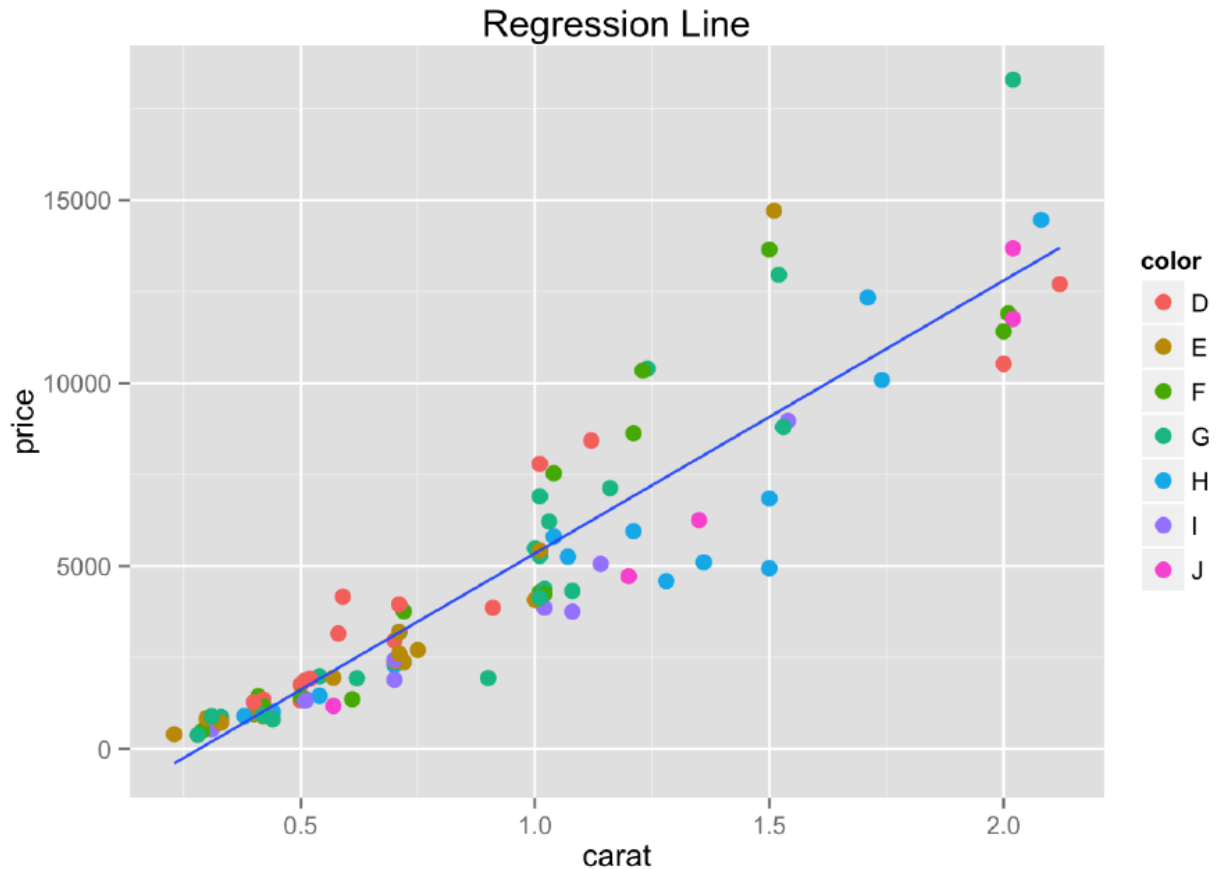


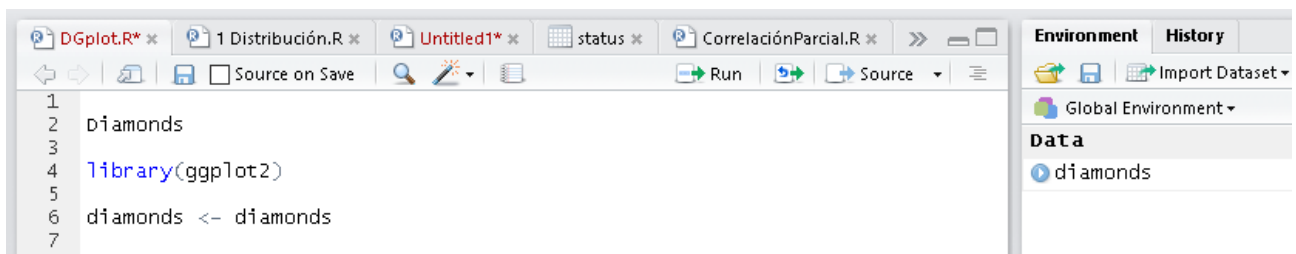
Ejercicio Ggplot2

Con el dataset `diamonds` intentar reproducir el gráfico:



Primero importamos libreria `ggplot2` que es donde viene el data set `diamonds`:

```
library(ggplot2)
```



Con `head(diamonds)` vamos a ver que columnas tiene:

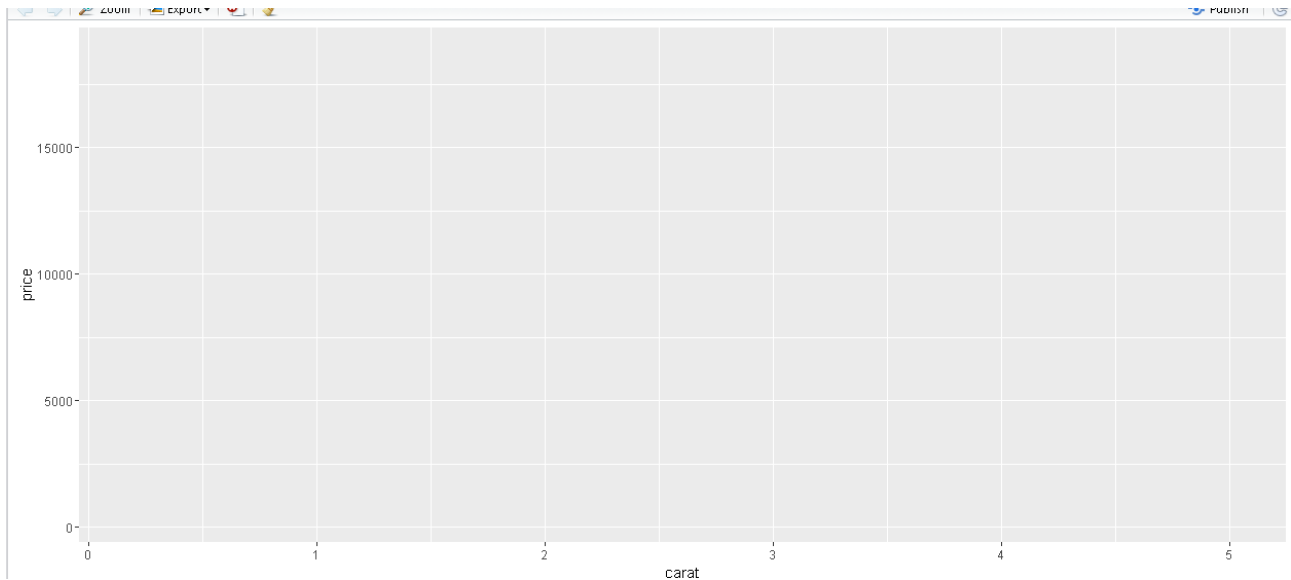
```
> head(diamonds)
  carat cut    color clarity depth table price     x     y     z
1  0.23 Ideal    E     SI2   61.5    55   326  3.95  3.98  2.43
2  0.21 Premium  E     SI1   59.8    61   326  3.89  3.84  2.31
3  0.23 Good    E     VS1   56.9    65   327  4.05  4.07  2.31
4  0.29 Premium  I     VS2   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
> |
```

Después creamos el gráfico ggplot2:

```
graf <- ggplot(diamonds, aes(x=carat, y=price))
```

```
graf
```

Con esto creamos la capa vacía con price es el eje Y y carat el X

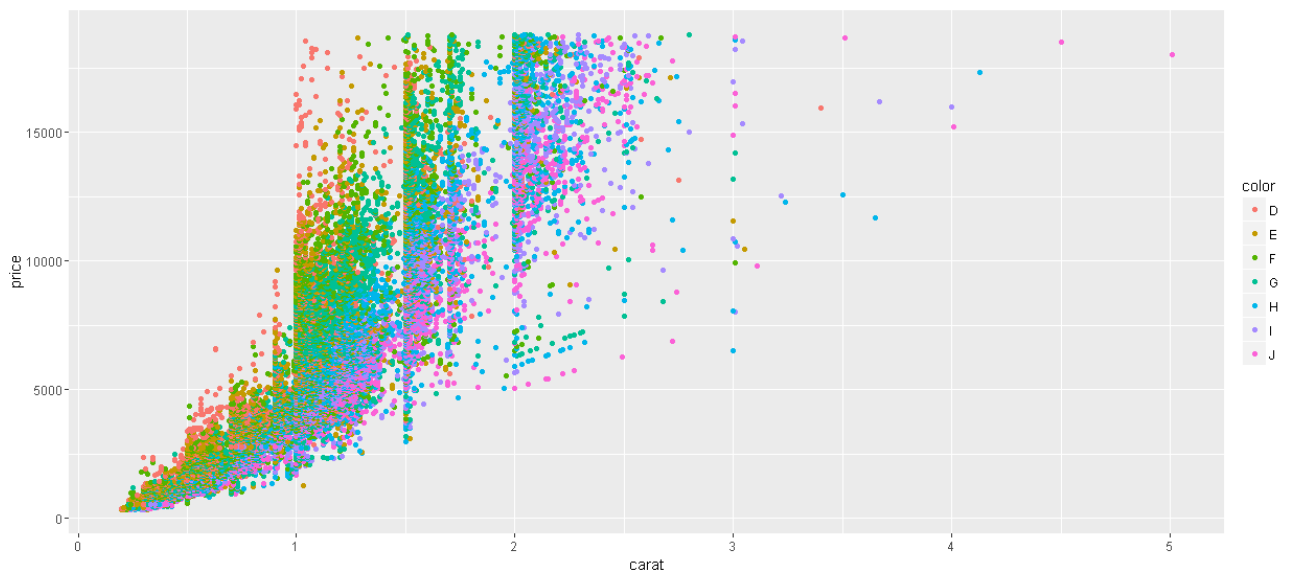


Siguiente capa son los puntos

```
graf + geom_point(aes(color=color))
```

```
graf
```

Y con esto estamos reflejando todos los puntos de la columna Color:



Para calcular la linea de regresión:

```
linea <- lm(price ~ carat, data=diamonds, aes(x=carat, y=price))
linea
```

```
> linea <- lm(price ~ carat, data=diamonds)
> linea

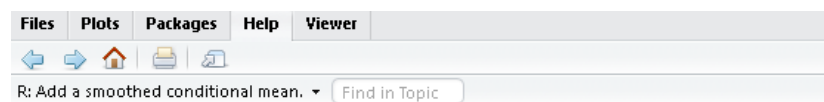
Call:
lm(formula = price ~ carat, data = diamonds)

Coefficients:
(Intercept)      carat 
    -2256      7756
```

Para añadirle la linea de valores de regresión se utiliza:

```
+ geom_smooth()
```

Como ni idea como se pone busco: [?geom_smooth](#)



Add a smoothed conditional mean.

Description

Aids the eye in seeing patterns in the presence of overplotting. `geom_smooth` and `stat_smooth` results with a non-standard geom.

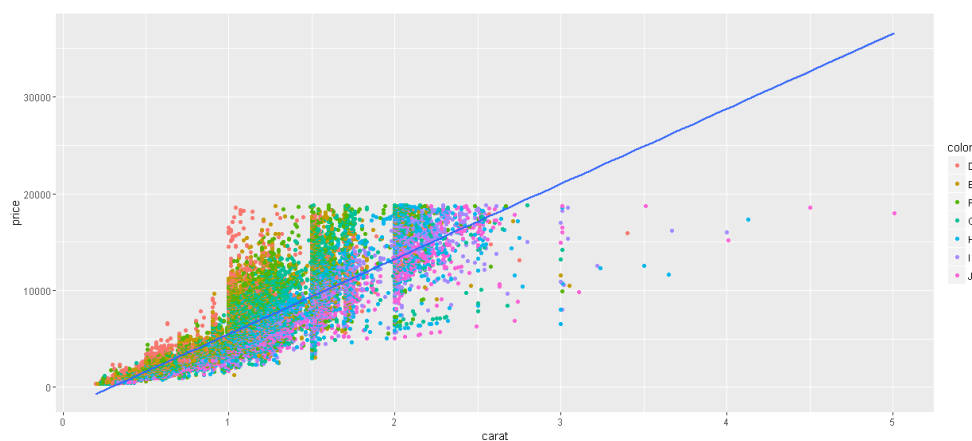
Usage

```
geom_smooth(mapping = NULL, data = NULL, stat = "smooth",
  position = "identity", ..., method = "auto", formula = y ~ x,
  se = TRUE, na.rm = FALSE, show.legend = NA, inherit.aes = TRUE)
```

Puesto todo junto nos sale:

```
graf <- ggplot (diamonds, aes(x=carat, y=price)) + geom_point(aes(color=color)) + geom_smooth(method =
lm, formula = y ~ x)
```

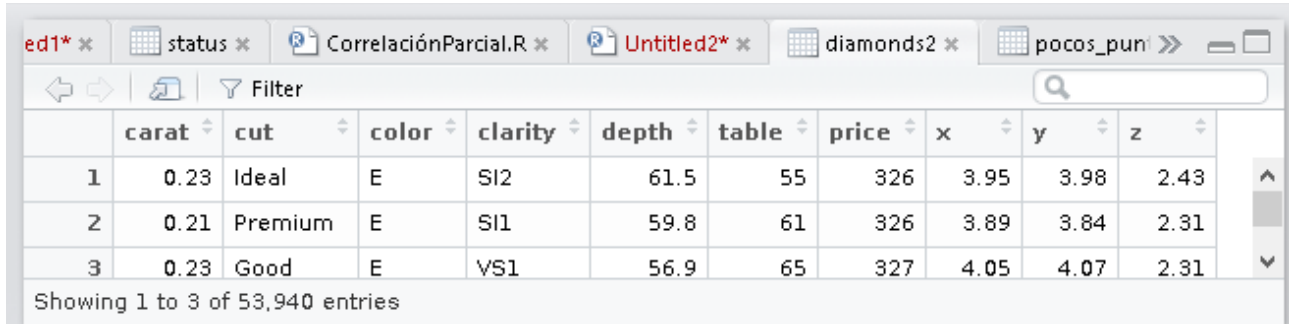
```
graf
```



El anterior era el grafico con todos lo puntos. Vvamoos a hacer uno con 111 puntos

Nos hacemos una copia de diamonds, que no queremos romper nada.

```
diamonds2 <- 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

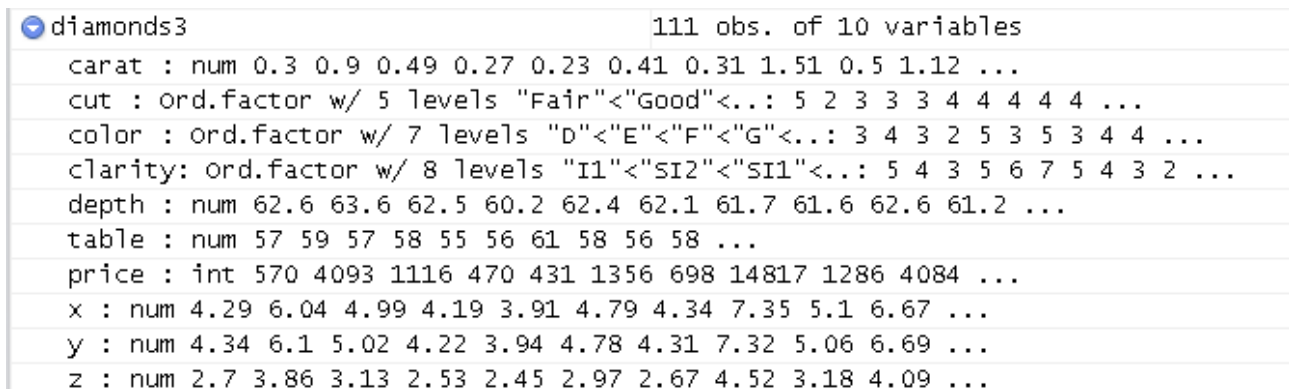
Showing 1 to 3 of 53,940 entries

Extraemos 111 aleatorios y generamos el data frame Diamonds3

```
diamonds3 <- diamonds2[sample(nrow(diamonds2), 111), ]
```

Lo comprobamos:

```
diamonds3
```



```

# A tibble: 111 x 10
#   carat cut      color clarity depth table price    x     y     z
#   <dbl> <ord> <ord> <ord>   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1  0.3   Fair    D      I1      62.6   57    570  4.29  4.34  2.7
2  0.9   Fair    D      I1      63.6   59   4093  6.04  6.1   3.86
3  0.49  Fair    D      I1      62.5   57   1116  4.99  5.02  3.13
4  0.27  Fair    D      I1      60.2   58    470  4.19  4.22  2.53
5  0.23  Fair    D      I1      62.4   55    431  3.91  3.94  2.45
6  0.41  Fair    D      I1      62.1   56   1356  4.79  4.78  2.97
7  0.31  Fair    D      I1      61.7   58    698  4.34  4.31  2.67
8  1.51  Fair    D      I1      61.6   56   14817  7.35  7.32  4.52
9  0.5   Fair    D      I1      62.6   58   1286  5.1   5.06  3.18
10 1.12  Fair    D      I1      61.2   58   4084  6.67  6.69  4.09
#> ...

```

Y después generamos un grafico con el dataframe de diamonds3

```
graf2 <- ggplot (diamonds3, aes(x=carat, y=price)) + geom_point(aes(color=color)) +  
geom_smooth(method = lm, formula = y ~ x)
```

```
graf2
```

