ggplot2

Una implementación de la gramática de las gráficas en R

Una breve introducción

Ernesto Barrios

Instituto Tecnológico Autónomo de México

XXV Foro Nacional de Estadística

Cuernavaca, Mor.

Septiembre 2010

Contenido

Gramática

- La gramática le proporciona reglas al lenguaje.
- La *gramática de Chomsky* (50's) es un sistema formal de reglas para generar declaraciones correctas en el lenguaje.
- Libre de un contexto específico, la *gramática de Chomsky* es el origen de los analizadores de sintaxis (*parser*) de los lenguajes modernos de computación.
- El lenguaje, con palabras pero no gramática (declaración = palabra) expresa solamente tantas ideas como palabras tenga.

La Gramática de las Gráficas

Leland Wilkinson

- Trabaja para *Systat* en los 80's.
- En los 90's trabaja en técnicas orientada a objetos, viendo éstos como árboles. Las gráficas se "montan" en árboles.
- A finales de los 90's trabaja en GPL (graphics production library) en Java con Dan Rope y Dan Carr. Desarrollan las componentes de las gráficas.
- Al rededor de 2005 se une a *SPSS*.
- En 2005 sale la segunda edición de su libro: *The Grammar of Graphics*, donde distingue la sintaxis y la semántica de las gráficas.

La gramática de las gráficas no tiene que ver necesariamente con *"gráficas bonitas"*. Para eso, vea por ejemplo, los trabajos, de Tufte, Cleveland, Tukey, etc.

Paquete ggplot2 de Hadley Wickham

Una implementación de la gramática de las gráficas

- Se basa en la *gramática de las gráficas* de Wilkinson. Consiste de un conjunto de elementos independientes que se pueden componer de varias maneras, en distintas formas.
- Las gráficas pueden construirse iterativamente y editarse después.
- Permite enfocarse en construir gráficas que revelen lo que se desea comunicar mas que en "hacerlas bonitas".
- ggplot2 está diseñado para trabajar en capas: comenzado con la capa que muestre los datos y añadiendo después capas con resúmenes estadísticos y anotaciones.
- La gramática y ggplot2 no sugiere que tipo o clase de gráfica es adecuada para sus datos. Para esto vea Chambers et al., Cleveland, Tukey, Tufte, etc.
- ggplot2 no describe interacción de gráficas o gráficas dinámicas como Ggobi.

ggplot2 y otros paquetes

- El paquete `base` implementa la graficación básica. Fue desarrollada por Ross Ihaka al implementar el *driver* de las gráficas de *S* y siguiendo las ideas de Chambers et al. La gráficas básicas son como dibujar con pluma y papel. No es posible eliminar o borrar, solamente encimar.
- El paquete `grid` fue desarrollado por Paul Murrell. Define los *grid grobs = objetos gráficos*. Permite modificar gráficas.
- El paquete `lattice` de Deepayan Sarkar implementa *Trellis* de Cleveland. Puede producir fácilmente gráficas condicionales. Provee de detalles pero adolece de un modelo formal (gramática) lo que dificulta su extensión.
- `ggplot` comienza en 2005, considerando lo bueno de los paquetes `base` y `lattice` y desarrolla una base respaldada en un modelo para la producción de gráficas estadísticas. Componentes independientes facilitan su extensión. Se basa también en `grid` lo que permite control a bajo nivel.

Componentes de ggplot2

1. Los datos (***data***) que desea visualizar y el conjunto de mapeos estéticos (***mapping***) que describe como las variables de los datos son mapeados a atributos estéticos.
2. Los objetos geométricos (***geom***) que representan lo que usted ve en realidad en la gráfica. Por ejemplo, puntos, líneas, polígonos, etc.
3. Transformaciones estadísticas (***stat***), que resumen los datos en distintas formas. Por ejemplo, conteos e intervalos en los histogramas; una relación 2D co un modelo lineal. stat es opcional pero muy útiles.

Componentes de ggplot2

1. Los datos (***data***) que desea visualizar y el conjunto de mapeos estéticos (***mapping***) que describe como las variables de los datos son mapeados a atributos estéticos.
2. Los objetos geométricos (***geom***) que representan lo que usted ve en realidad en la gráfica. Por ejemplo, puntos, líneas, polígonos, etc.
3. Transformaciones estadísticas (***stat***), que resumen los datos en distintas formas. Por ejemplo, conteos e intervalos en los histogramas; una relación 2D co un modelo lineal. stat es opcional pero muy útiles.

Componentes de ggplot2

1. Los datos (***data***) que desea visualizar y el conjunto de mapeos estéticos (***mapping***) que describe como las variables de los datos son mapeados a atributos estéticos.
2. Los objetos geométricos (***geom***) que representan lo que usted ve en realidad en la gráfica. Por ejemplo, puntos, líneas, polígonos, etc.
3. Transformaciones estadísticas (***stat***), que resumen los datos en distintas formas. Por ejemplo, conteos e intervalos en los histogramas; una relación 2D co un modelo lineal. stat es opcional pero muy útiles.

Componentes de ggplot2

1. Las escalas (***scale***) del mapeo en el espacio de datos al espacio de atributos estéticos, como color, tamaño o forma. Provee de información al margen que permite el mapeo inverso.
2. Un sistema coordenado (***coord***) que describe como las coordenadas de los datos son mapeados al plano de la gráfica. Usualmente son las coordenadas cartecianas pero también están las polares y algunas proyecciones.
3. La presentación (***facet***) que describe como partir o separar los datos en subconjuntos y como mostrar esos subconjuntos en pequeños múltiplos. Conocido en otros paquetes como *enrrejado* (*trellis*, *lattice*).

Componentes de ggplot2

1. Las escalas (***scale***) del mapeo en el espacio de datos al espacio de atributos estéticos, como color, tamaño o forma. Provee de información al margen que permite el mapeo inverso.
2. Un sistema coordenado (***coord***) que describe como las coordenadas de los datos son mapeados al plano de la gráfica. Usualmente son las coordenadas cartecianas pero también están las polares y algunas proyecciones.
3. La presentación (***facet***) que describe como partir o separar los datos en subconjuntos y como mostrar esos subconjuntos en pequeños múltiplos. Conocido en otros paquetes como *enrrejado* (*trellis*, *lattice*).

Componentes de ggplot2

1. Las escalas (***scale***) del mapeo en el espacio de datos al espacio de atributos estéticos, como color, tamaño o forma. Provee de información al margen que permite el mapeo inverso.
2. Un sistema coordenado (***coord***) que describe como las coordenadas de los datos son mapeados al plano de la gráfica. Usualmente son las coordenadas cartecianas pero también están las polares y algunas proyecciones.
3. La presentación (***facet***) que describe como partir o separar los datos en subconjuntos y como mostrar esos subconjuntos en pequeños múltiplos. Conocido en otros paquetes como *enrrejado* (*trellis*, *lattice*).

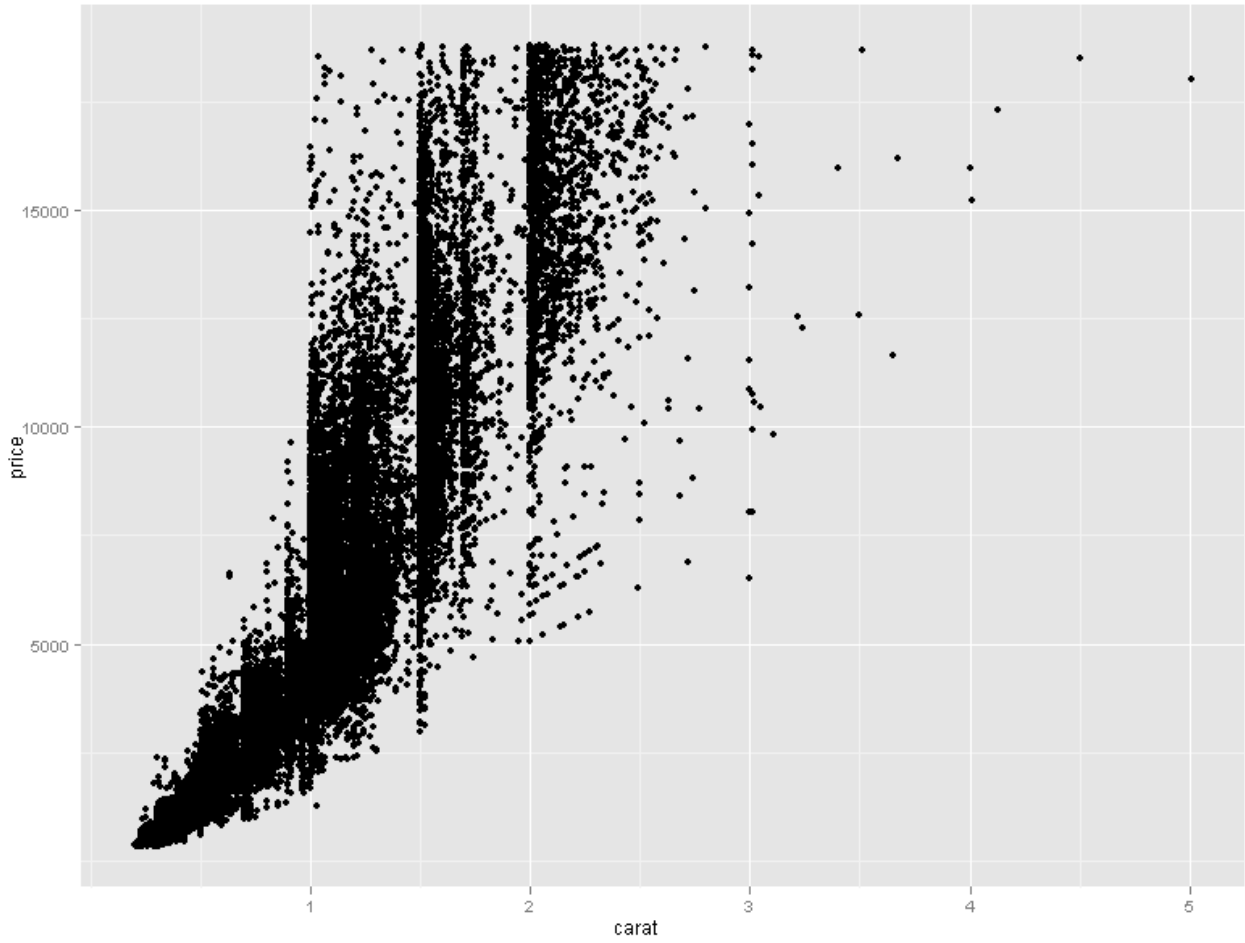
Instalación del paquete y datos

```
R > install.packages("ggplot2")

R > library(ggplot2)
Loading required package: reshape
Loading required package: plyr
Loading required package: grid
Loading required package: proto

R > set.seed(1410) # Make the sample reproducible
R > dsmall <- diamonds[sample(nrow(diamonds), 100), ]

R > head(diamonds)
  carat       cut color clarity depth table price    x    y    z
1  0.23     Ideal     E     SI2  61.5    55   326 3.95 3.98 2.43
2  0.21   Premium     E     SI1  59.8    61   326 3.89 3.84 2.31
3  0.23      Good     E     VS1  56.9    65   327 4.05 4.07 2.31
4  0.29   Premium     I     VS2  62.4    58   334 4.20 4.23 2.63
5  0.31      Good     J     SI2  63.3    58   335 4.34 4.35 2.75
6  0.24 Very Good     J    VVS2  62.8    57   336 3.94 3.96 2.48

R > dim(diamonds)
[1] 53940    10
```

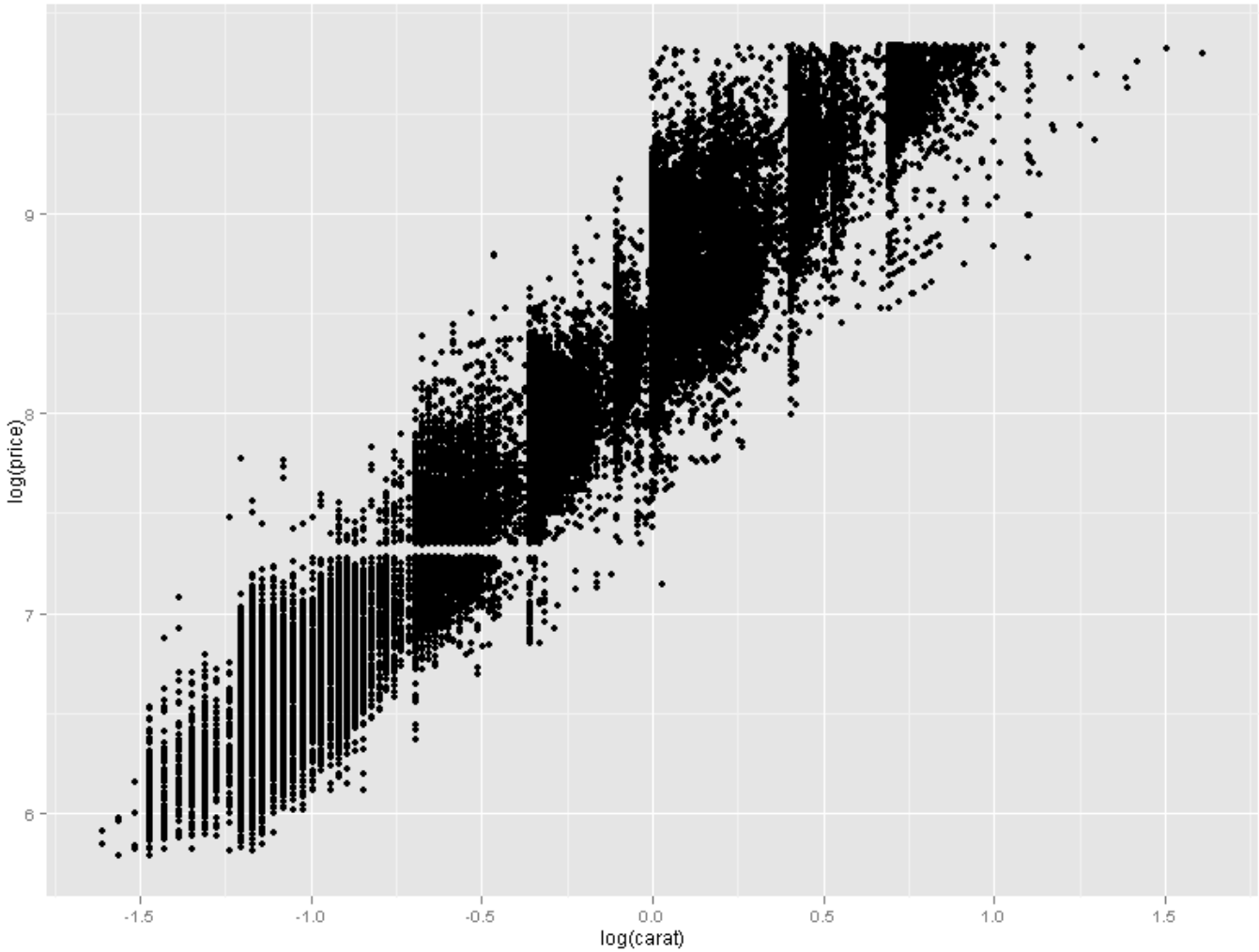
Función qplot

Uso básico

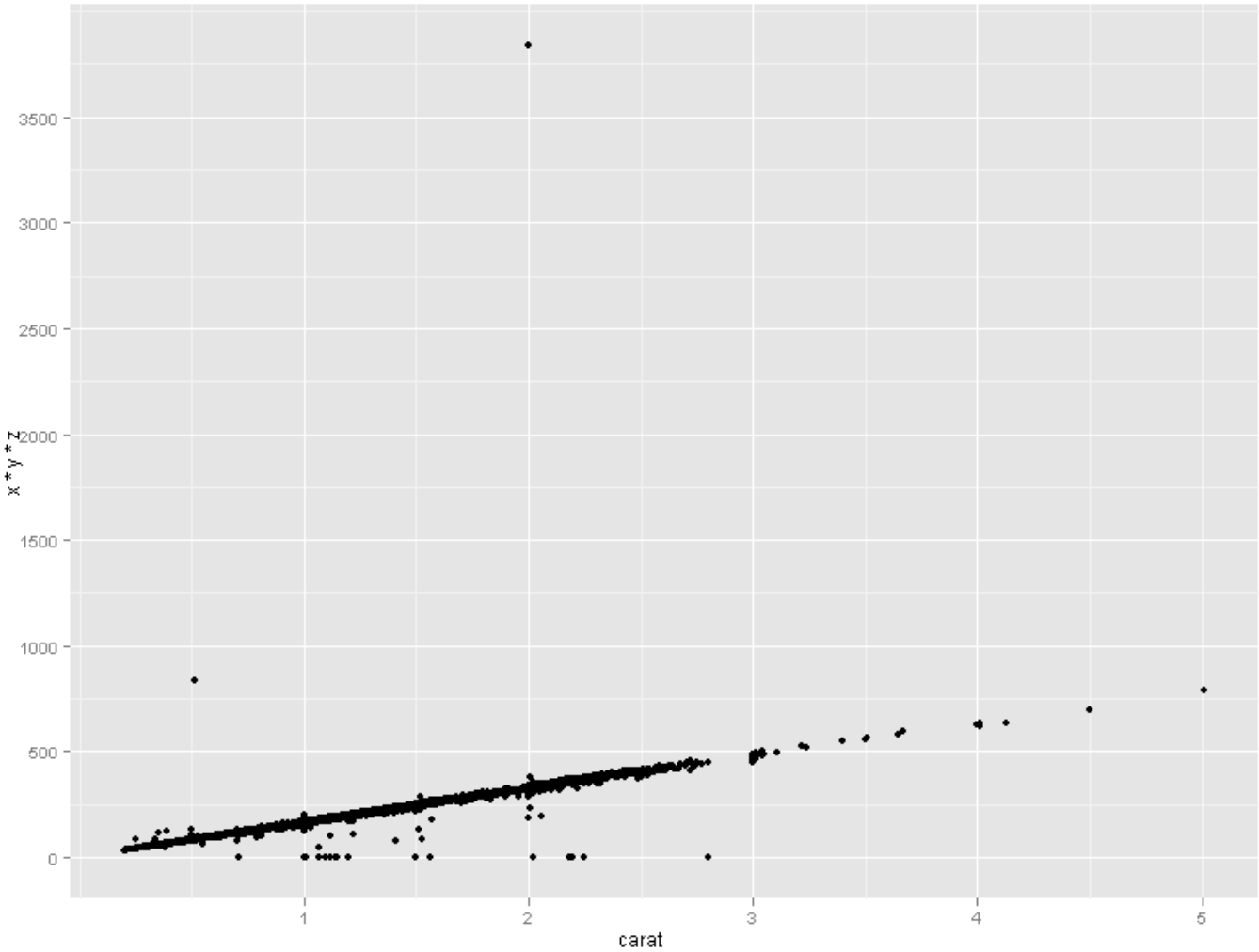
```
R > qplot(carat, price, data = diamonds)
```



```
R > qplot(log(carat), log(price), data = diamonds))
```

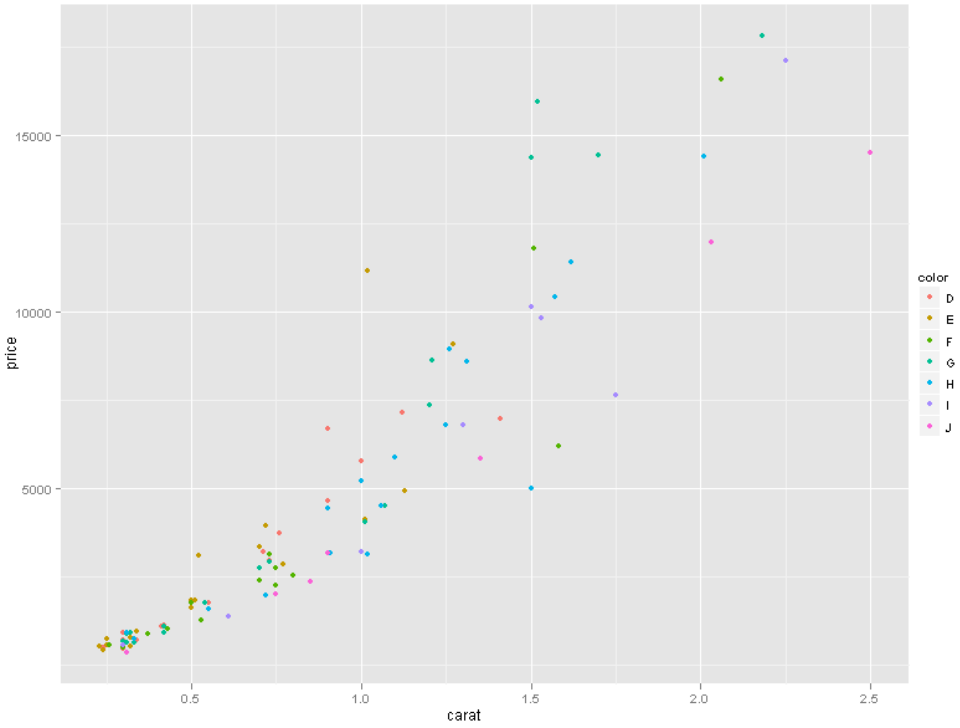


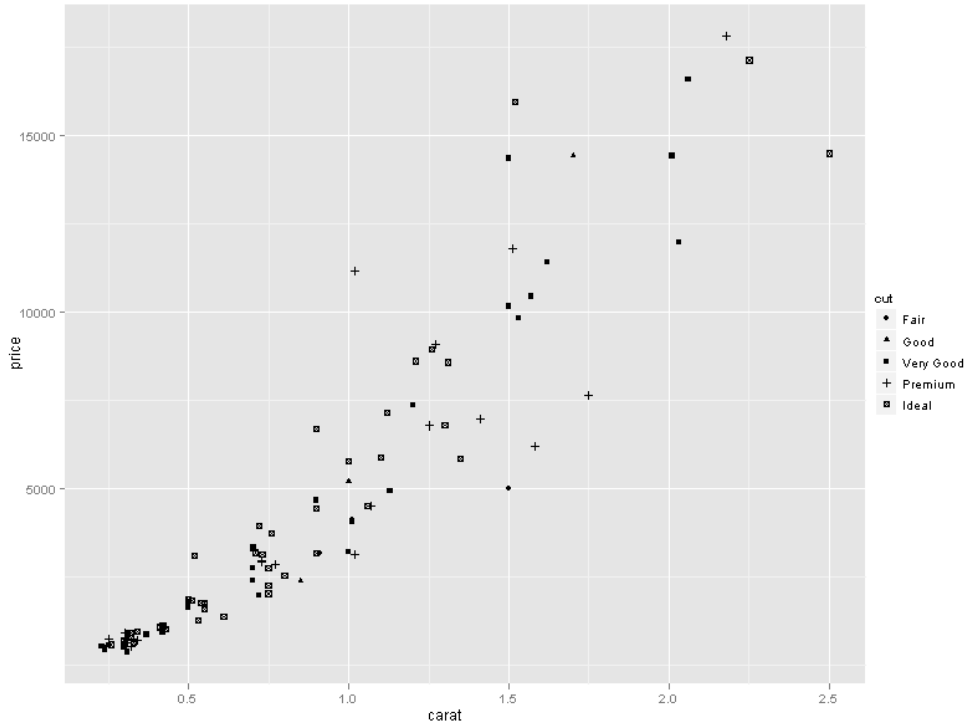
```
R > qplot(carat, x * y * z, data = diamonds))
```



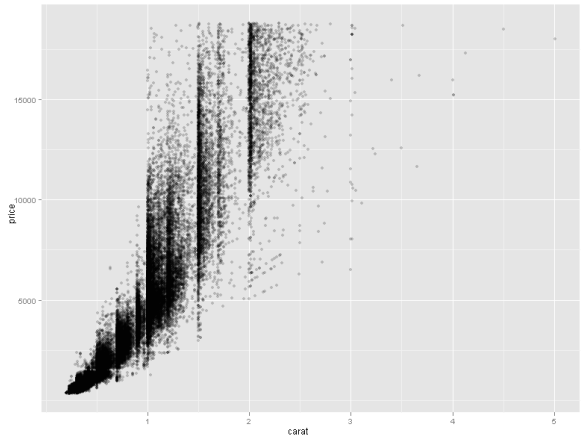
Color, tamaño, formas y otros atributos estéticos

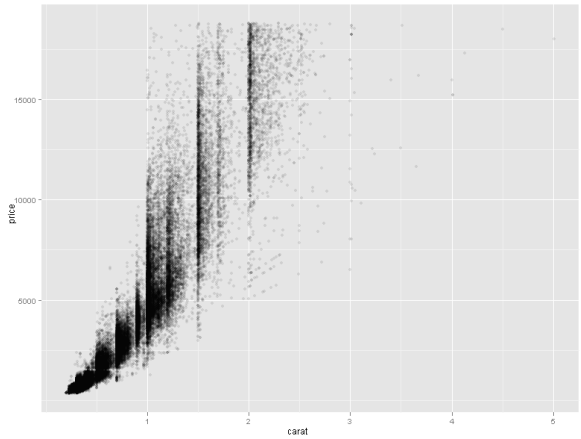
```
R > qplot(carat, price, data = dsmall, colour = color))
R > qplot(carat, price, data = dsmall, shape = cut))
```

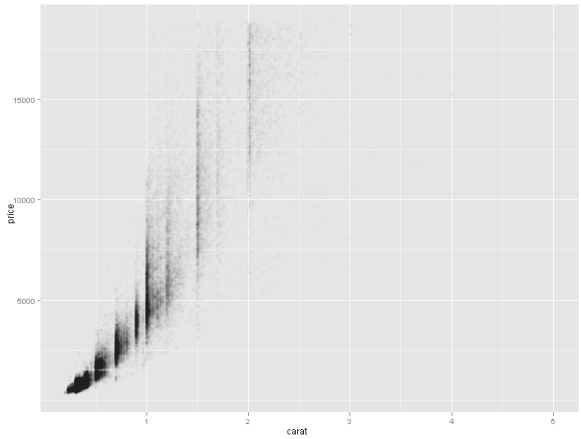




```
R > qplot(carat, price, data = diamonds, alpha = I(1/5))
R > qplot(carat, price, data = diamonds, alpha = I(1/10))
R > qplot(carat, price, data = diamonds, alpha = I(1/50))
```



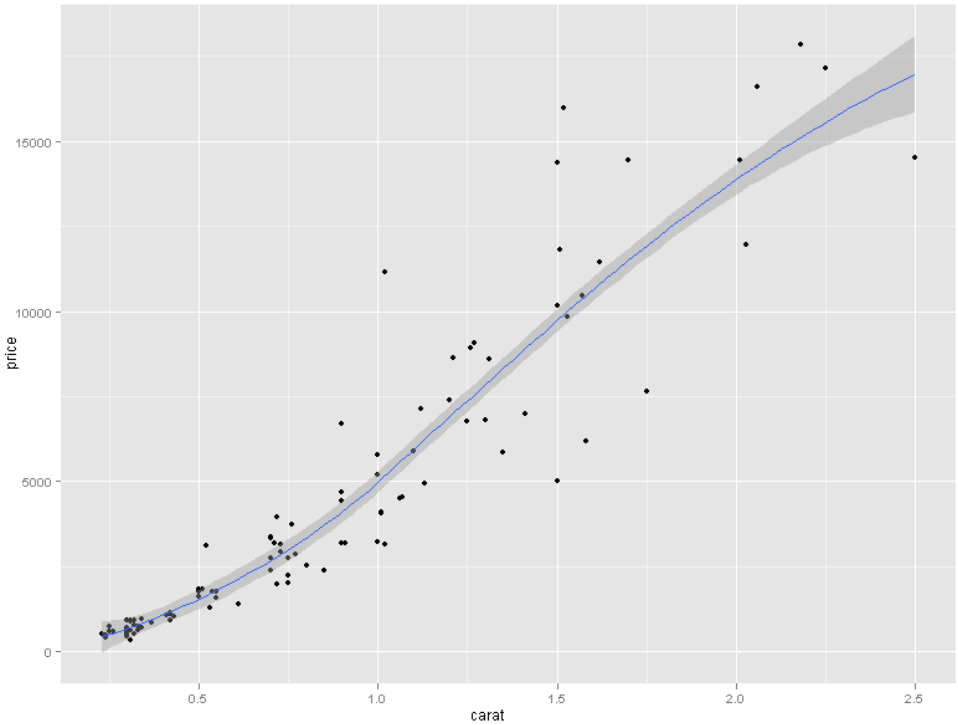


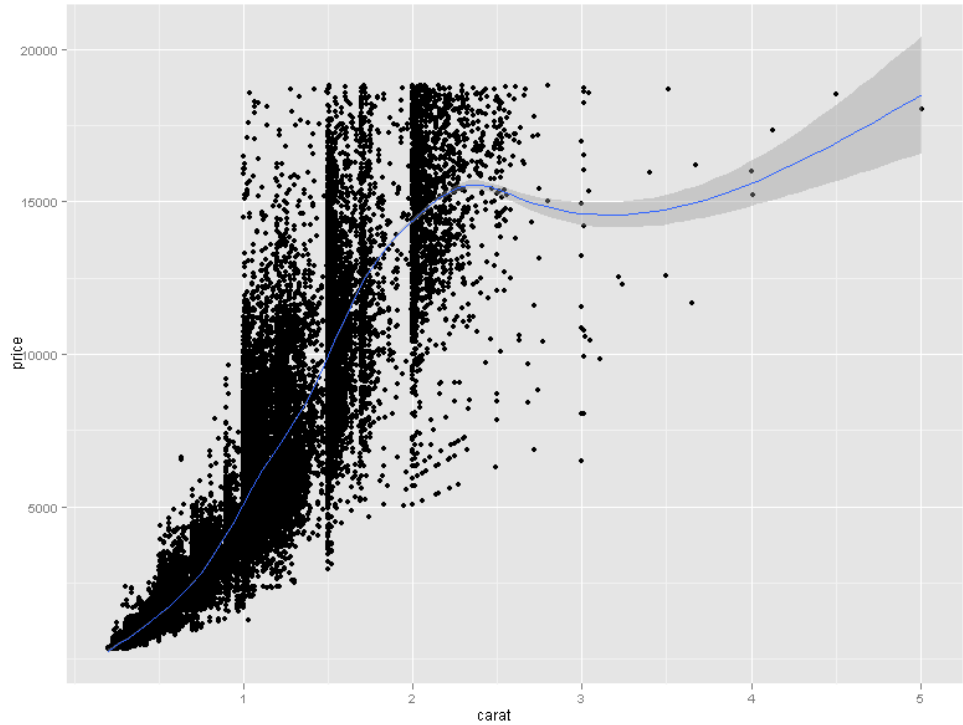


Graficando objetos geométricos (geom)

Añadiendo un alisador al a gráfica

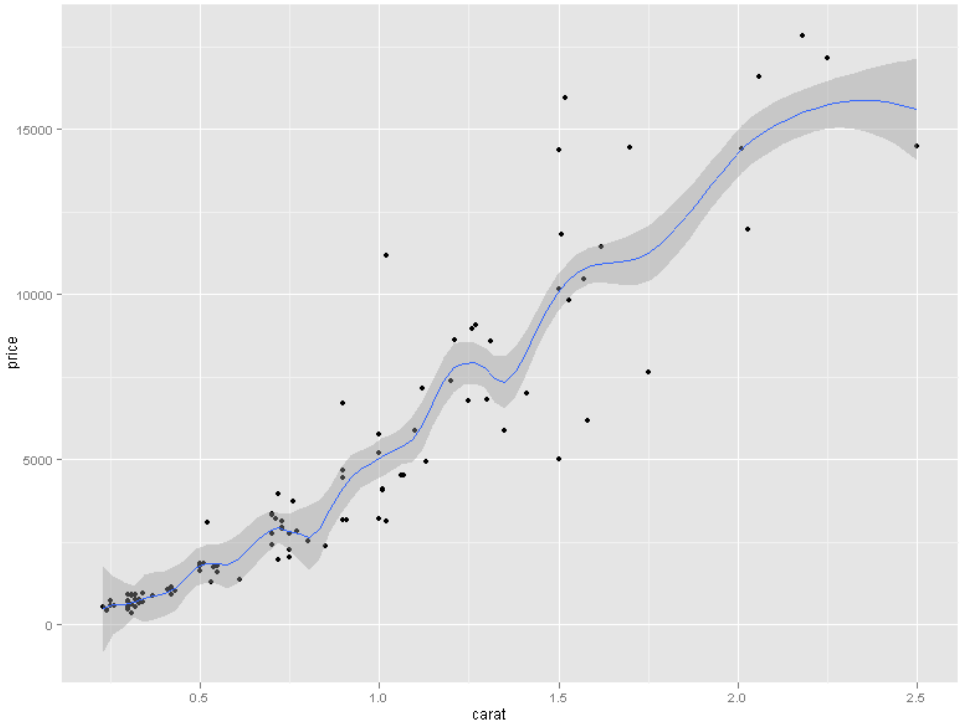
```
R > qplot(carat, price, data = dsmall, geom = c("point", "smooth"))
R > qplot(carat, price, data = diamonds, geom = c("point","smooth"))
```

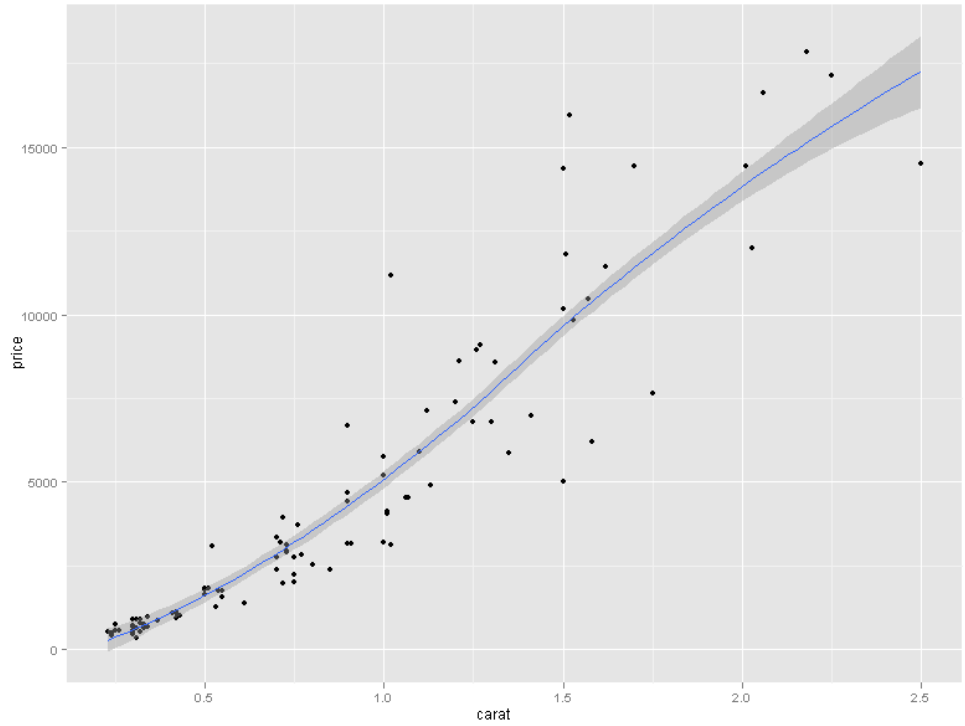




Por defecto loess es usado en pocos datos ($n \leq 1000$)

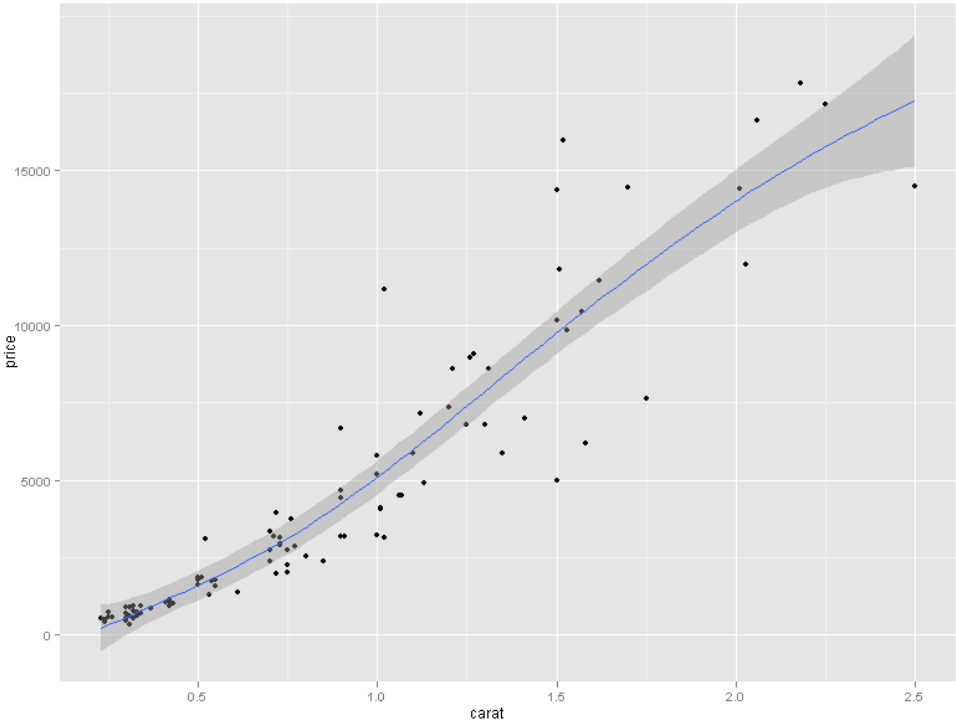
```
R > qplot(carat, price, data = dsmall, geom = c("point","smooth"), span
= 0.2)
R > qplot(carat, price, data = dsmall, geom = c("point","smooth"), span
= 1)
```

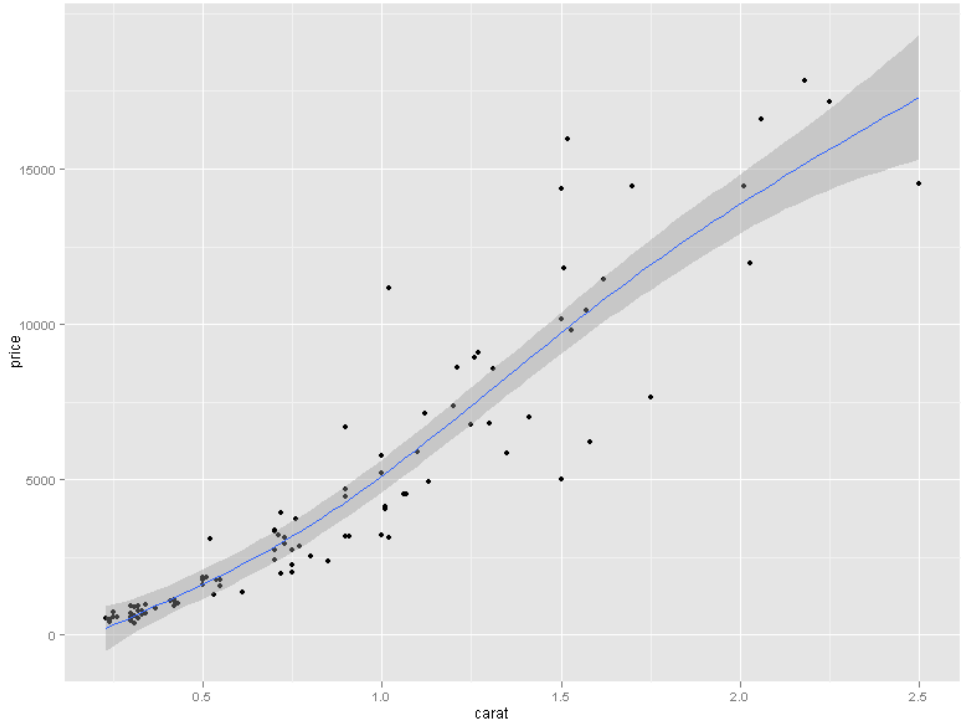




Para juegos de datos grandes gam (modelos aditivos)

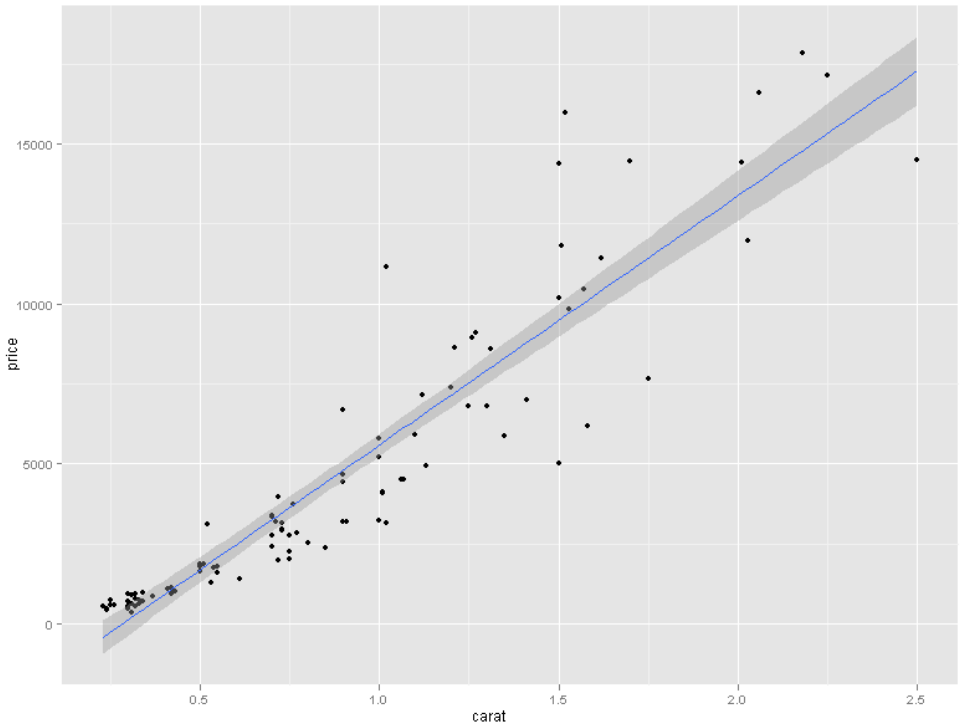
```
R > qplot(carat, price, data = dsmall, geom = c("point", "smooth"),
method = "gam", formula = y ~ s(x))
R > qplot(carat, price, data = dsmall, geom = c("point", "smooth"),
method = "gam", formula = y ~ s(x, bs = "cs"))
```

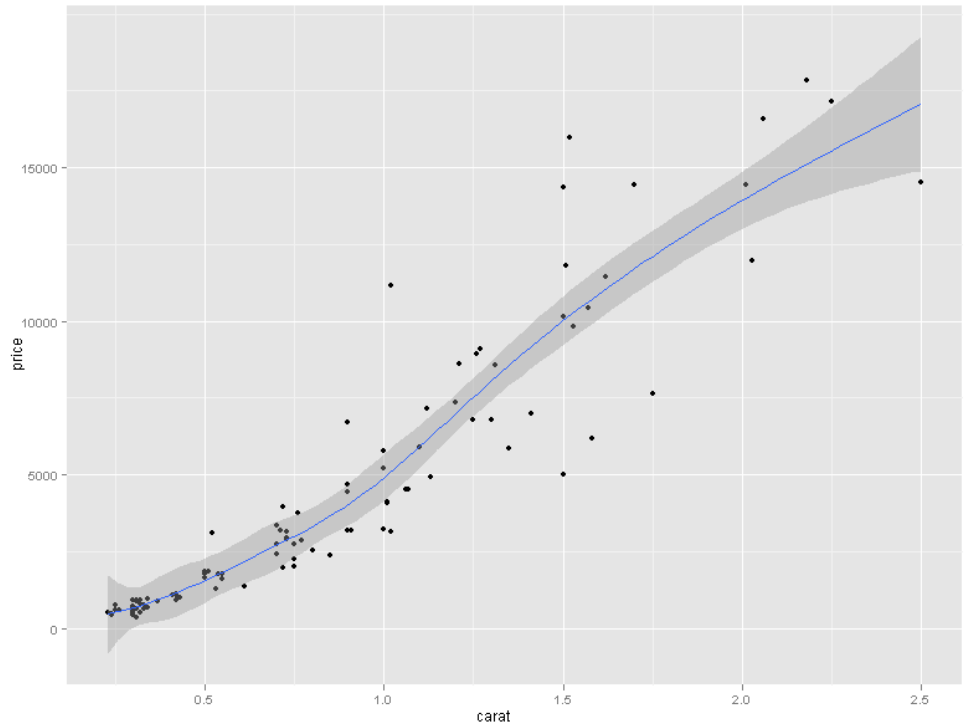




Se pueden modelos lineales (lm), splines, o ajuste robusto (rlm) para alisamiento

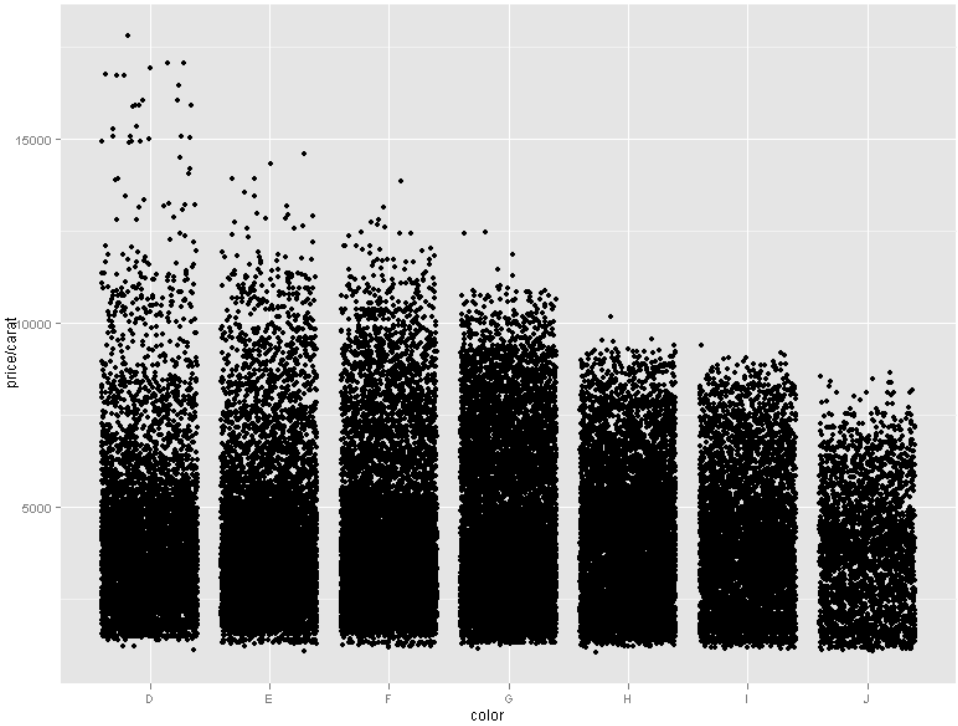
```
R > qplot(carat, price, data = dsmall, geom = c("point", "smooth"),
method = "lm")
R > qplot(carat, price, data = dsmall, geom = c("point", "smooth"),
method = "lm", formula = y ~ ns(x,5))
```

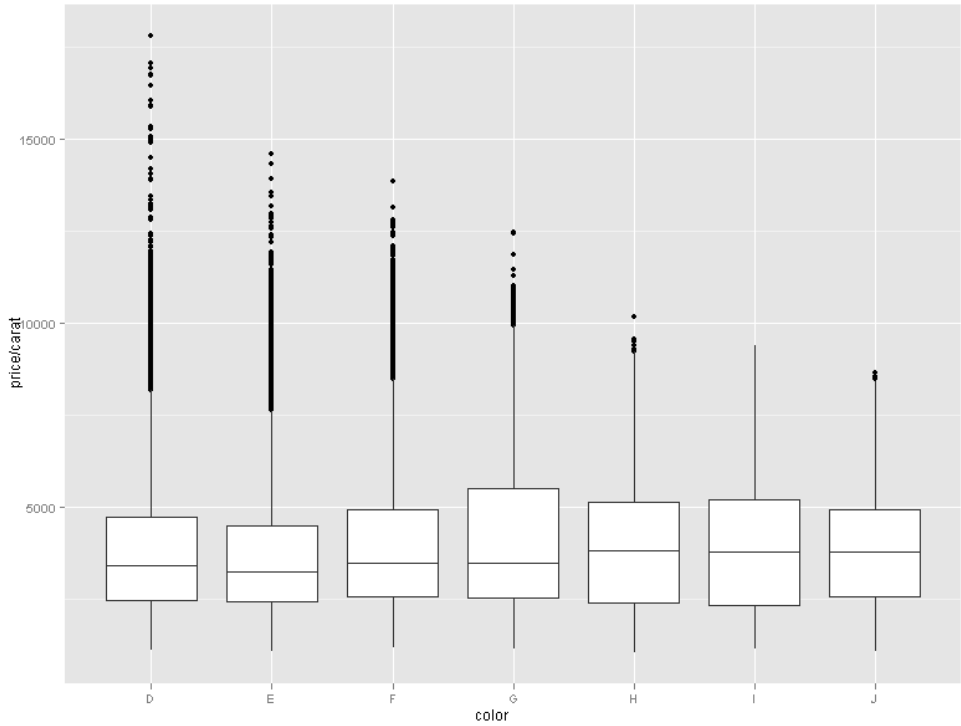




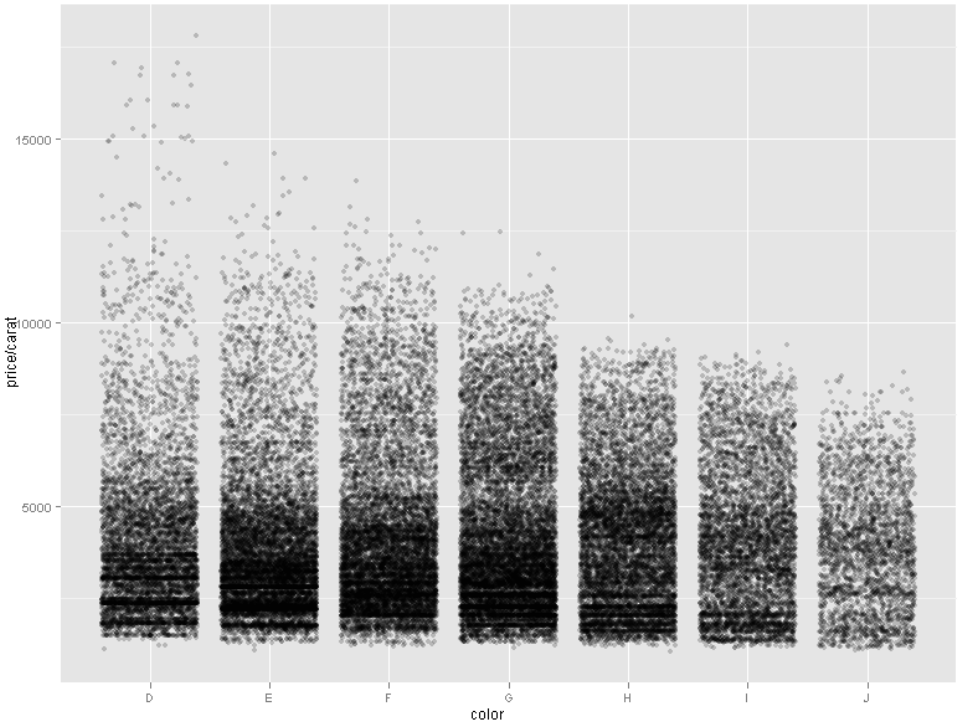
Diagramas de caja y puntos *"agitados"*

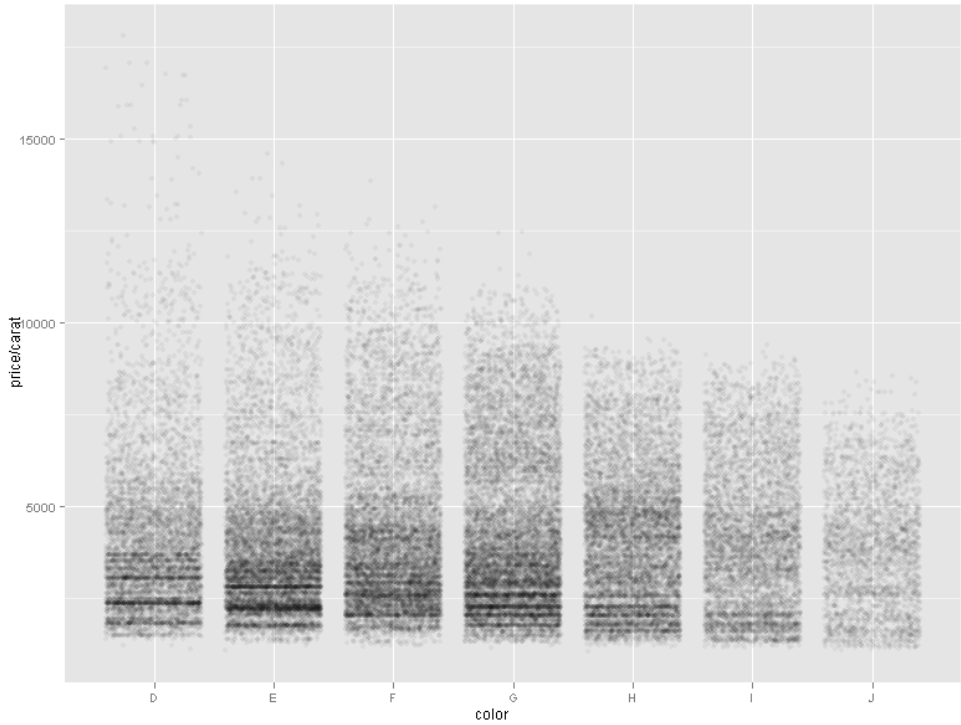
```
R > qplot(color, price / carat, data = diamonds, geom = "jitter")
R > qplot(color, price / carat, data = diamonds, geom = "boxplot")
```





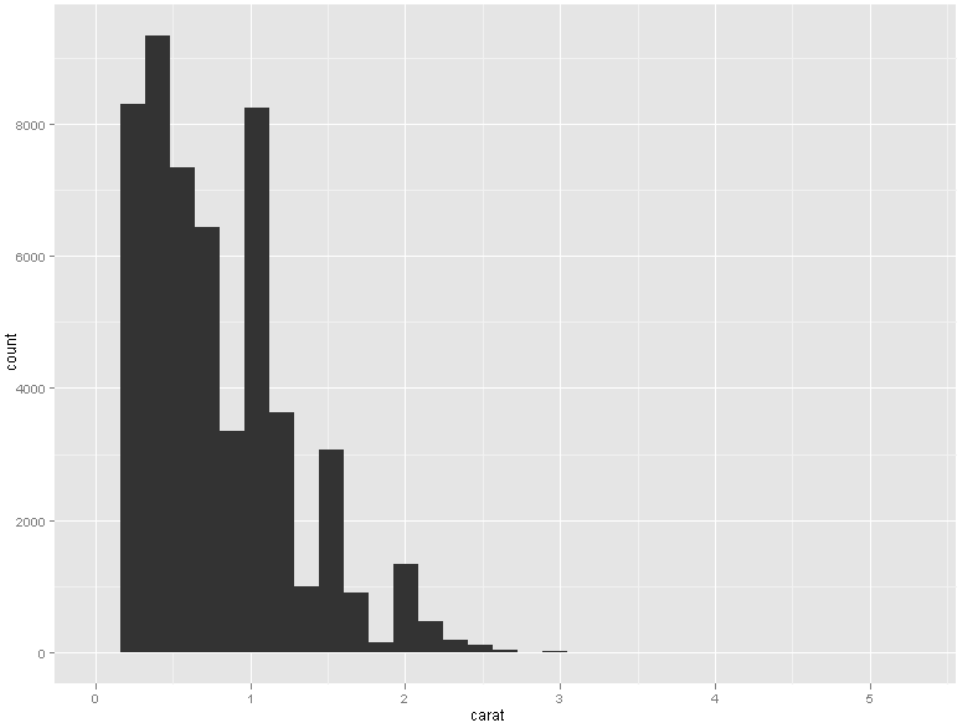
```
R> qplot(color, price / carat, data = diamonds, geom = "jitter", alpha
= I(1 / 5))
R > qplot(color, price / carat, data = diamonds, geom = "jitter", alpha
= I(1 / 20))
```

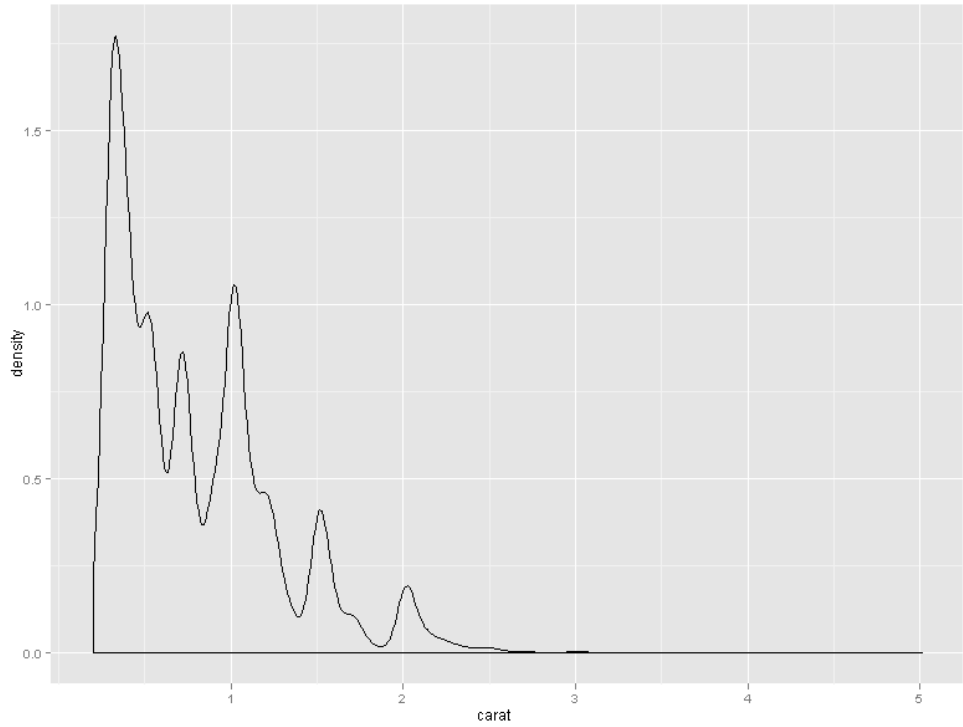




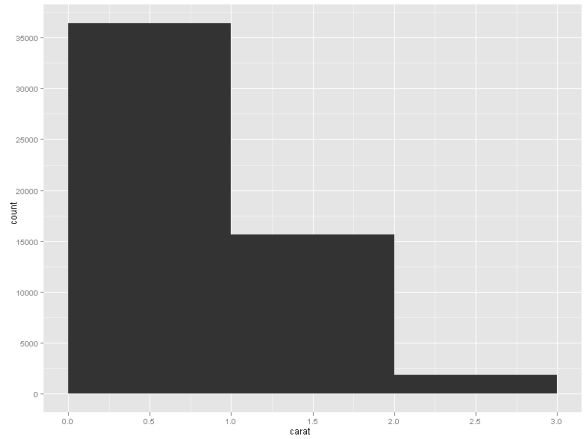
Histogramas, gráfica de densidades

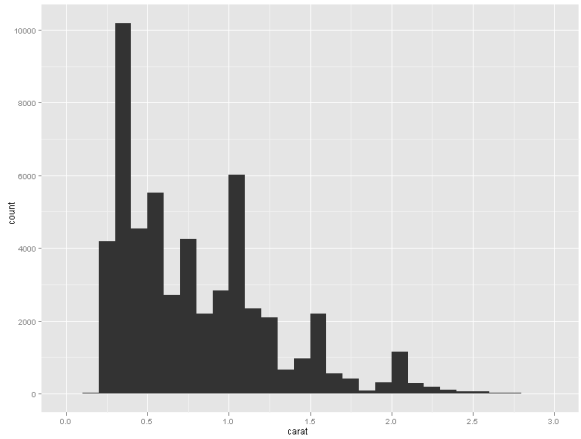
```
R > qplot(carat, data = diamonds, geom = "histogram")
R > qplot(carat, data = diamonds, geom = "density")
```

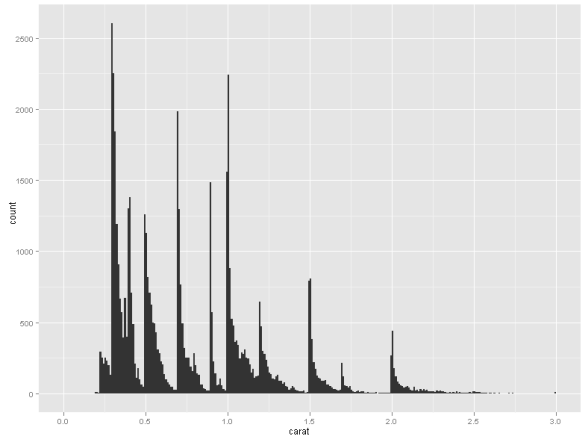




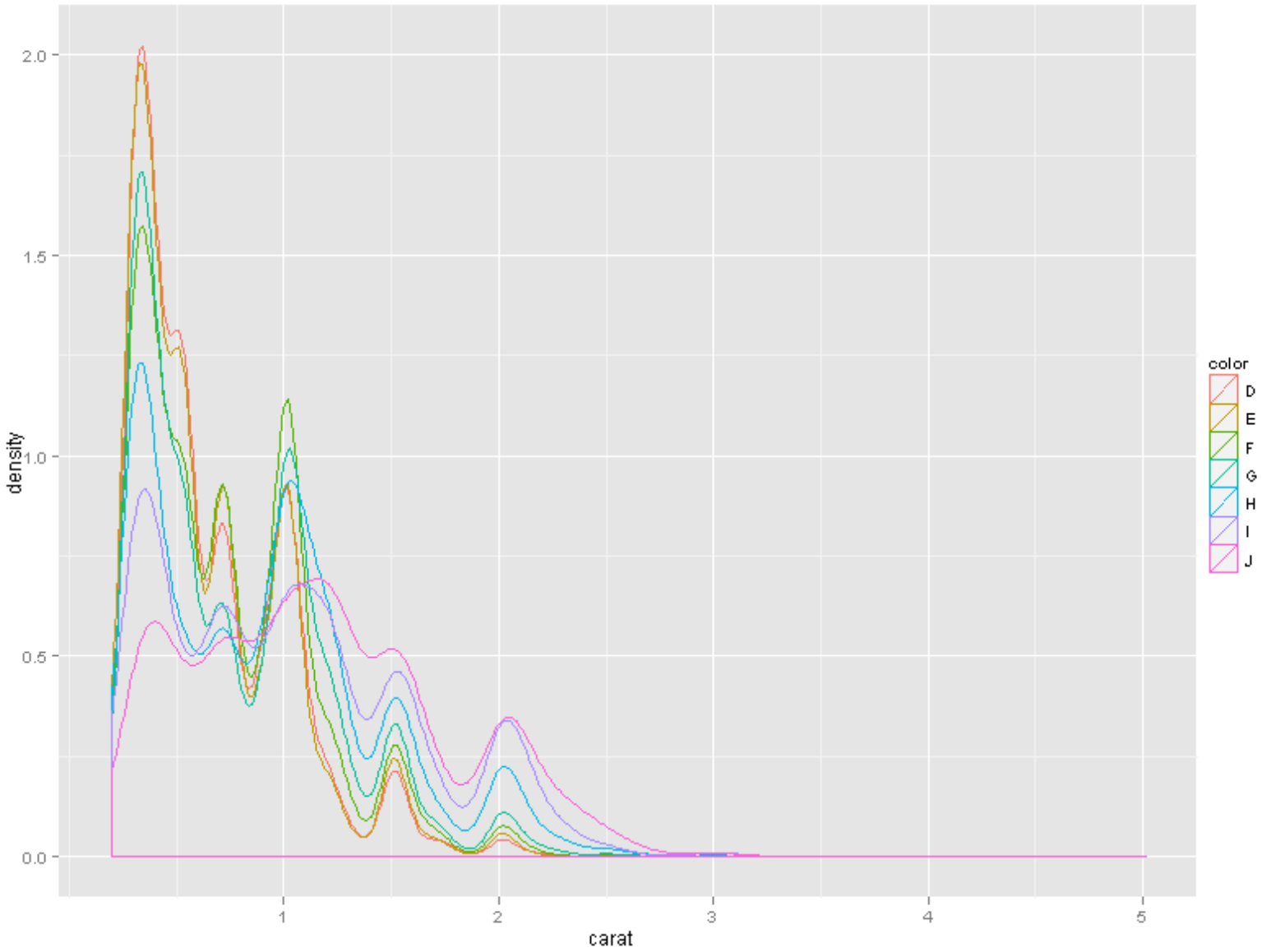
```
R > qplot(carat, data = diamonds, geom = "histogram", binwidth = 1,
xlim = c(0,3))
R > qplot(carat, data = diamonds, geom = "histogram", binwidth = 0.1,
xlim = c(0,3))
R > qplot(carat, data = diamonds, geom = "histogram", binwidth = 0.01,
xlim = c(0,3))
```







```
R > qplot(carat, data = diamonds, geom = "density", colour = color)
```



```
R > qplot(carat, data = diamonds, geom = "histogram", fill = color)
```

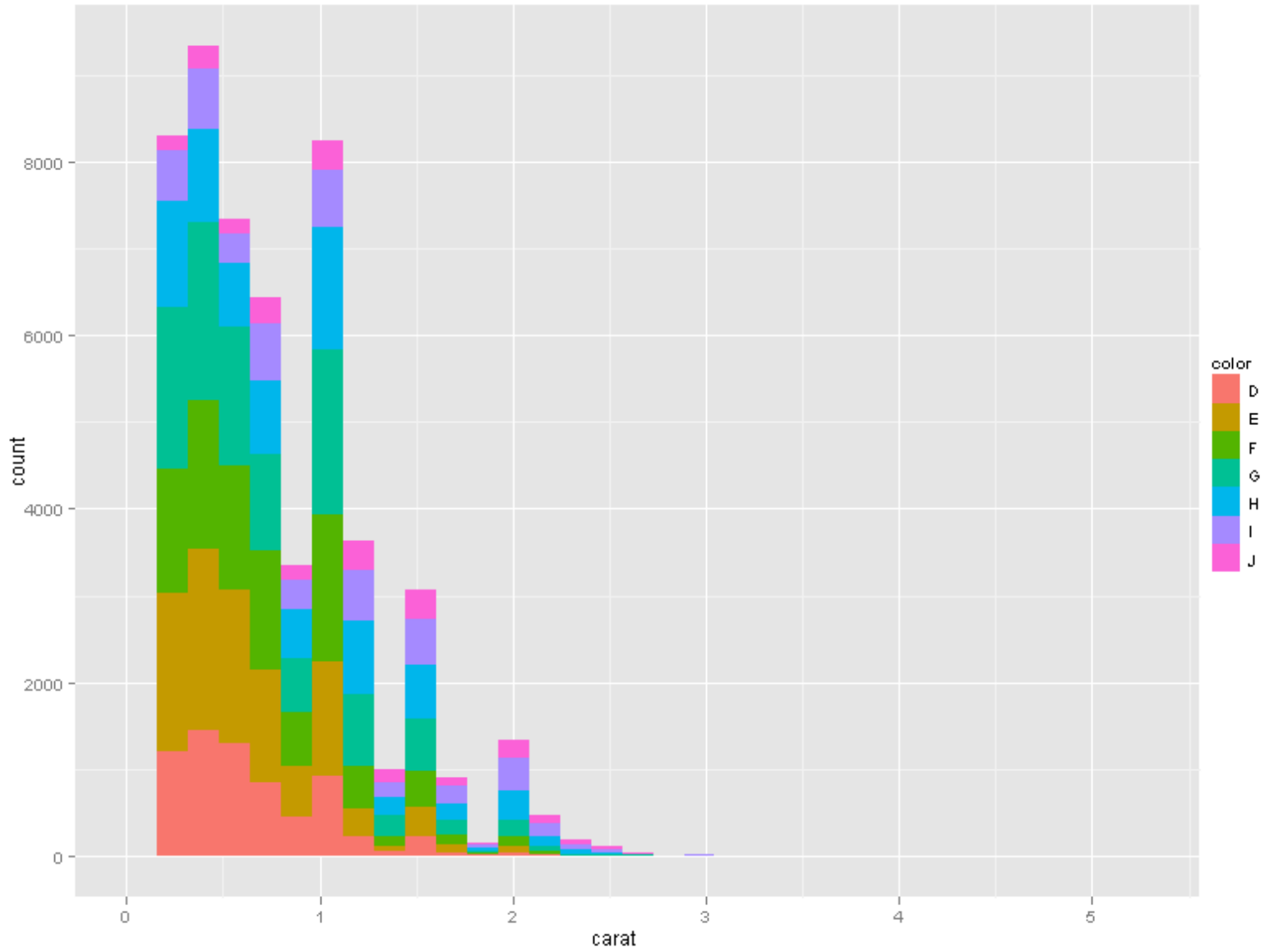
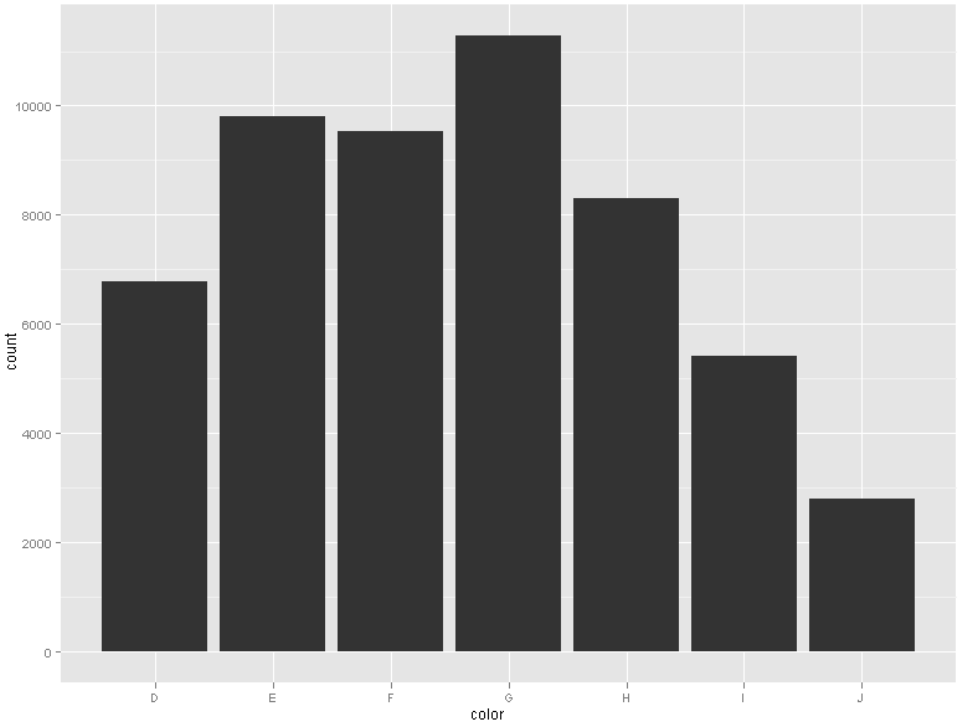
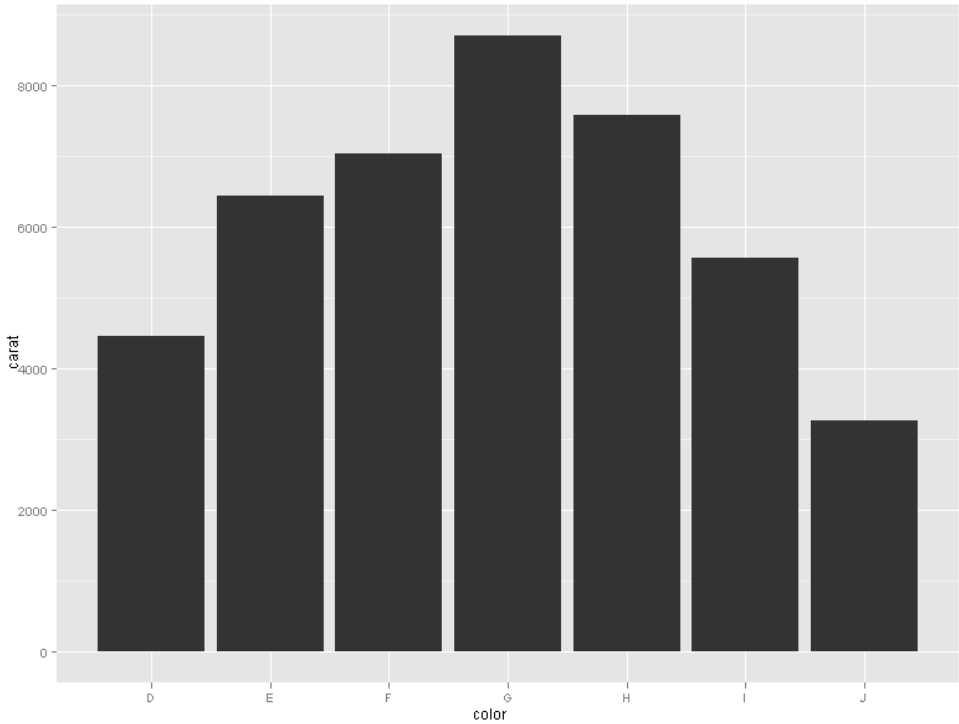


Diagrama de barras

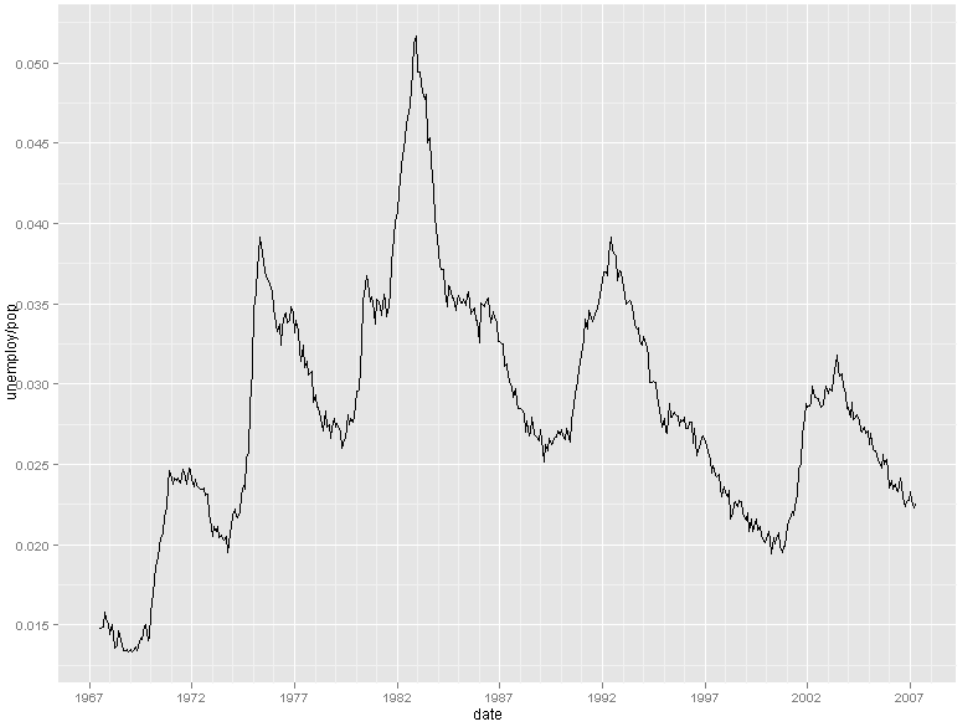
```
R > qplot(color, data = diamonds, geom = "bar")
R > qplot(color, data = diamonds, geom = "bar", weight = carat) +
scale_y_continuous("carat")
```



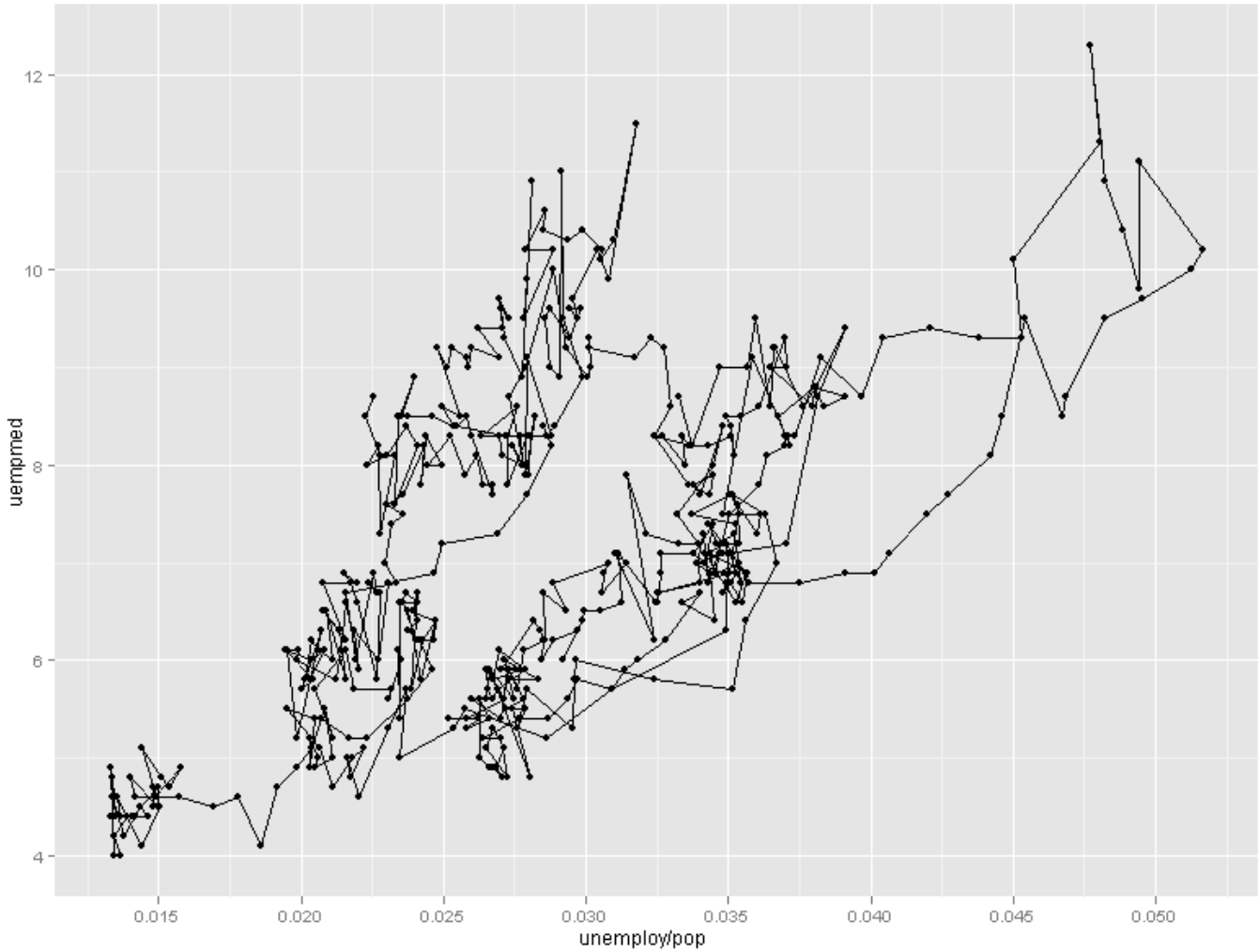


Mostrando series de tiempo con líneas y trayectorias

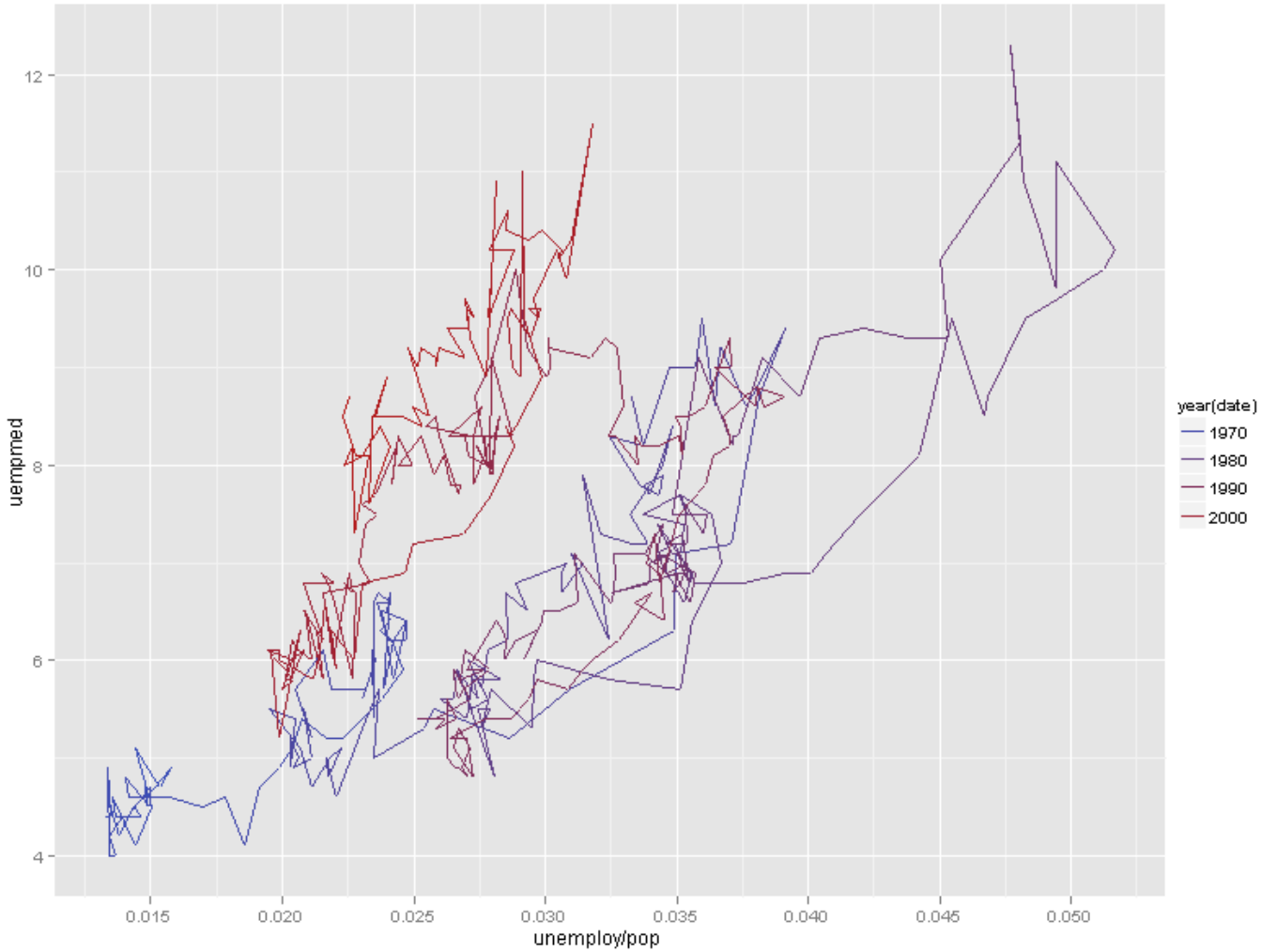
```
R > year <- function(x) as.POSIXlt(x)$year + 1900
R > qplot(date, unemploy / pop, data = economics, geom = "line")
R > qplot(date, uempmed, data = economics, geom = "line")
```



```
R > year <- function(x) as.POSIXlt(x)$year + 1900
R > qplot(unemploy / pop, uempmed, data = economics, geom = c("point",
"path"))
```

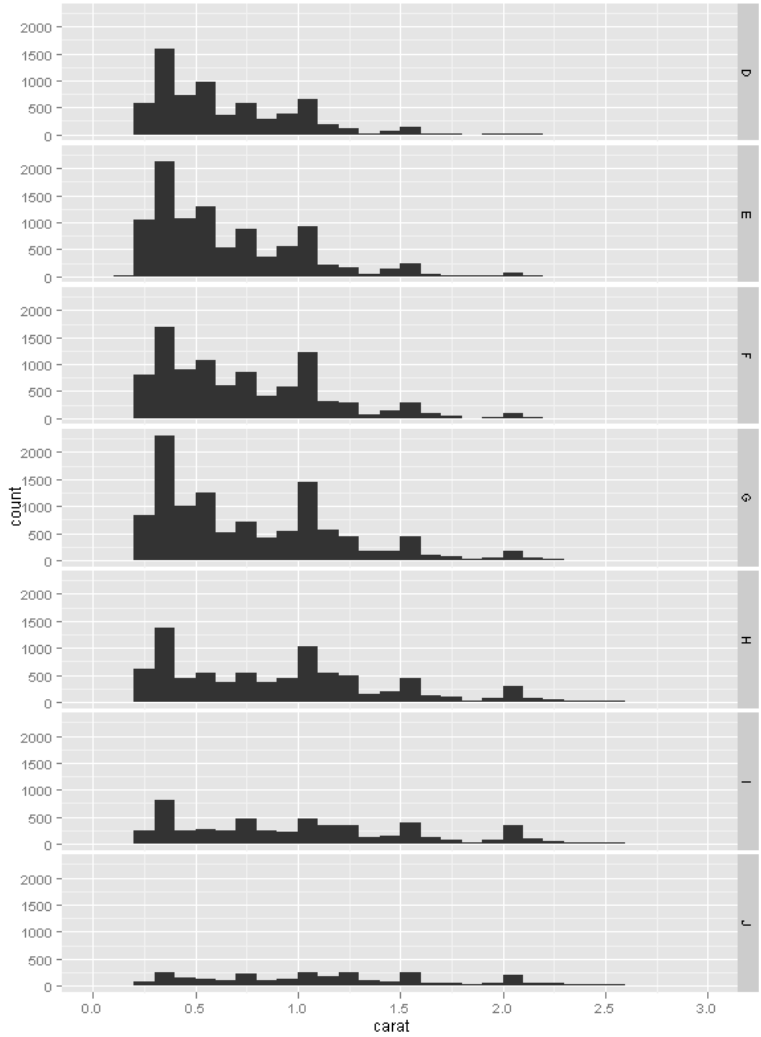


```
R > year <- function(x) as.POSIXlt(x)$year + 1900
R > qplot(unemploy / pop, uempmed, data = economics, geom = "path",
colour = year(date)) + scale_area()
```

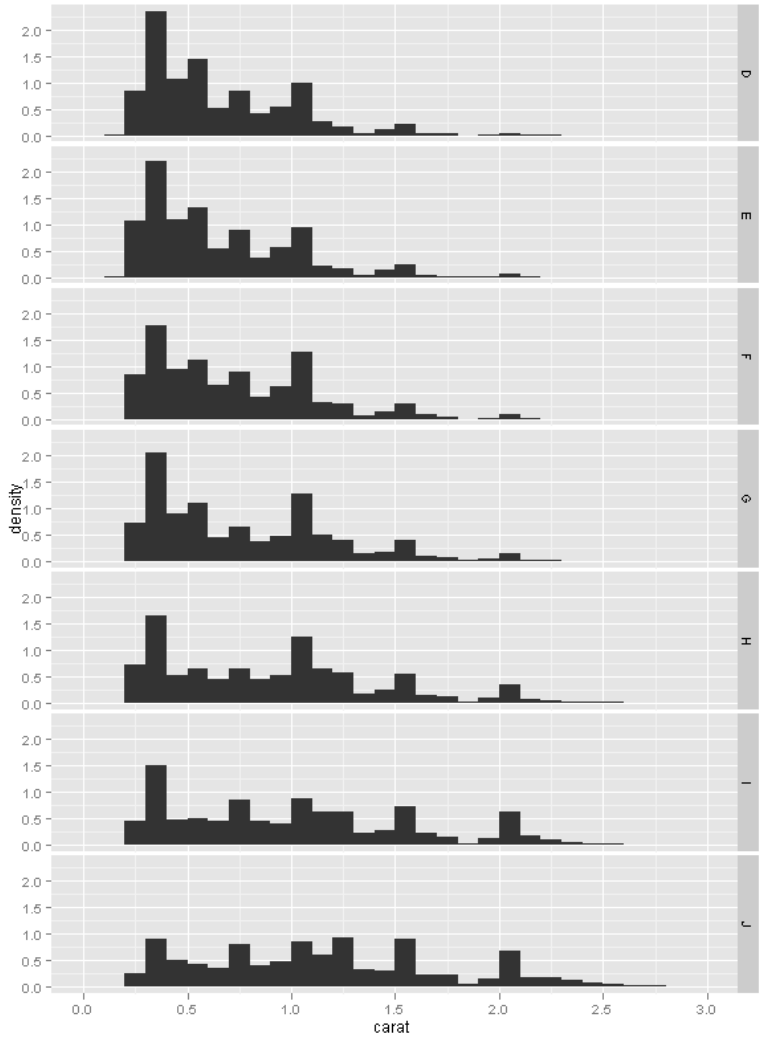


Faceting: partir los datos para distintas presentaciones

```
R > qplot(carat, data = diamonds, facets = color ~ ., geom =
"histogram", binwidth = 0.1, xlim = c(0, 3))
```



```
R > qplot(carat, ..density.., data = diamonds, facets = color ~ .,
geom = "histogram", binwidth = 0.1, xlim = c(0, 3))
```



Creando una gráfica

```
R > p <- ggplot(diamonds,aes(carat, price, colour = cut))
```

Capas

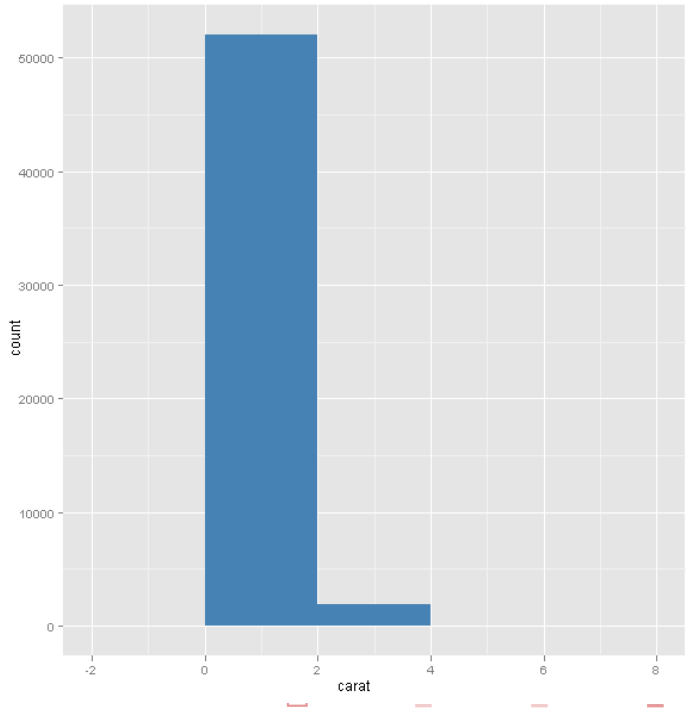
```
R > p <- p + layer(geom = "point")
```

Argumentos de la función `layer`

```
layer(geom, geom_params, stat, stat_params, data, mapping,position)
```

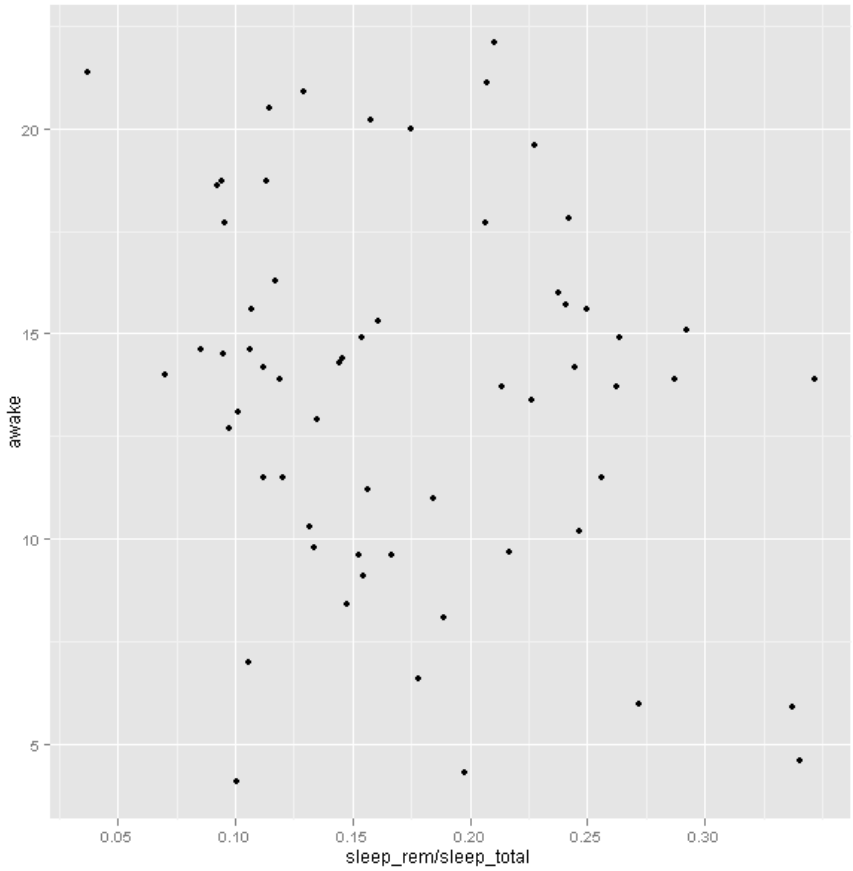
Ejemplo:

```
p <- ggplot(diamonds, aes(x = carat))
p <- p + layer(
  geom = "bar",
  geom_params = list(fill = "steelblue"),
  stat = "bin",
  stat_params = list(binwidth = 2)
)
p
```



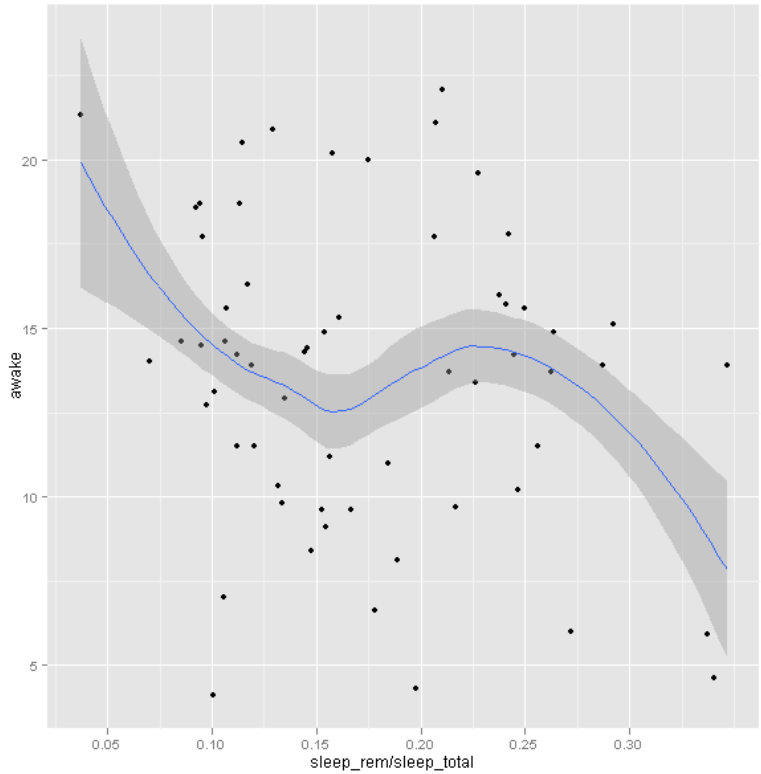
Gráficas equivalentes.

```
R > ggplot(msleep, aes(sleep_rem / sleep_total, awake)) +  geom_point()

R> qplot(sleep_rem / sleep_total, awake, data = msleep)
```



qplot también acepta capas adicionales

```
R > qplot(sleep_rem / sleep_total, awake, data = msleep) +
    geom_smooth()

R > qplot(sleep_rem / sleep_total, awake, data = msleep,
    geom = c("point", "smooth"))

R > ggplot(msleep, aes(sleep_rem / sleep_total, awake)) +
    geom_point() + geom_smooth()
```



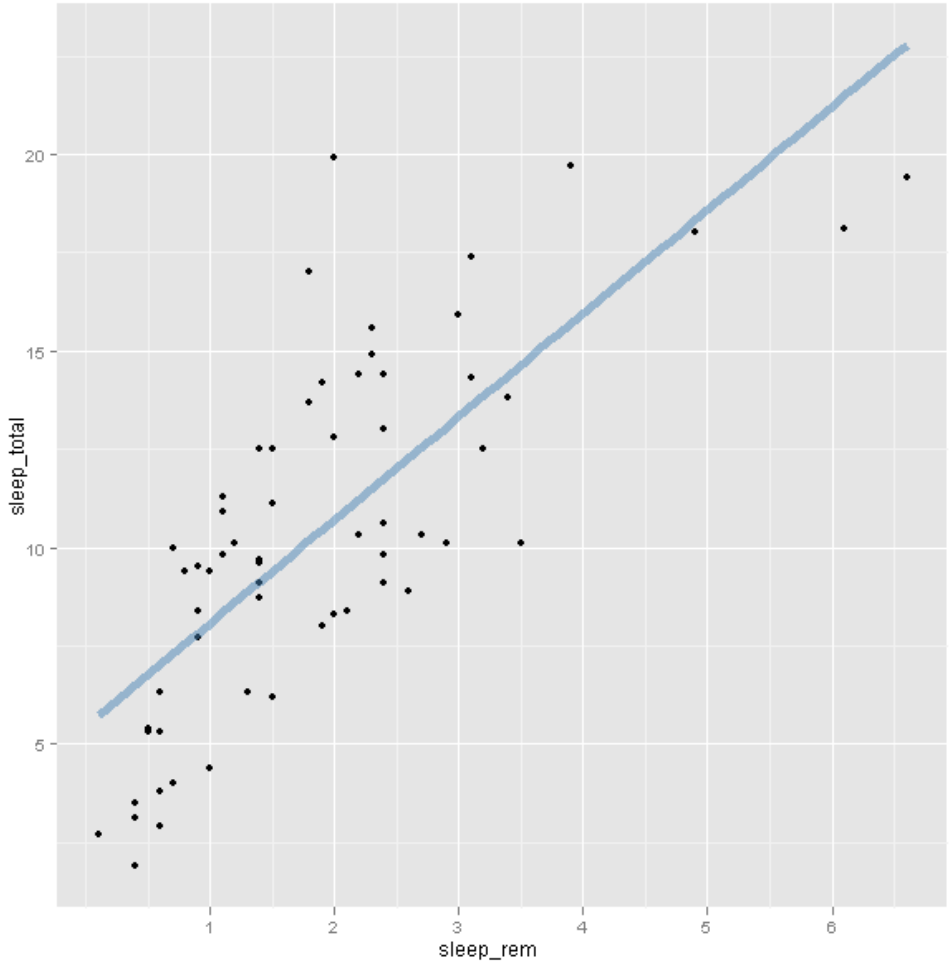
Información contenida en los objetos gráfica

```
R > p <- ggplot(msleep, aes(sleep_rem / sleep_total, awake))
R > summary(p)
data: name, genus, vore, order, conservation, sleep_total, sleep_rem,
  sleep_cycle, awake, brainwt, bodywt [83x11]
mapping:  x = sleep_rem/sleep_total, y = awake
faceting: facet_grid(. ~ ., FALSE)
```

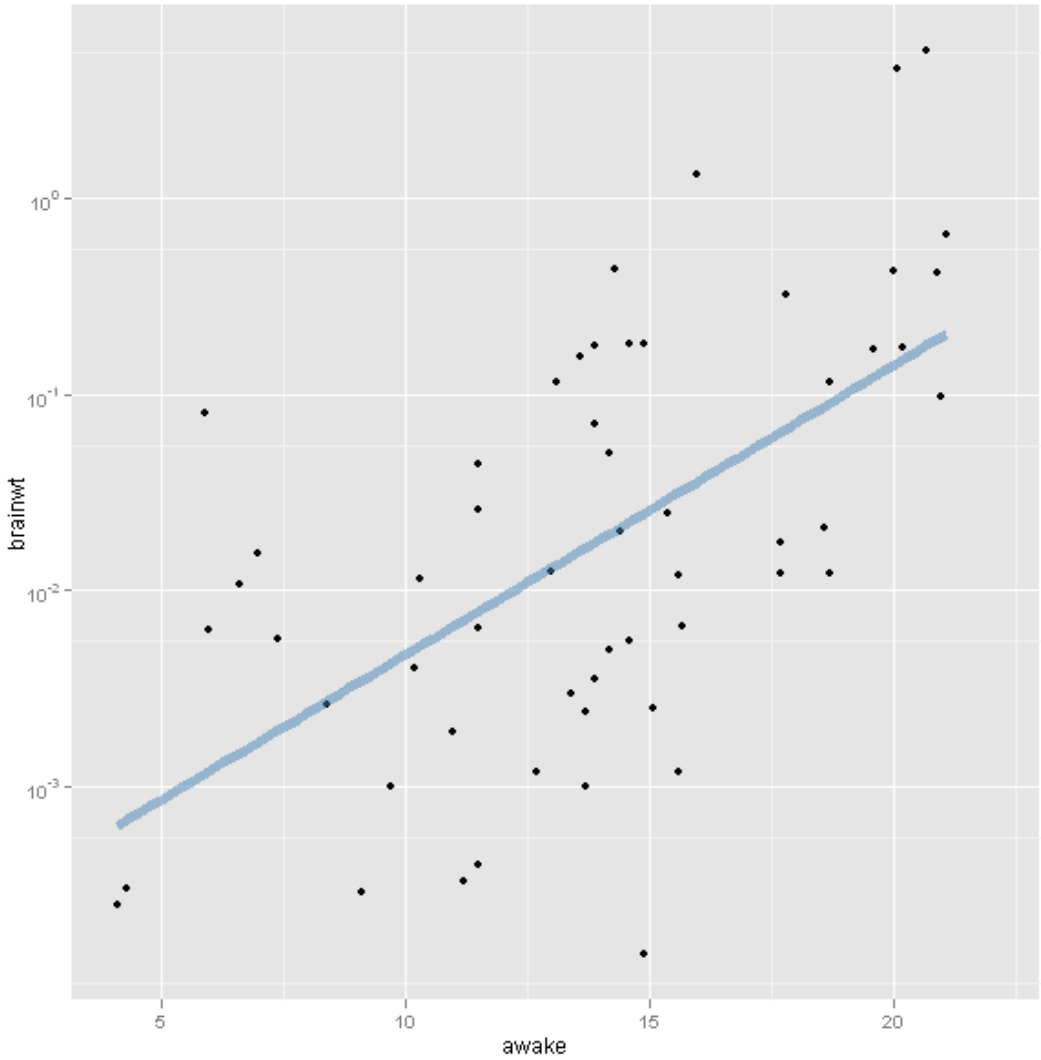
```
R > p <- p + geom_point()
R > summary(p)
data: name, genus, vore, order, conservation, sleep_total, sleep_rem,
  sleep_cycle, awake, brainwt, bodywt [83x11]
mapping:  x = sleep_rem/sleep_total, y = awake
faceting: facet_grid(. ~ ., FALSE)
-----------------------------------
geom_point: na.rm = FALSE
stat_identity:
position_identity: (width = NULL, height = NULL)
```

Las gráficas son objetos R

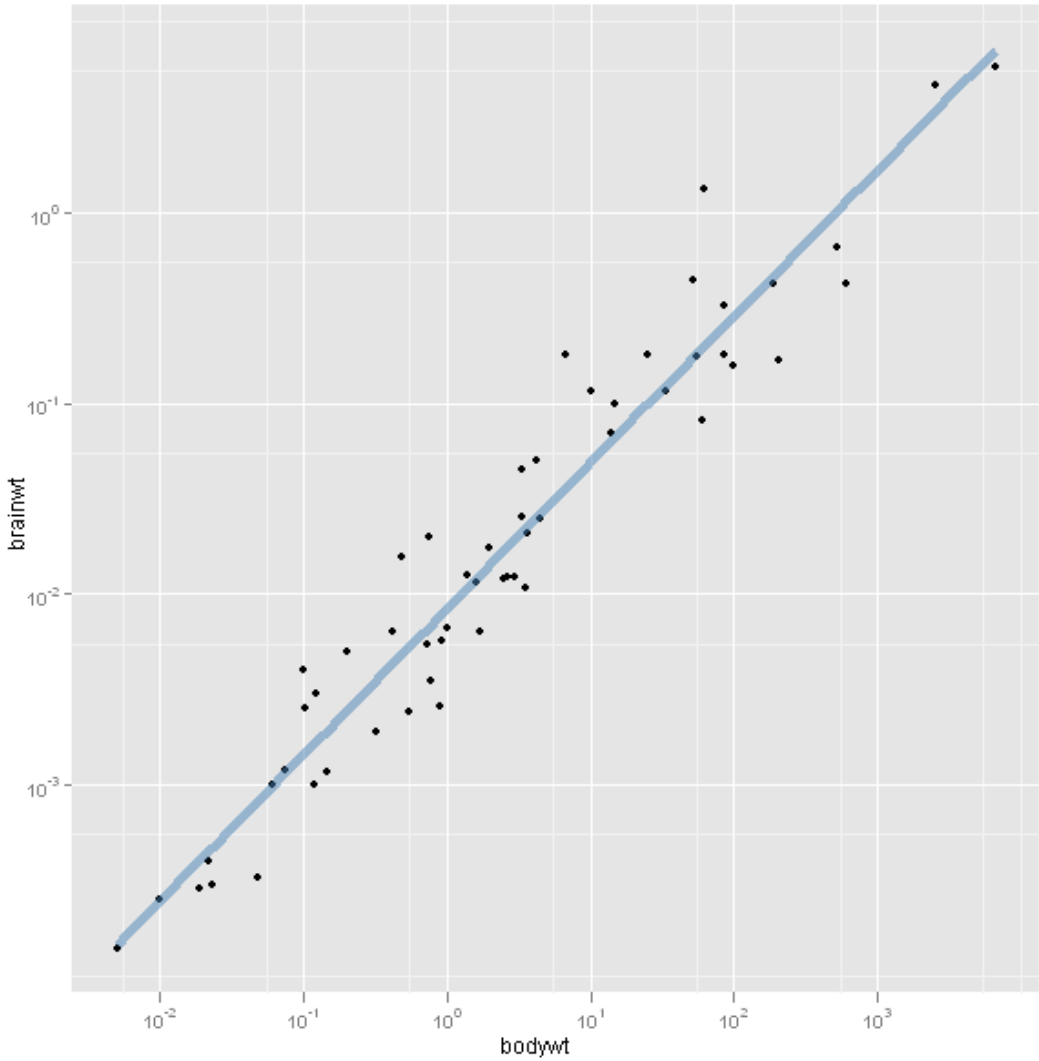
```
R > bestfit <- geom_smooth(method = "lm", se = F,
  colour = alpha("steelblue", 0.5), size = 2)

R > qplot(sleep_rem, sleep_total, data = msleep) + bestfit
```



```
R > qplot(awake, brainwt, data = msleep, log = "y") + bestfit
```



```
R > qplot(bodywt, brainwt, data = msleep, log = "xy") + bestfit
```



ggplot2 website

Información disponible en la red:
http://had.co.nz/ggplot2/

Resumen

- `ggplot2`, de Hadley Wickham, es la implementación en *R* de la gramática de gráficas de Wilkinson.
- Por lo mismo detrás de `ggplot2` hay un modelo formal, lo que facilita la expansión.
- Las gráficas se construyen por capas que se pueden adicionar en tiempo real o después.

Agradecimientos

1. A *Hadley Wickham* por compartir su material generosamente.
2. Comunidad *R*.
3. Comunidad *código libre* y *código abierto*.

Referencias

Las incluidas en la presentación de las pláticas.