, aes(x = Sepal.Length, y = Sepal.Width, col = Species)) +
    geom_point() +
    geom_smooth()
```



smooth

stat_	geom_
stat_smooth()	geom_smooth()

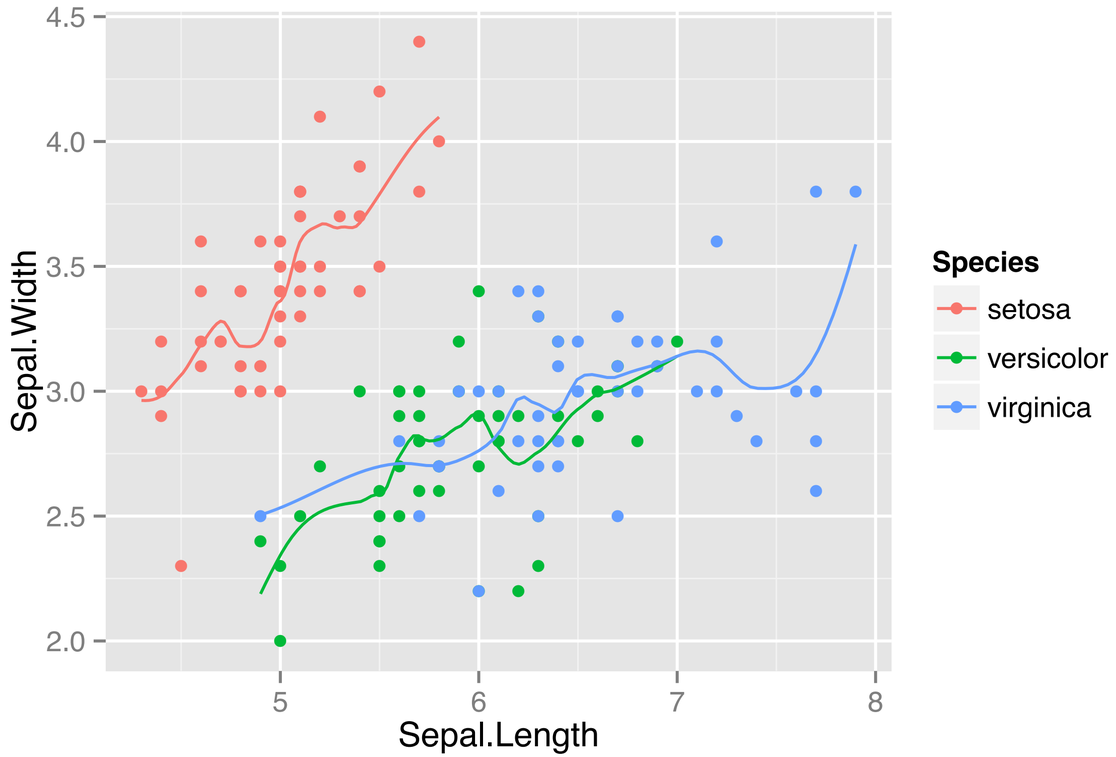
```
> ggplot(iris, aes(x = Sepal.Length, y = Sepal.Width, col = Species)) +
    geom_point() +
    geom_smooth(se = FALSE)
geom_smooth: method="auto" and size of largest group is <1000, so using
loess. Use 'method = x' to change the smoothing method.
```



smooth

stat_	geom_
stat_smooth()	geom_smooth()

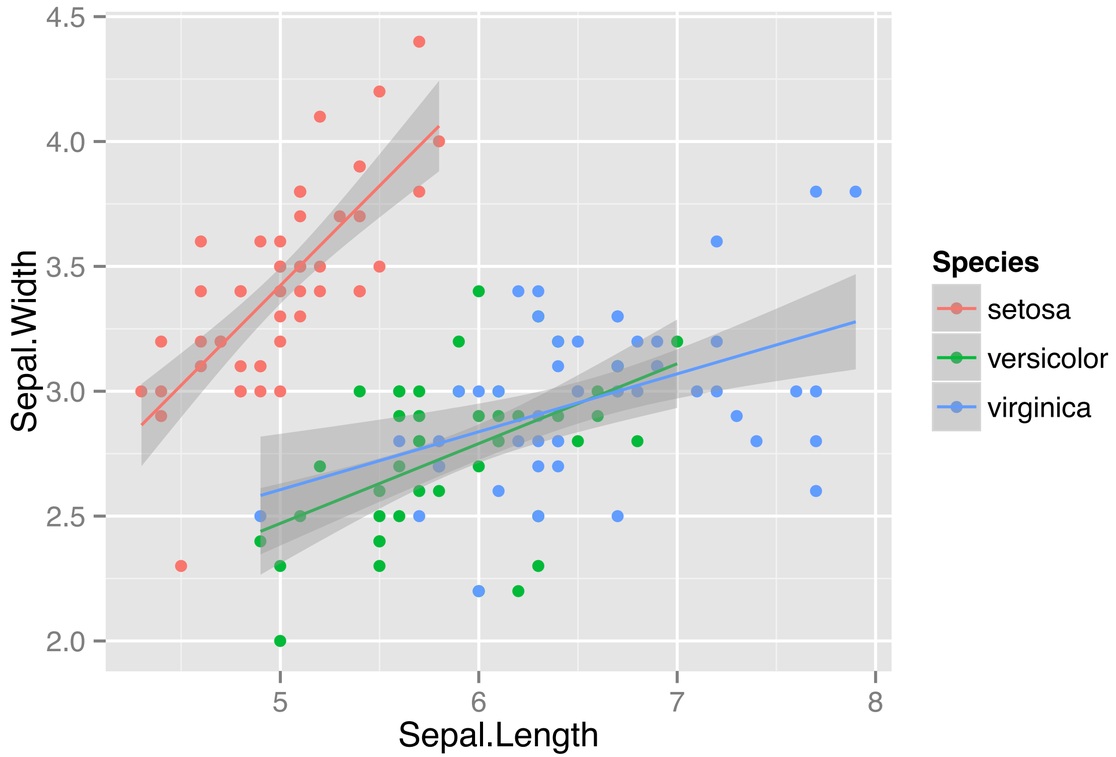
```
> ggplot(iris, aes(x = Sepal.Length, y = Sepal.Width, col = Species)) +
    geom_point() +
    geom_smooth(se = FALSE, span = 0.4)
```



smooth

stat_	geom_
stat_smooth()	geom_smooth()

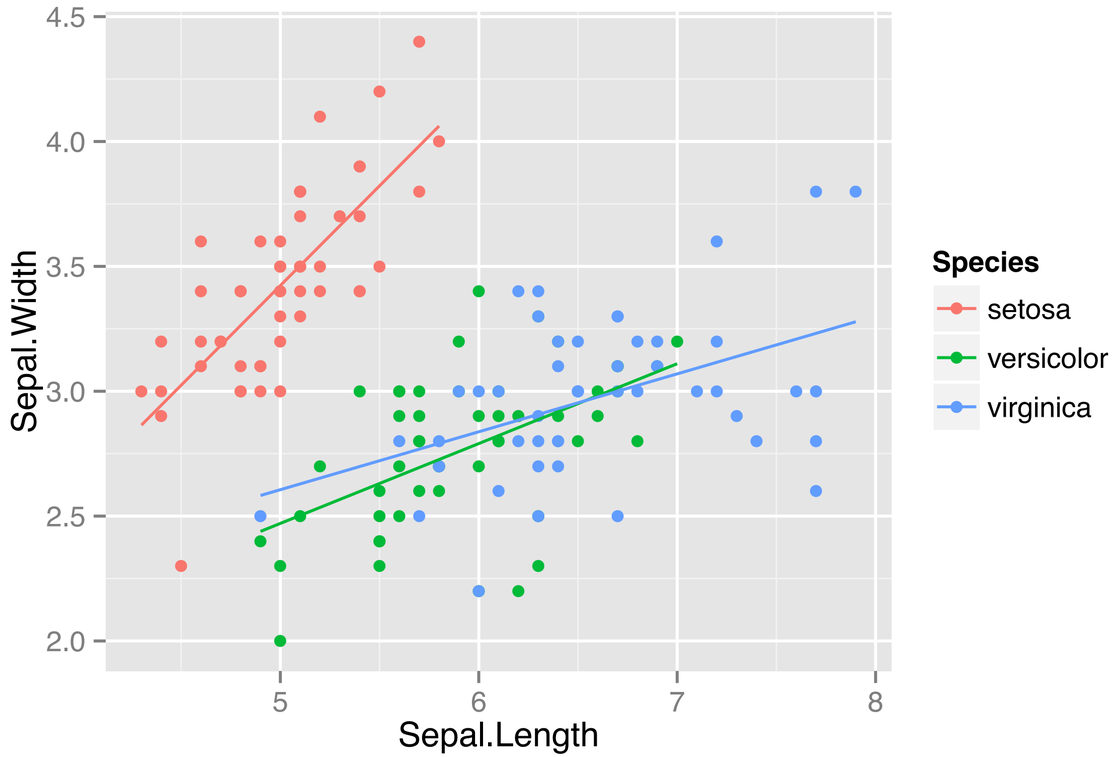
```
> ggplot(iris, aes(x = Sepal.Length, y = Sepal.Width, col = Species)) +
    geom_point() +
    geom_smooth(method = "lm")
```



smooth

stat_	geom_
stat_smooth()	geom_smooth()

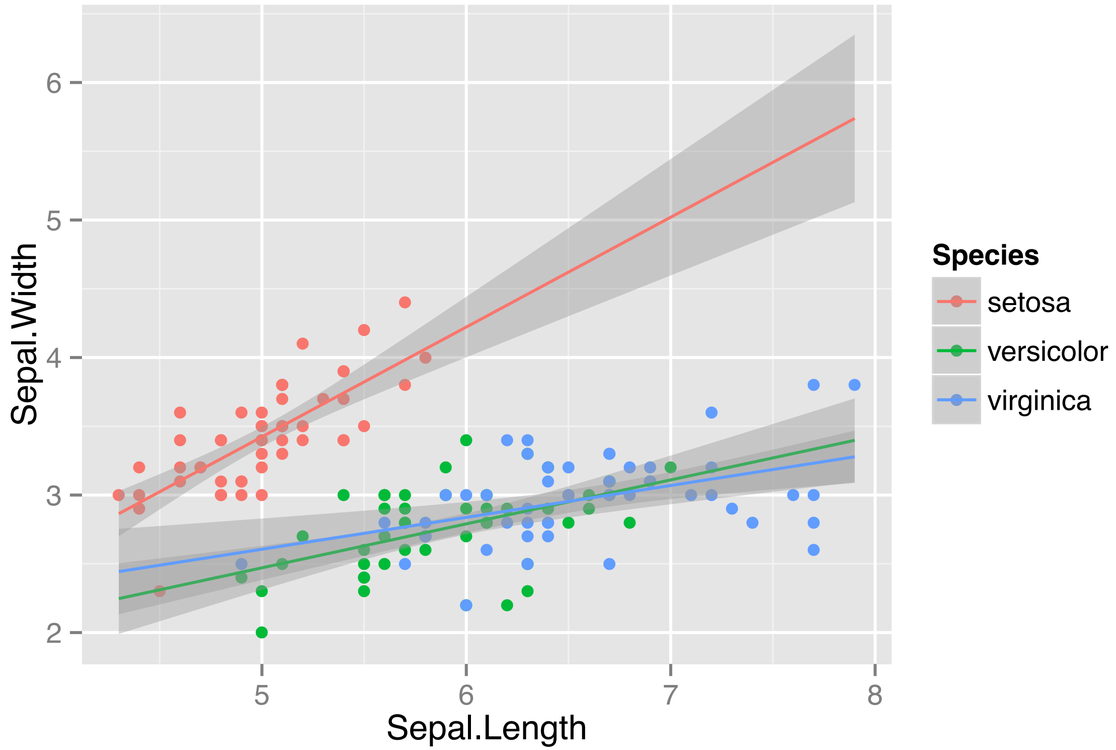
```
> ggplot(iris, aes(x = Sepal.Length, y = Sepal.Width, col = Species)) +
    geom_point() +
    geom_smooth(method = "lm", se = FALSE)
```



smooth

stat_	geom_
stat_smooth()	geom_smooth()

```
> ggplot(iris, aes(x = Sepal.Length, y = Sepal.Width, col = Species)) +
    geom_point() +
    geom_smooth(method = "lm", fullrange = TRUE)
```



Other stats_ functions

stat_	geom_
stat_bin()	geom_histogram()
stat_bin()	geom_bar()
stat_bin()	geom_freqpoly()
stat_smooth()	geom_smooth()
stat_boxplot()	geom_boxplot()
stat_bindot()	geom_dotplot()
stat_bin2d()	geom_bin2d()
stat_binhex()	geom_hex()
stat_contour()	geom_contour()
stat_quantile()	geom_quantile()
stat_sum()	geom_count()

DATA VISUALIZATION WITH GGPLOT2

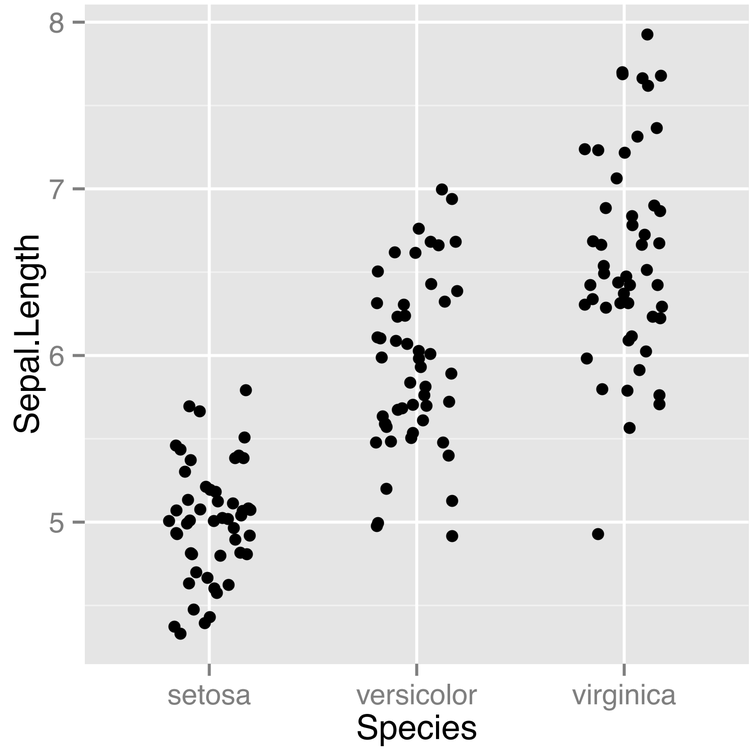
Let’s practice!

DATA VISUALIZATION WITH GGPLOT2

Statistics outside Geoms

Basic Plot

```
> ggplot(iris, aes(x = Species, y = Sepal.Length)) +
    geom_point(position = position_jitter(0.2))
```



Calculating Statistics

```
> set.seed(123)
> xx <- rnorm(100)

> mean(xx)
[1] 0.09040591

> mean(xx) + (sd(xx)* c(-1, 1))
[1] -0.822410  1.003222

> library(Hmisc)
> smean.sdl(xx, mult = 1)
      Mean        Lower        Upper
 0.09040591 -0.82240997  1.00322179
```

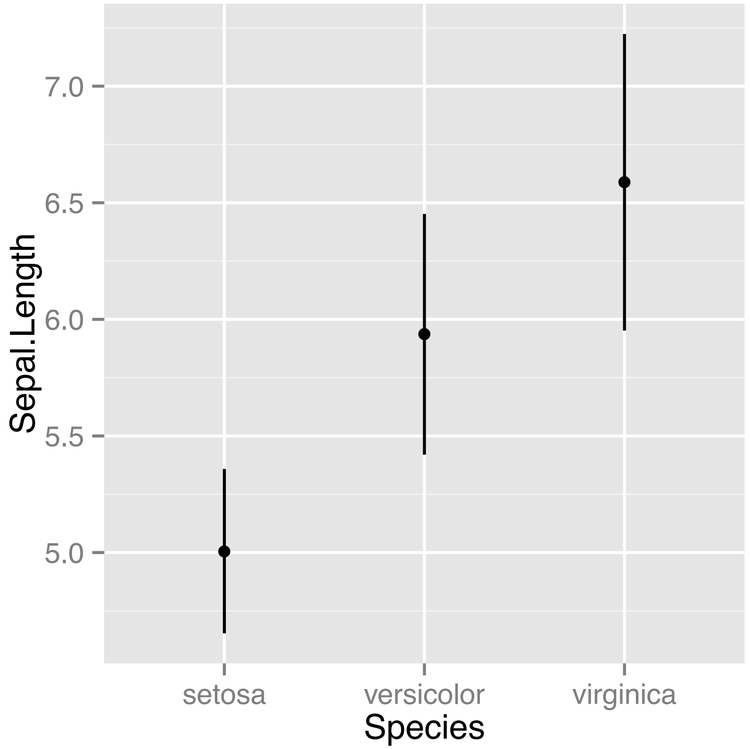
Calculating Statistics

```
> # Hmisc
> smean.sdl(xx, mult = 1)
       Mean       Lower       Upper
 0.09040591 -0.82240997  1.00322179

> # ggplot2
> mean_sdl(xx, mult = 1)
           y     ymin     ymax
1 0.09040591 -0.82241 1.003222
```

stat_summary()

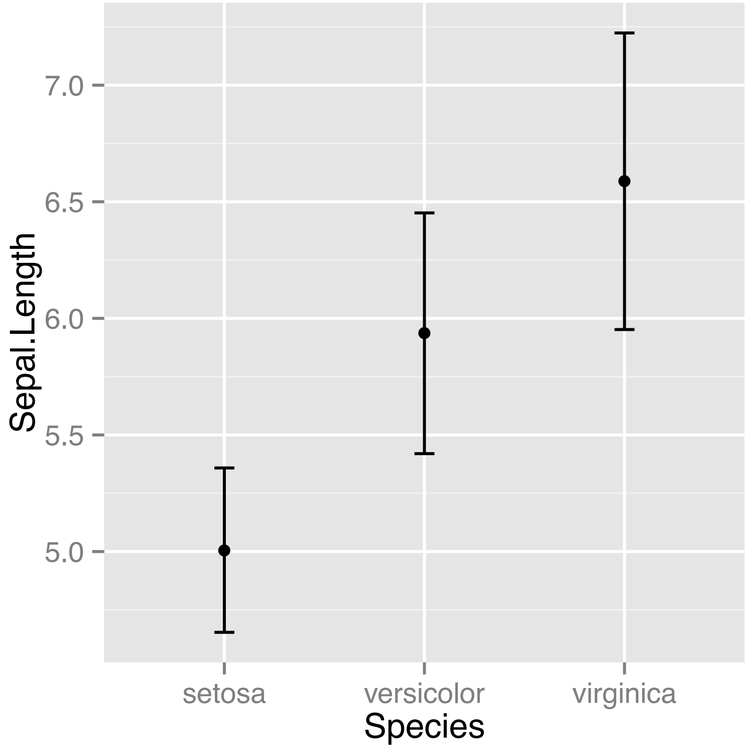
```
> ggplot(iris, aes(x = Species, y = Sepal.Length)) +
    stat_summary(fun.data = mean_sdl, fun.args = list(mult = 1))
```

uses geom_pointrange() by default



stat_summary()

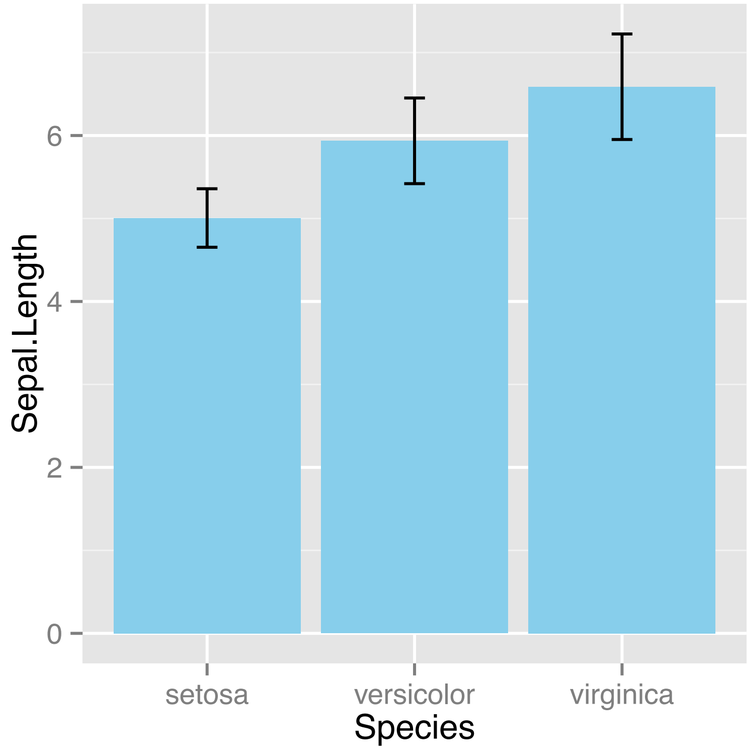
```
> ggplot(iris, aes(x = Species, y = Sepal.Length)) +
    stat_summary(fun.y = mean, geom = "point") +
    stat_summary(fun.data = mean_sdl, fun.args = list(mult = 1),
                 geom = "errorbar", width = 0.1)
```



stat_summary()

```
> ggplot(iris, aes(x = Species, y = Sepal.Length)) +
    stat_summary(fun.y = mean, geom = "bar", fill = "skyblue") +
    stat_summary(fun.data = mean_sdl, fun.args = list(mult = 1),
                 geom = "errorbar", width = 0.1)
```

NOT RECOMMENDED!



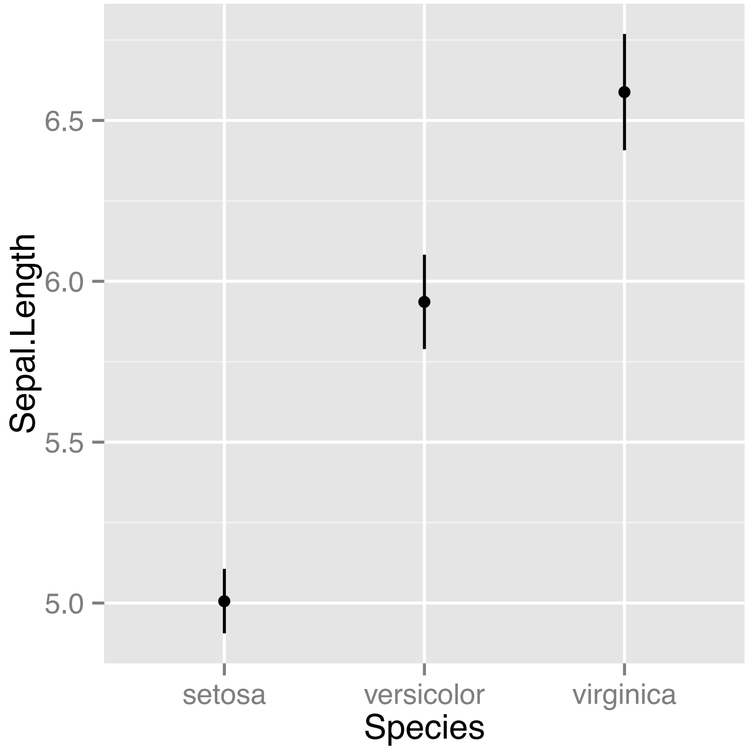
95% Confidence Interval

```
> ERR <- qt(0.975, length(xx) - 1) * (sd(xx) / sqrt(length(xx)))

> mean(xx) + (ERR* c(-1, 1)
[1] -0.09071657  0.27152838

> # Hmisc
> smean.cl.normal(xx)
       Mean        Lower        Upper
 0.09040591 -0.09071657  0.27152838

> # ggplot2
> mean_cl_normal(xx)
           y        ymin      ymax
1 0.09040591 -0.09071657 0.2715284
```

stat_summary()

```
> ggplot(iris, aes(x = Species, y = Sepal.Length)) +
    stat_summary(fun.data = mean_cl_normal, width = 0.1)
```

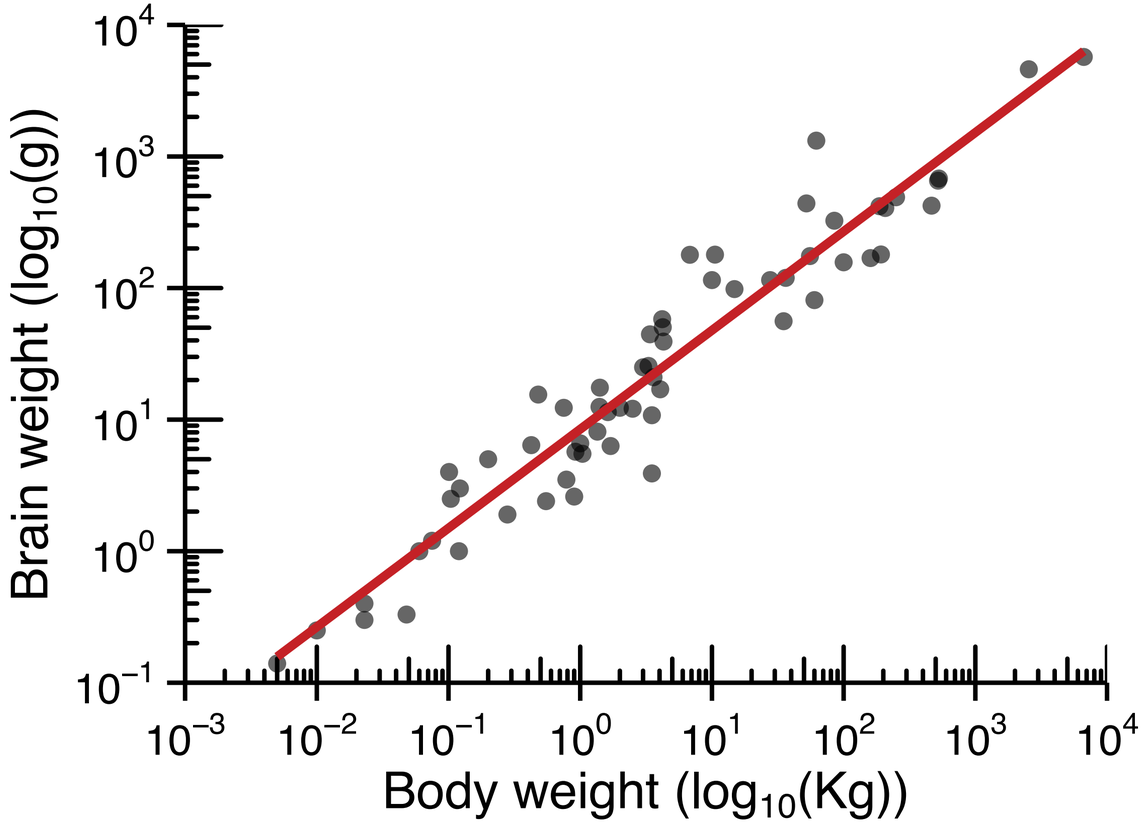
use any function, as long as output has expected format



Other stat_ functions

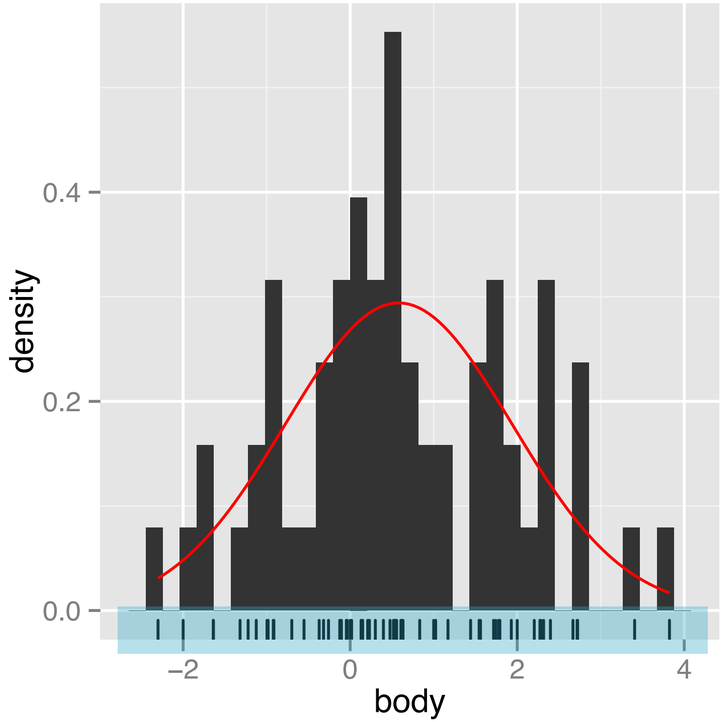
stat_	description
stat_summary()	Summarise y values at distinct x values
stat_function()	Compute y values from a function of x values
stat_qq()	Perform calculations for a quantile-quantile plot

MASS::mammals



Normal distribution

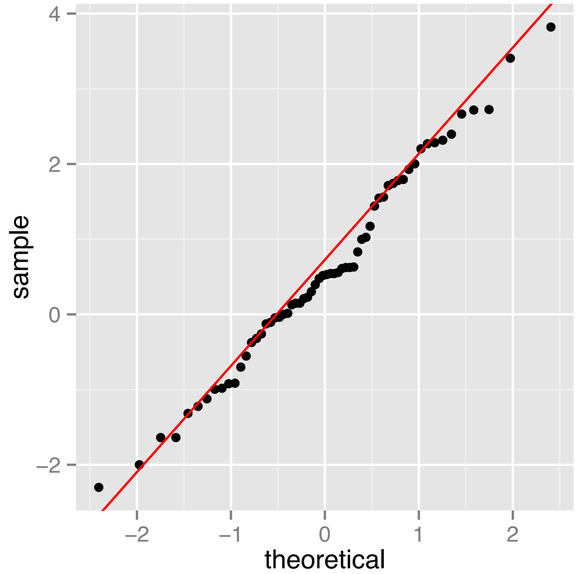
```
> library(MASS)
> mam.new <- data.frame(body = log10(mammals$body))
> ggplot(mam.new, aes(x = body)) +
    geom_histogram(aes( y = ..density..)) +
    geom_rug() +
    stat_function(fun = dnorm, colour = "red",
                  arg = list(mean = mean(mam.new$body),
                             sd = sd(mam.new$body)))
```



QQ plot

```
> mam.new$slope <- diff(quantile(mam.new$body, c(0.25, 0.75))) /
                           diff(qnorm(c(0.25, 0.75)))

> mam.new$int <- quantile(mam.new$body, 0.25) -
                      mam.new$slope * qnorm(0.25)

> ggplot(mam.new, aes(sample = body)) +
    stat_qq() +
    geom_abline(aes(slope = slope, intercept = int), col = "red")
```

manual calculations



DATA VISUALIZATION WITH GGPLOT2

Let’s practice!