Untitled

2/17/2021

```
library(ggplot2)
library(pastecs)
library(psych)
library(DescTools)
```

Vaise usar para practicar os exemplos uns datos sobre pesos e prezos de diamantes, que é unha submostra doutra colección incluida no paquete *ggplot2()*

```
datos=read.csv2("exemplo.csv")

set.seed(2021)
D=rbinom(100,5,0.45)
X=rnorm(100,3*D/3,0.5)
Y=4*X+rnorm(100)
```

Sumarios

Obter unha colección de estatísticos descritivos sobre os nosos datos. Xa sexa unha ou varias variables, ou xa sexa global ou separados polos niveis dun atributo.

Colección de exemplos;Outro exemplo e outro mais

- summary()

Aplicado a un vector ou a un data.frame, cando son variables numéricas.

```
summary(datos)
```

```
##      carat           colour            clarity          certification
##  Min.   :0.1800   Length:96          Length:96          Length:96
##  1st Qu.:0.2600   Class :character   Class :character   Class :character
##  Median :0.5100   Mode  :character   Mode  :character   Mode  :character
##  Mean   :0.5170
##  3rd Qu.:0.7125
##  Max.   :1.0400
##      price
##  Min.   :  725
##  1st Qu.: 1246
##  Median : 3518
##  Mean   : 4251
##  3rd Qu.: 5894
##  Max.   :16008
```

```
summary(X)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -0.9012  1.4782  2.2335  2.3112  3.2004  6.2268
```

```
stat.desc(diamonds)#library(pastecs)
```

```
##                     carat cut color clarity        depth        table
## nbr.val      5.394000e+04  NA    NA      NA 5.394000e+04 5.394000e+04
## nbr.null     0.000000e+00  NA    NA      NA 0.000000e+00 0.000000e+00
## nbr.na       0.000000e+00  NA    NA      NA 0.000000e+00 0.000000e+00
## min          2.000000e-01  NA    NA      NA 4.300000e+01 4.300000e+01
## max          5.010000e+00  NA    NA      NA 7.900000e+01 9.500000e+01
## range        4.810000e+00  NA    NA      NA 3.600000e+01 5.200000e+01
## sum          4.304087e+04  NA    NA      NA 3.330763e+06 3.099240e+06
## median       7.000000e-01  NA    NA      NA 6.180000e+01 5.700000e+01
## mean         7.979397e-01  NA    NA      NA 6.174940e+01 5.745718e+01
## SE.mean      2.040954e-03  NA    NA      NA 6.168448e-03 9.621063e-03
## CI.mean.0.95 4.000286e-03  NA    NA      NA 1.209021e-02 1.885736e-02
## var          2.246867e-01  NA    NA      NA 2.052404e+00 4.992948e+00
## std.dev      4.740112e-01  NA    NA      NA 1.432621e+00 2.234491e+00
## coef.var     5.940439e-01  NA    NA      NA 2.320057e-02 3.888966e-02
##                     price            x            y            z
## nbr.val      5.394000e+04 5.394000e+04 5.394000e+04 5.394000e+04
## nbr.null     0.000000e+00 8.000000e+00 7.000000e+00 2.000000e+01
## nbr.na       0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
## min          3.260000e+02 0.000000e+00 0.000000e+00 0.000000e+00
## max          1.882300e+04 1.074000e+01 5.890000e+01 3.180000e+01
## range        1.849700e+04 1.074000e+01 5.890000e+01 3.180000e+01
## sum          2.121352e+08 3.091386e+05 3.093203e+05 1.908793e+05
## median       2.401000e+03 5.700000e+00 5.710000e+00 3.530000e+00
## mean         3.932800e+03 5.731157e+00 5.734526e+00 3.538734e+00
## SE.mean      1.717736e+01 4.829974e-03 4.917698e-03 3.038533e-03
## CI.mean.0.95 3.366776e+01 9.466787e-03 9.638727e-03 5.955549e-03
## var          1.591563e+07 1.258347e+00 1.304472e+00 4.980109e-01
## std.dev      3.989440e+03 1.121761e+00 1.142135e+00 7.056988e-01
## coef.var     1.014402e+00 1.957302e-01 1.991681e-01 1.994213e-01
```

```
describe(diamonds)#library(psych); Ollo!! cos atributos, que os numeriza
```

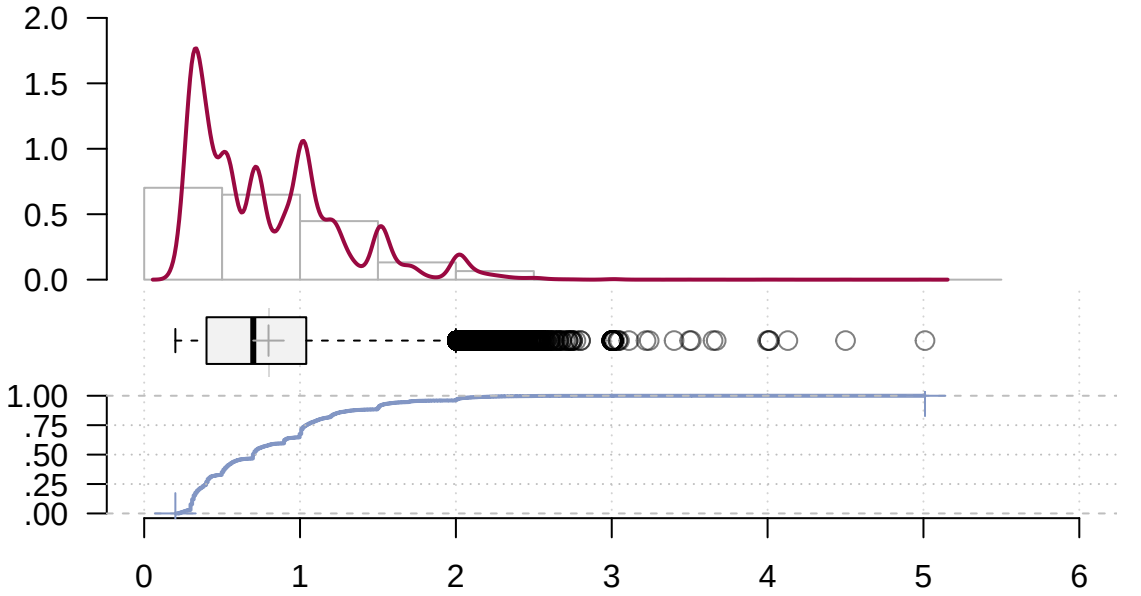
```
##          vars     n    mean      sd  median trimmed     mad   min      max
## carat       1 53940    0.80    0.47    0.70    0.73    0.47   0.2     5.01
## cut*        2 53940    3.90    1.12    4.00    4.04    1.48   1.0     5.00
## color*      3 53940    3.59    1.70    4.00    3.55    1.48   1.0     7.00
## clarity*    4 53940    4.05    1.65    4.00    3.91    1.48   1.0     8.00
## depth       5 53940   61.75    1.43   61.80   61.78    1.04  43.0    79.00
## table       6 53940   57.46    2.23   57.00   57.32    1.48  43.0    95.00
## price       7 53940 3932.80 3989.44 2401.00 3158.99 2475.94 326.0 18823.00
## x           8 53940    5.73    1.12    5.70    5.66    1.38   0.0    10.74
## y           9 53940    5.73    1.14    5.71    5.66    1.36   0.0    58.90
## z          10 53940    3.54    0.71    3.53    3.49    0.85   0.0    31.80
##             range  skew kurtosis    se
## carat        4.81  1.12     1.26  0.00
## cut*         4.00 -0.72    -0.40  0.00
## color*       6.00  0.19    -0.87  0.01
## clarity*     7.00  0.55    -0.39  0.01
## depth       36.00 -0.08     5.74  0.01
## table       52.00  0.80     2.80  0.01
## price    18497.00  1.62     2.18 17.18
```

```
## x           10.74  0.38    -0.62  0.00
## y           58.90  2.43    91.20  0.00
## z           31.80  1.52    47.08  0.00
```

```
Desc(diamonds)#library(DescTools) isto é un monstro
```

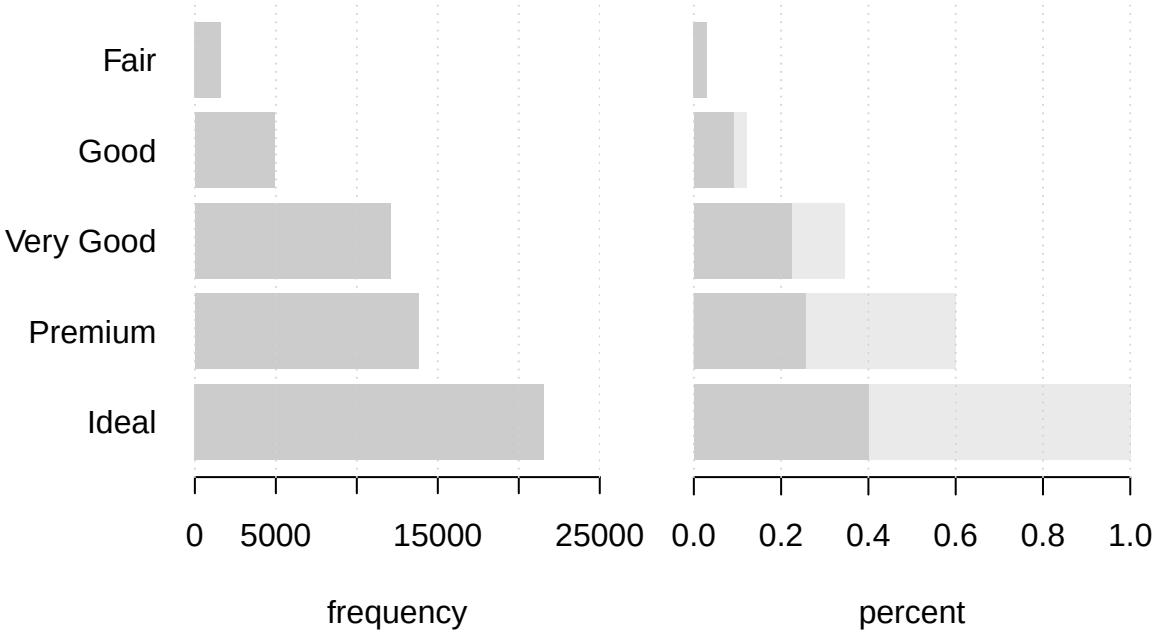
```
## ------------------------------------------------------------------------------
## Describe diamonds (tbl_df, tbl, data.frame):
##
## data frame:  53940 obs. of  10 variables
##      53940 complete cases (100.0%)
##
##   Nr  ColName  Class            NAs  Levels
##   1   carat    numeric          .
##   2   cut      ordered, factor  .    (5): 1-Fair, 2-Good, 3-Very Good,
##                                      4-Premium, 5-Ideal
##   3   color    ordered, factor  .    (7): 1-D, 2-E, 3-F, 4-G, 5-H, ...
##   4   clarity  ordered, factor  .    (8): 1-I1, 2-SI2, 3-SI1, 4-VS2, 5-VS1,
##                                      ...
##   5   depth    numeric          .
##   6   table    numeric          .
##   7   price    integer          .
##   8   x        numeric          .
##   9   y        numeric          .
##   10  z        numeric          .
##
##
## ------------------------------------------------------------------------------
## 1 - carat (numeric)
##
##   length       n    NAs  unique    0s  mean  meanCI'
##   53'940  53'940      0     273     0  0.80    0.79
##           100.0%   0.0%          0.0%          0.80
##
##      .05     .10    .25  median   .75   .90     .95
##     0.30    0.31   0.40    0.70  1.04  1.51    1.70
##
##    range      sd  vcoef     mad   IQR  skew    kurt
##     4.81    0.47   0.59    0.47  0.64  1.12    1.26
##
## lowest : 0.2 (12), 0.21 (9), 0.22 (5), 0.23 (293), 0.24 (254)
## highest: 4.0, 4.01 (2), 4.13, 4.5, 5.01
##
## ' 95%-CI (classic)
```

1 − carat (numeric)

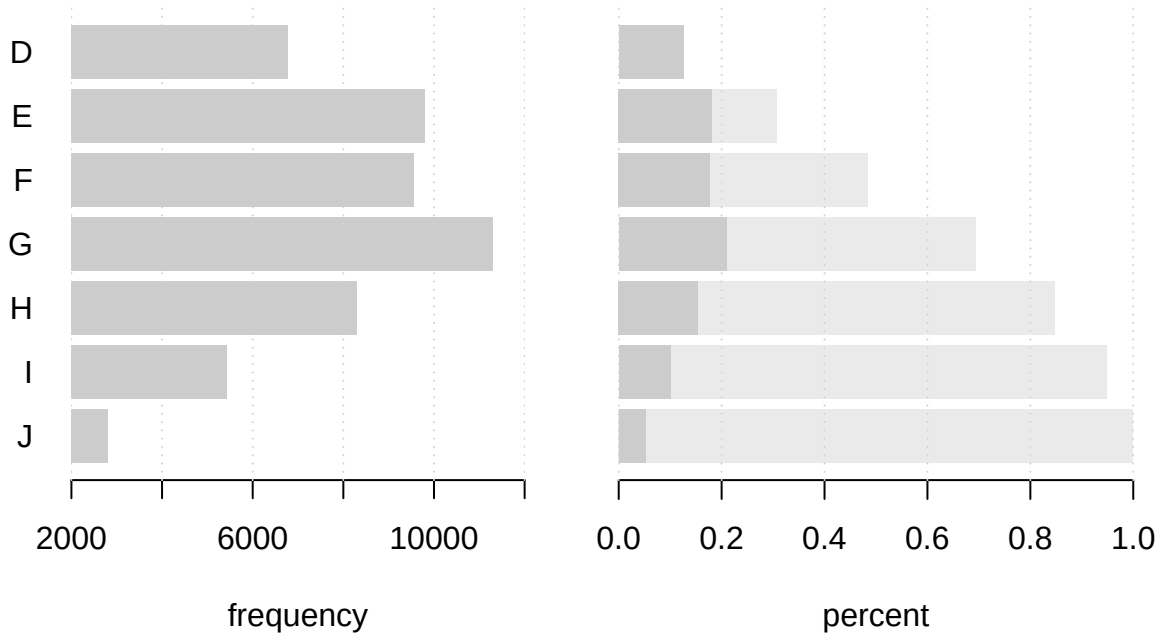



```
## ------------------------------------------------------------------------------
## 2 - cut (ordered, factor)
##
##   length      n    NAs unique levels  dupes
##   53'940 53'940      0      5      5      y
##          100.0%   0.0%
##
##        level    freq   perc  cumfreq  cumperc
## 1       Fair   1'610   3.0%    1'610     3.0%
## 2       Good   4'906   9.1%    6'516    12.1%
## 3  Very Good  12'082  22.4%   18'598    34.5%
## 4    Premium  13'791  25.6%   32'389    60.0%
## 5      Ideal  21'551  40.0%   53'940   100.0%
```

2 − cut (ordered, factor)

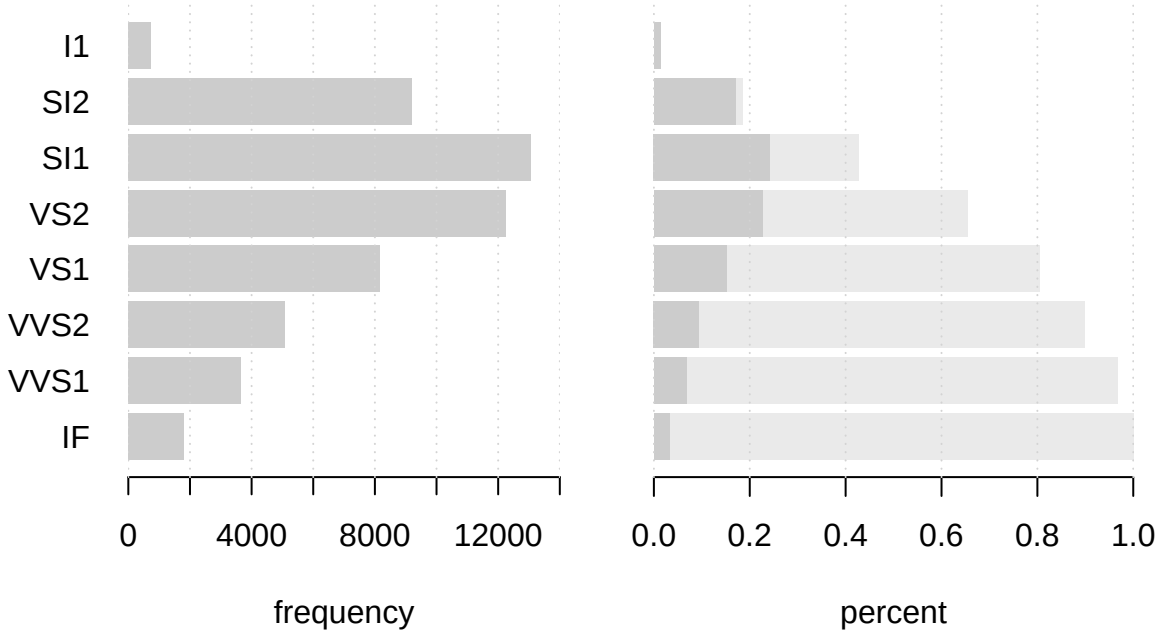



```
## ------------------------------------------------------------------------------
## 3 - color (ordered, factor)
##
##   length      n    NAs unique levels  dupes
##   53'940 53'940      0      7      7      y
##          100.0%   0.0%
##
##    level    freq   perc  cumfreq  cumperc
## 1      D   6'775  12.6%    6'775    12.6%
## 2      E   9'797  18.2%   16'572    30.7%
## 3      F   9'542  17.7%   26'114    48.4%
## 4      G  11'292  20.9%   37'406    69.3%
## 5      H   8'304  15.4%   45'710    84.7%
## 6      I   5'422  10.1%   51'132    94.8%
## 7      J   2'808   5.2%   53'940   100.0%
```



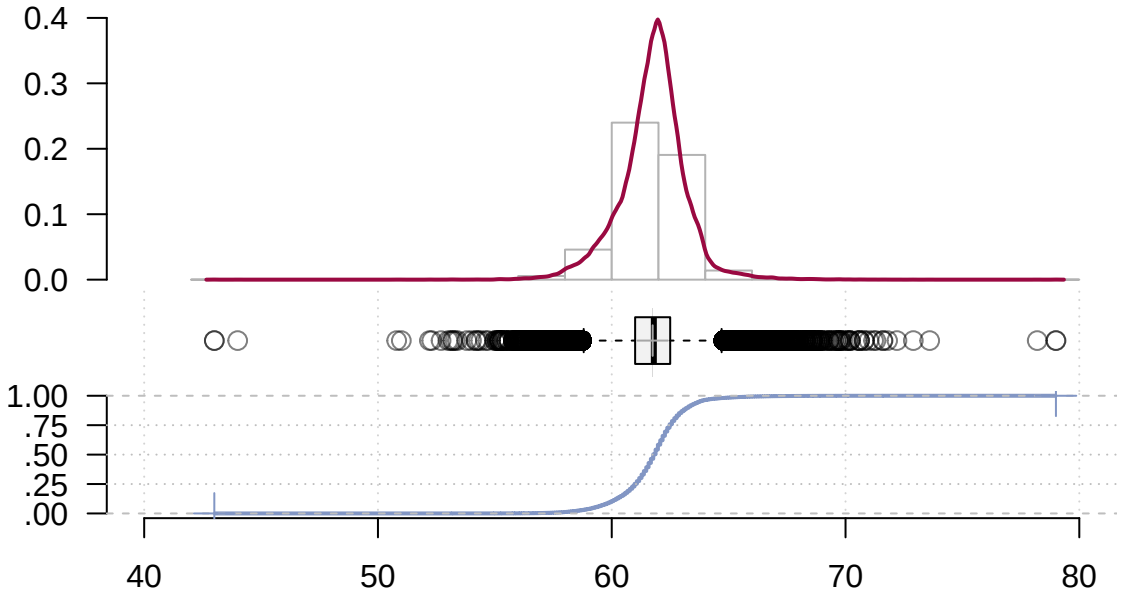

```
## ------------------------------------------------------------------------------
## 4 - clarity (ordered, factor)
##
##   length      n    NAs unique levels  dupes
##   53'940 53'940      0      8      8      y
##          100.0%   0.0%
##
##    level    freq   perc  cumfreq  cumperc
## 1     I1     741   1.4%      741     1.4%
## 2    SI2   9'194  17.0%    9'935    18.4%
## 3    SI1  13'065  24.2%   23'000    42.6%
## 4    VS2  12'258  22.7%   35'258    65.4%
## 5    VS1   8'171  15.1%   43'429    80.5%
## 6   VVS2   5'066   9.4%   48'495    89.9%
## 7   VVS1   3'655   6.8%   52'150    96.7%
## 8     IF   1'790   3.3%   53'940   100.0%
```

4 − clarity (ordered, factor)



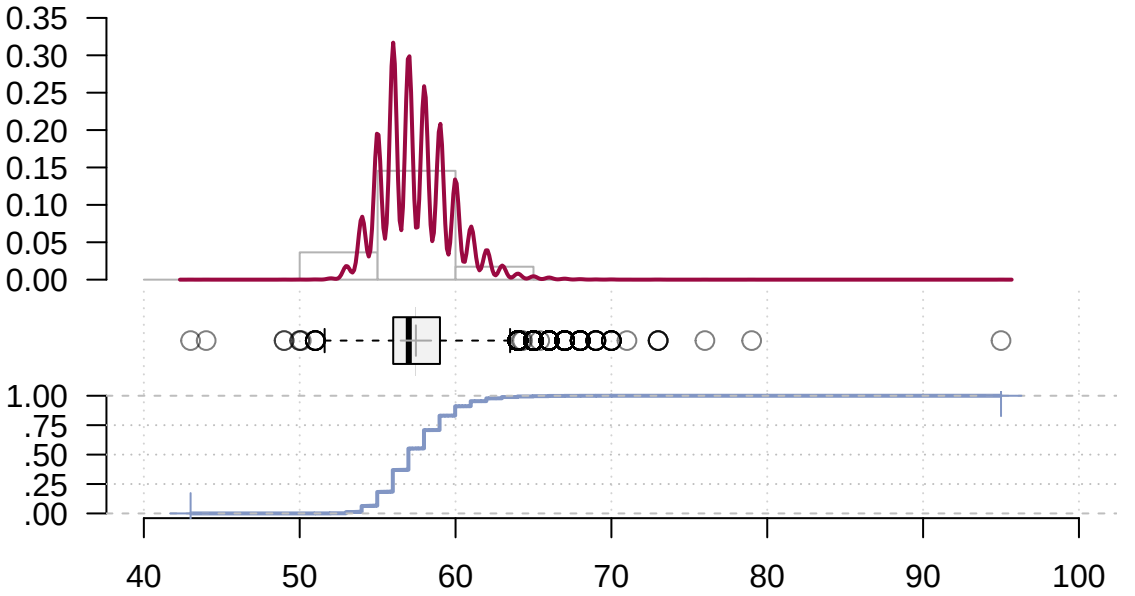

```
## ------------------------------------------------------------------------------
## 5 - depth (numeric)
##
##   length       n    NAs  unique     0s   mean  meanCI'
##   53'940  53'940      0     184      0  61.75   61.74
##           100.0%   0.0%           0.0%          61.76
##
##      .05     .10    .25  median    .75    .90     .95
##    59.30   60.00  61.00   61.80  62.50  63.30   63.80
##
##    range      sd  vcoef     mad    IQR   skew    kurt
##    36.00    1.43   0.02    1.04   1.50  -0.08    5.74
##
## lowest : 43.0 (2), 44.0, 50.8, 51.0, 52.2
## highest: 72.2, 72.9, 73.6, 78.2, 79.0 (2)
##
## ' 95%-CI (classic)
```

5 − depth (numeric)



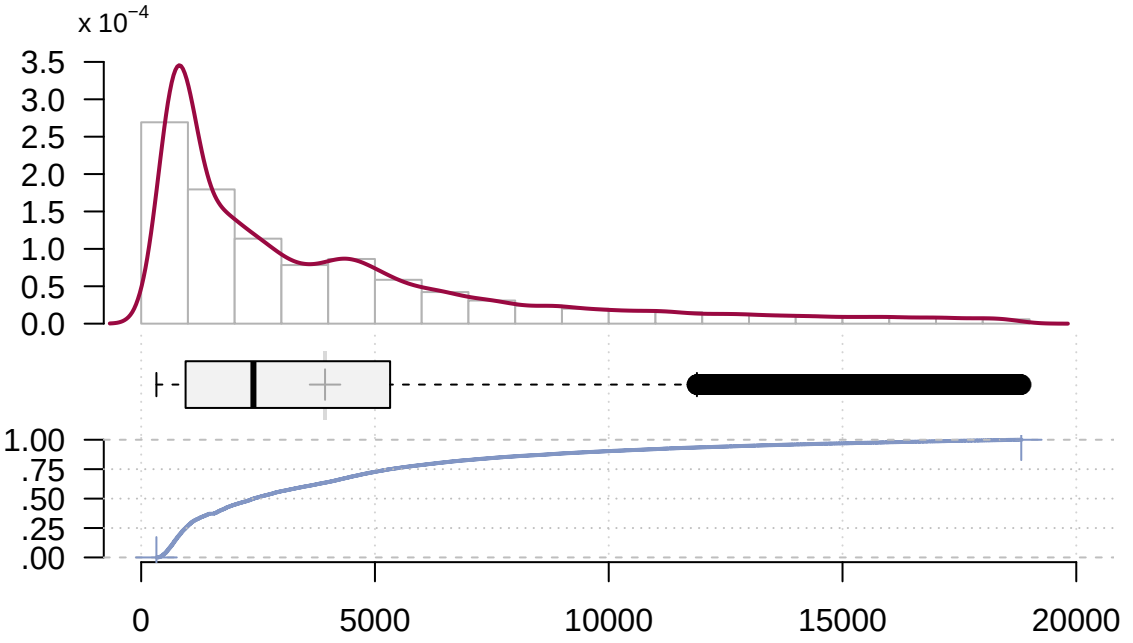

```
## ------------------------------------------------------------------------------
## 6 - table (numeric)
##
##   length       n    NAs  unique     0s   mean  meanCI'
##   53'940  53'940      0     127      0  57.46   57.44
##           100.0%   0.0%          0.0%          57.48
##
##      .05     .10    .25  median    .75    .90     .95
##    54.00   55.00  56.00   57.00  59.00  60.00   61.00
##
##    range      sd  vcoef     mad    IQR   skew    kurt
##    52.00    2.23   0.04    1.48   3.00   0.80    2.80
##
## lowest : 43.0, 44.0, 49.0 (2), 50.0 (2), 50.1
## highest: 71.0, 73.0 (4), 76.0, 79.0, 95.0
##
## heap(?): remarkable frequency (18.3%) for the mode(s) (= 56)
##
## ' 95%-CI (classic)
```

6 − table (numeric)



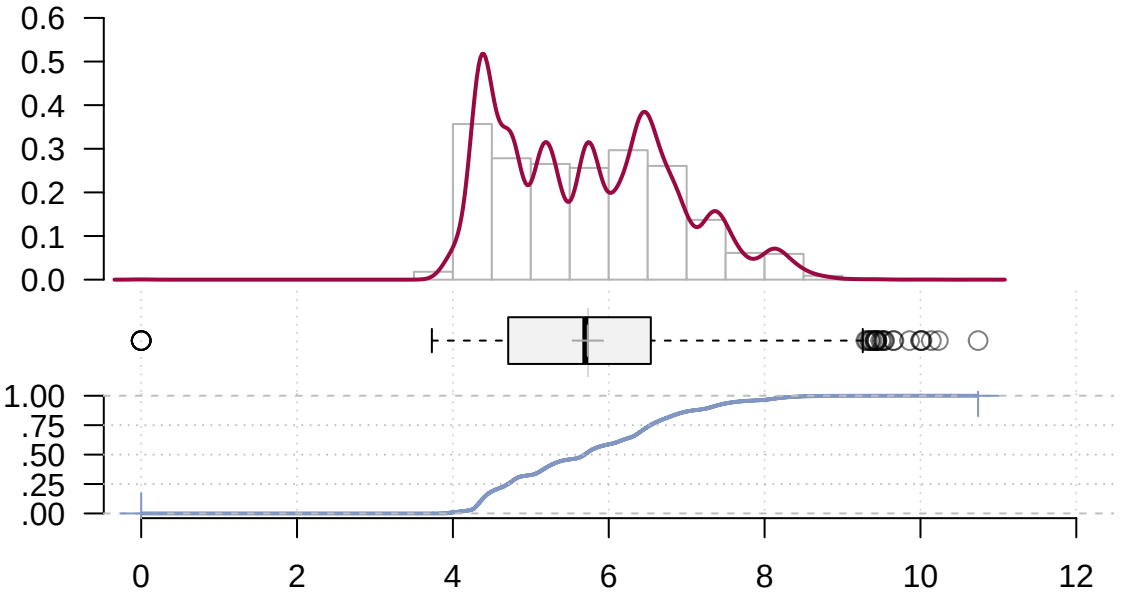

```
## ------------------------------------------------------------------------------
## 7 - price (integer)
##
##      length         n      NAs    unique        0s      mean     meanCI'
##      53'940    53'940        0    11'602         0  3'932.80   3'899.13
##                100.0%     0.0%                0.0%             3'966.47
##
##         .05       .10      .25    median       .75       .90        .95
##      544.00    646.00   950.00  2'401.00  5'324.25  9'821.00  13'107.10
##
##       range        sd    vcoef       mad       IQR      skew       kurt
##   18'497.00  3'989.44     1.01  2'475.94  4'374.25      1.62       2.18
##
## lowest : 326 (2), 327, 334, 335, 336 (2)
## highest: 18'803, 18'804, 18'806, 18'818, 18'823
##
## ' 95%-CI (classic)
```

7 − price (integer)



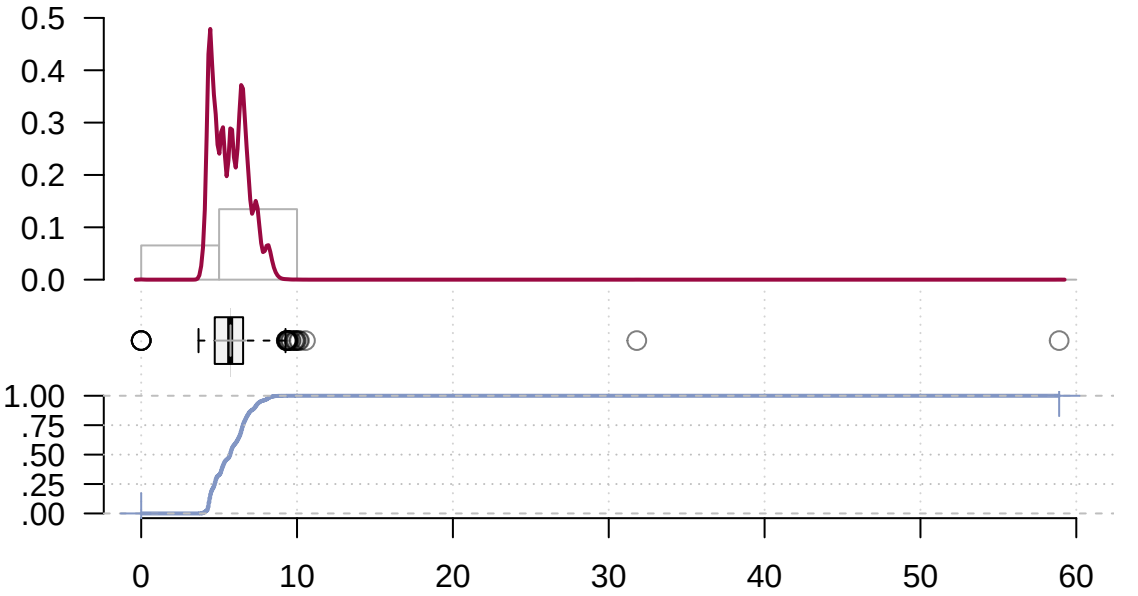

```
## ------------------------------------------------------------------------------
## 8 - x (numeric)
##
##   length       n    NAs  unique     0s  mean  meanCI'
##   53'940  53'940      0     554      8  5.73    5.72
##           100.0%   0.0%          0.0%          5.74
##
##      .05     .10    .25  median   .75   .90     .95
##     4.29    4.36   4.71    5.70  6.54  7.31    7.66
##
##    range      sd  vcoef     mad   IQR  skew    kurt
##    10.74    1.12   0.20    1.38  1.83  0.38   -0.62
##
## lowest : 0.0 (8), 3.73 (2), 3.74, 3.76, 3.77
## highest: 10.01, 10.02, 10.14, 10.23, 10.74
##
## ' 95%-CI (classic)
```

8 − x (numeric)



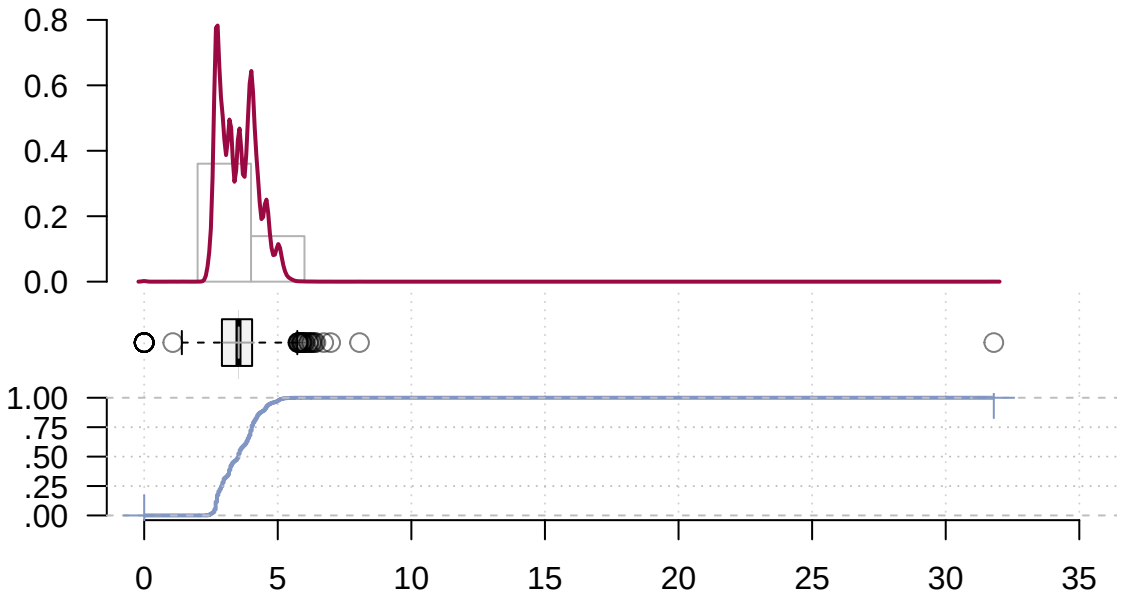

```
## ------------------------------------------------------------------------------
## 9 - y (numeric)
##
##   length       n    NAs  unique     0s  mean  meanCI'
##   53'940  53'940      0     552      7  5.73    5.72
##           100.0%   0.0%          0.0%          5.74
##
##      .05     .10    .25  median   .75   .90     .95
##     4.30    4.36   4.72    5.71  6.54  7.30    7.65
##
##    range      sd  vcoef     mad   IQR  skew    kurt
##    58.90    1.14   0.20    1.36  1.82  2.43   91.20
##
## lowest : 0.0 (7), 3.68, 3.71 (2), 3.72, 3.73
## highest: 10.1, 10.16, 10.54, 31.8, 58.9
##
## ' 95%-CI (classic)
```

9 − y (numeric)



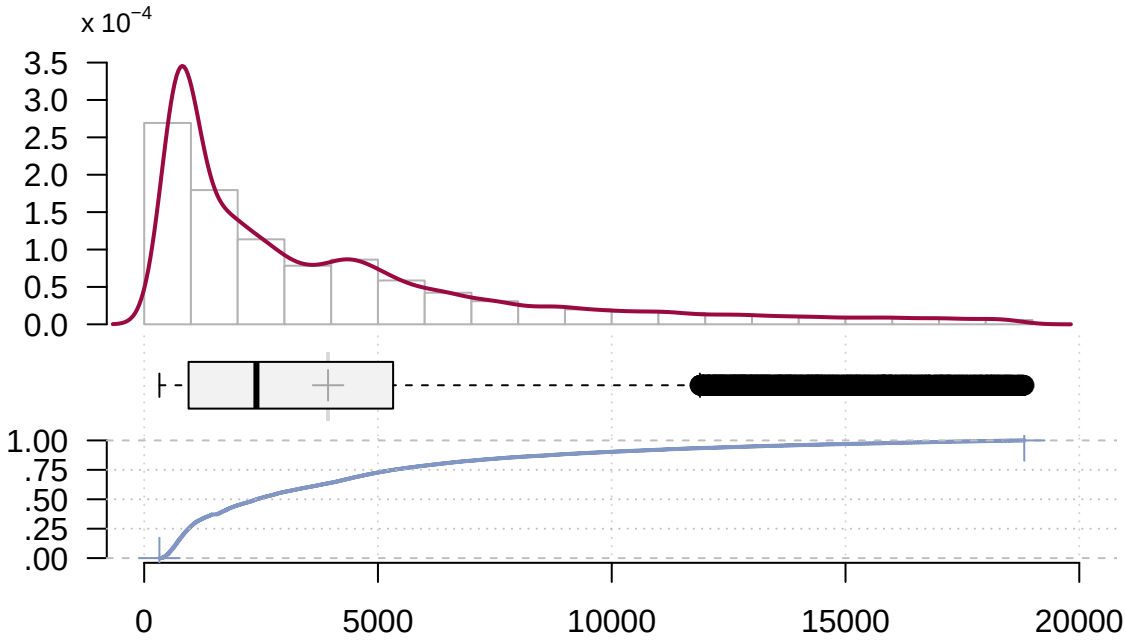

```
## ------------------------------------------------------------------------------
## 10 - z (numeric)
##
##   length       n    NAs  unique    0s  mean  meanCI'
##   53'940  53'940      0     375    20  3.54    3.53
##           100.0%   0.0%          0.0%          3.54
##
##      .05     .10    .25  median   .75   .90     .95
##     2.65    2.69   2.91    3.53  4.04  4.52    4.73
##
##    range      sd  vcoef     mad   IQR  skew    kurt
##    31.80    0.71   0.20    0.85  1.13  1.52   47.08
##
## lowest : 0.0 (20), 1.07, 1.41, 1.53, 2.06
## highest: 6.43, 6.72, 6.98, 8.06, 31.8
##
## ' 95%-CI (classic)
```

10 − z (numeric)




```
Desc(diamonds$price)
```

```
## ------------------------------------------------------------------------------
## diamonds$price (integer)
##
##      length         n      NAs    unique        0s      mean     meanCI'
##      53'940    53'940        0    11'602         0  3'932.80   3'899.13
##                100.0%     0.0%                0.0%             3'966.47
##
##         .05       .10      .25    median       .75       .90        .95
##      544.00    646.00   950.00  2'401.00  5'324.25  9'821.00  13'107.10
##
##       range        sd    vcoef       mad       IQR      skew       kurt
##   18'497.00  3'989.44     1.01  2'475.94  4'374.25      1.62       2.18
##
## lowest : 326 (2), 327, 334, 335, 336 (2)
## highest: 18'803, 18'804, 18'806, 18'818, 18'823
##
## ' 95%-CI (classic)
```



Sumarios por grupo

- by()

```
by(X,D,mean)
```

```
## D: 0
## [1] -0.2504651
## ------------------------------------------------------------
## D: 1
## [1] 0.9304218
## ------------------------------------------------------------
## D: 2
## [1] 1.89653
## ------------------------------------------------------------
## D: 3
## [1] 2.93481
## ------------------------------------------------------------
## D: 4
## [1] 3.694099
## ------------------------------------------------------------
## D: 5
## [1] 5.217365
```

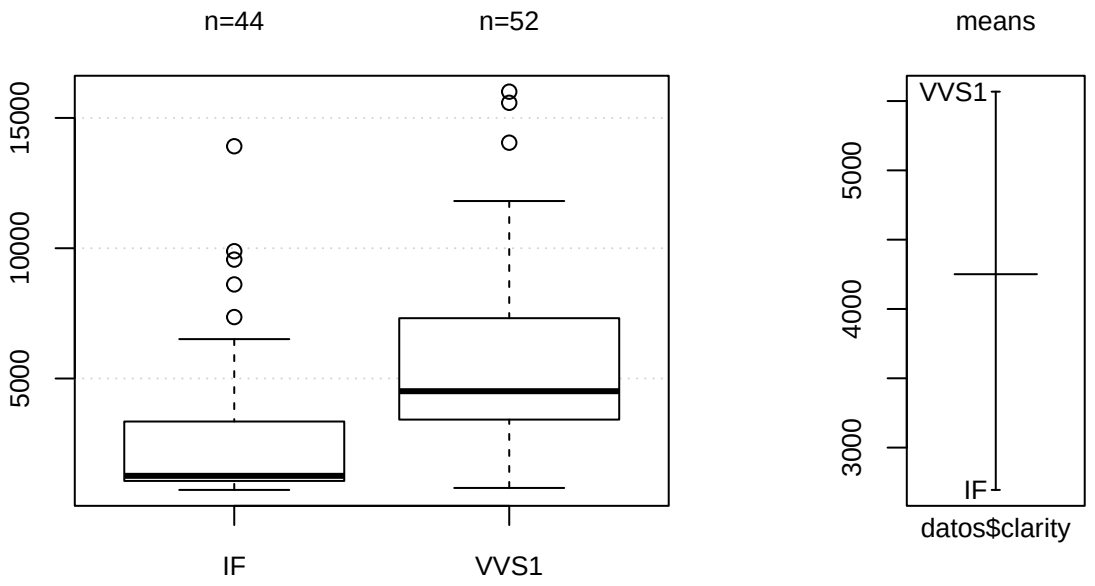
```
by(diamonds$price,diamonds$clarity,mean)
```

```
## diamonds$clarity: I1
## [1] 3924.169
## ------------------------------------------------------------
## diamonds$clarity: SI2
## [1] 5063.029
## ------------------------------------------------------------
```

```
## diamonds$clarity: SI1
## [1] 3996.001
## ------------------------------------------------------------
## diamonds$clarity: VS2
## [1] 3924.989
## ------------------------------------------------------------
## diamonds$clarity: VS1
## [1] 3839.455
## ------------------------------------------------------------
## diamonds$clarity: VVS2
## [1] 3283.737
## ------------------------------------------------------------
## diamonds$clarity: VVS1
## [1] 2523.115
## ------------------------------------------------------------
## diamonds$clarity: IF
## [1] 2864.839
```

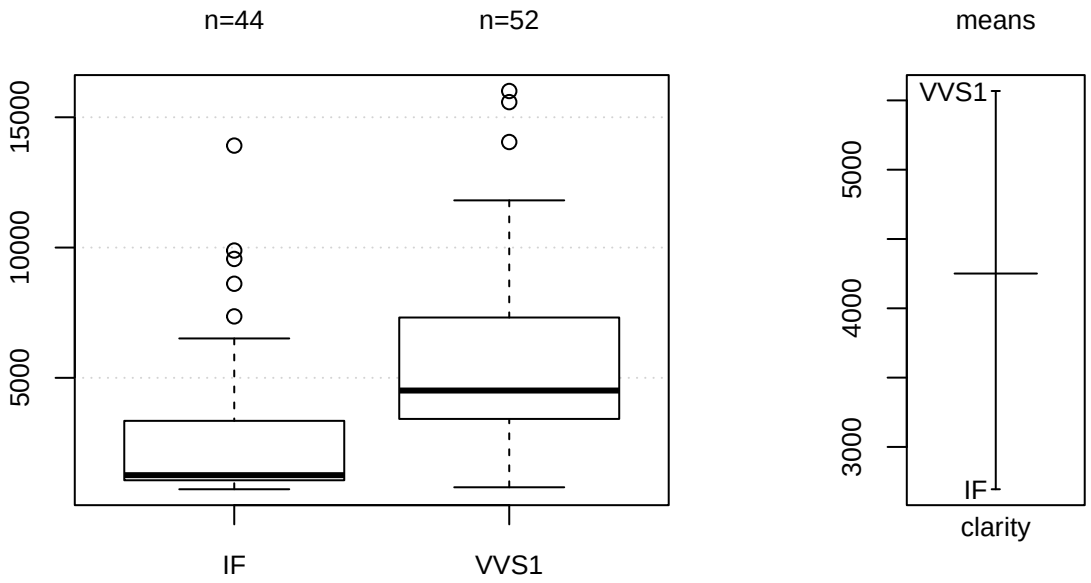
```
# library(DescTools)
Desc(datos$price~datos$clarity)
```

```
## ------------------------------------------------------------------------------
## datos$price ~ datos$clarity
##
## Summary:
## n pairs: 96, valid: 96 (100.0%), missings: 0 (0.0%), groups: 2
##
##
##                IF       VVS1
## mean    2'694.773  5'567.635
## median  1'265.500  4'513.500
## sd      3'016.061  3'673.286
## IQR     2'146.750  3'689.250
## n              44         52
## np        45.833%    54.167%
## NAs             0          0
## 0s              0          0
##
## Kruskal-Wallis rank sum test:
##   Kruskal-Wallis chi-squared = 23.697, df = 1, p-value = 1.128e-06
```

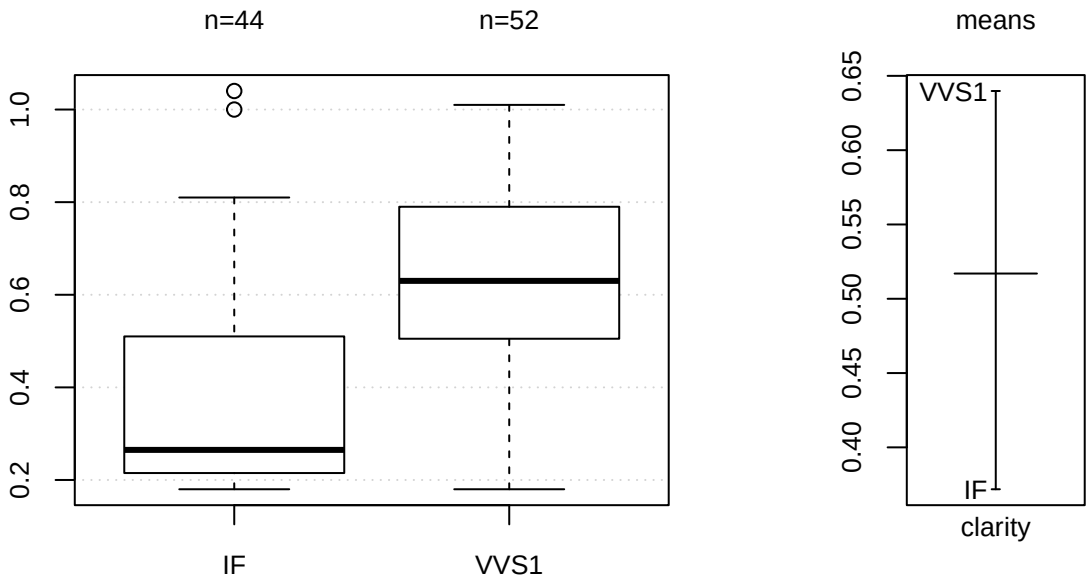



```
Desc(price+carat~clarity,data=datos)
```

```
## ------------------------------------------------------------------------------
## price ~ clarity (datos)
##
## Summary:
## n pairs: 96, valid: 96 (100.0%), missings: 0 (0.0%), groups: 2
##
##
##                IF       VVS1
## mean    2'694.773  5'567.635
## median  1'265.500  4'513.500
## sd      3'016.061  3'673.286
## IQR     2'146.750  3'689.250
## n              44         52
## np        45.833%    54.167%
## NAs             0          0
## 0s              0          0
##
## Kruskal-Wallis rank sum test:
##   Kruskal-Wallis chi-squared = 23.697, df = 1, p-value = 1.128e-06
```




```
## ------------------------------------------------------------------------------
## carat ~ clarity (datos)
##
## Summary:
## n pairs: 96, valid: 96 (100.0%), missings: 0 (0.0%), groups: 2
##
##
##               IF     VVS1
## mean       0.372    0.640
## median     0.265    0.630
## sd         0.232    0.243
## IQR        0.287    0.272
## n             44       52
## np        45.833%  54.167%
## NAs            0        0
## 0s             0        0
##
## Kruskal-Wallis rank sum test:
##   Kruskal-Wallis chi-squared = 24.121, df = 1, p-value = 9.048e-07
```



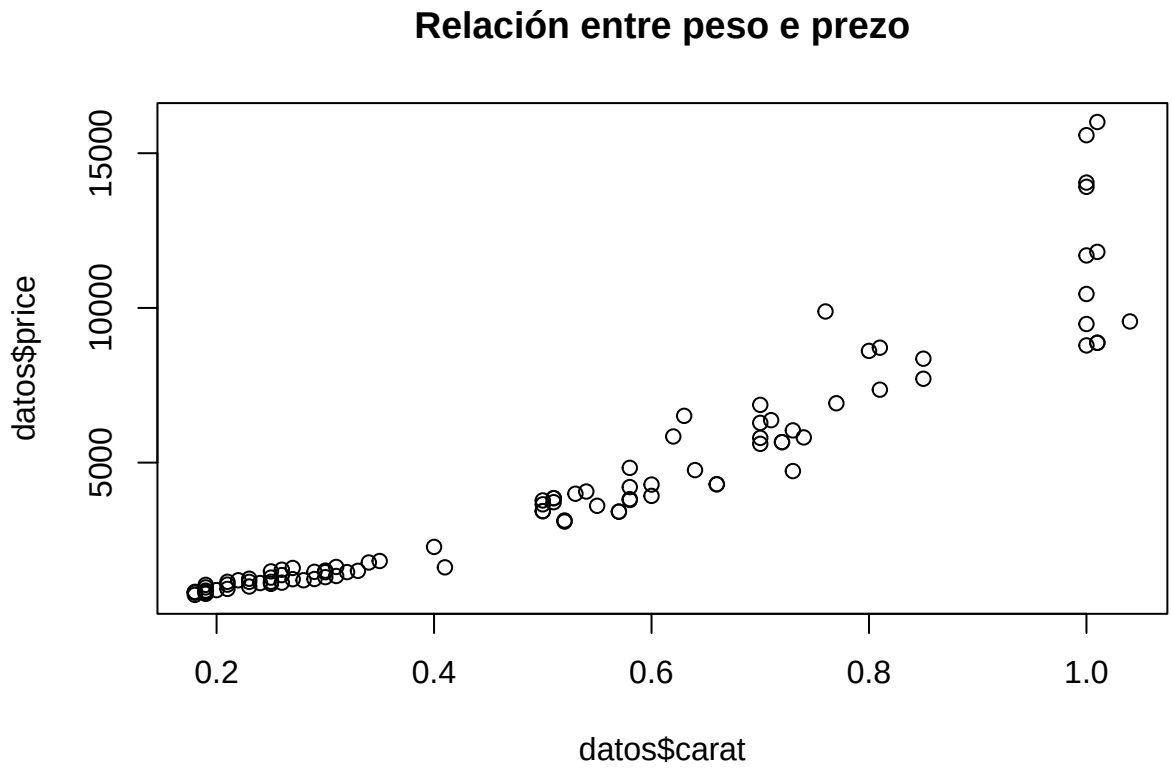

```
# library(psych)
describeBy(datos$price,datos$clarity)#funciona mal en Rmd
```

```
##
##  Descriptive statistics by group
## group: IF
##    vars  n    mean      sd median trimmed    mad min   max range skew kurtosis
## X1    1 44 2694.77 3016.06 1265.5 2041.92 424.02 725 13913 13188 1.99     3.33
##        se
## X1 454.69
## ------------------------------------------------------------
## group: VVS1
##    vars  n    mean      sd median trimmed     mad min   max range skew kurtosis
## X1    1 52 5567.63 3673.29 4513.5 5120.45 2445.55 800 16008 15208 1.05     0.68
##        se
## X1 509.39
```

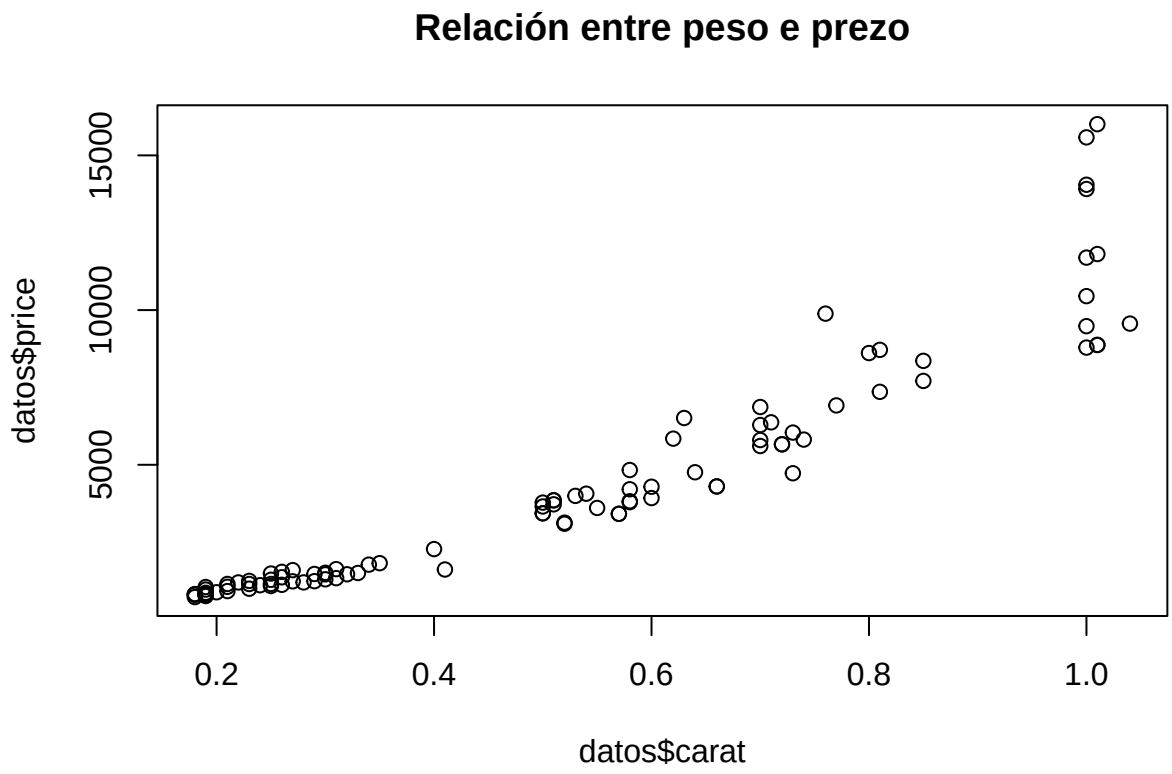
graficas en R

Gráficas en R base

```
plot(datos$carat,datos$price)
#engade o título
title(main = "Relación entre peso e prezo")
```




```
plot(datos$carat,datos$price,main = "Relación entre peso e prezo")
```

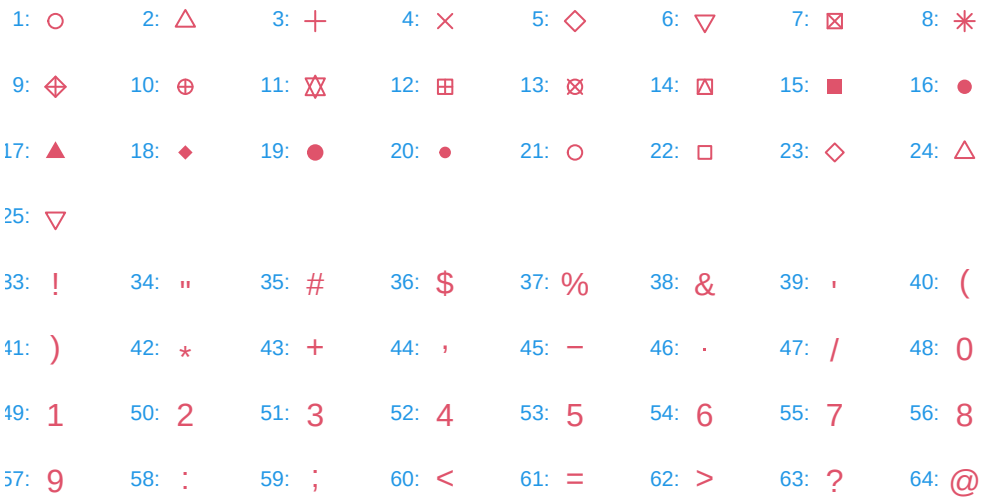


Algúns parámetros gráficos

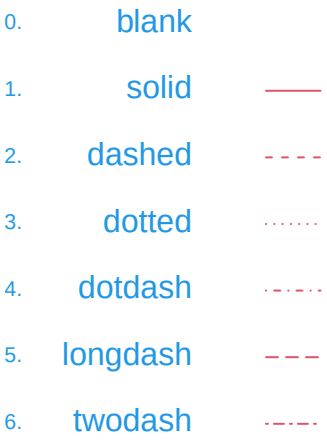
Parámetros dos gráficos elementais

Os seguintes son algúns dos parámetros que se manejan cos diferentes gráficos elementais de R, cambian diferentes aspectos, como poden ser as cores ou os títulos dos eixes. Máis información sobre eles aparece buscando a *axuda* dos comandos, se ben en algun caso esta axuda pode redirixirnos á axuda do comando *par()*.

- pch: simbolo para crear puntos, por exemplo en diagramas de dispersión.



- lty: Tipo de liña, (sólida, de puntos, de raias, . . .)



- lwd: O ancho de liña
- col: cores aplicadas no gráfico.

Sobre a utilización de cores en R hai exemplos e información na web **R Graphics Gallery**:http://research.stowers-institute.org/efg/R/index.htm, na sección ***Color***

- main: Título para o gráfico
- xlab: Título para o eixe das X
- ylab: Título para o eixe das Y

gardar graficos no disco duro Un exemplo de uso é o seguinte, que garda o gráfico nun ficheiro con formato *png*. Con este código o gráfico debúxase directamente no ficheiro.

```
#Este primeiro comando abre un 'device' png, ou sexa, o dispositivo que manexa os gráficos
#vai escribilos directamente nun ficheiro dese tipo.
#ademais faino cun fondo transparente (bg="transparent")
```

```
#ancho de 1024 pixels e altura de 768
png(file = "myplot.png", bg = "transparent", width = 1024, height = 768)
barplot(c(3,1,1,4)) #realiza o gráfico, coma sempre
dev.off() #pecha o "device", agora os gráficos apareceran onda sempre
```

Usando ese sistema é necesario pechar o dispositivo para que se vexa o gráfico. Ademais, calquera outro gráfico que se realizase iría a parar ao mesmo sitio, co que eliminaría o xa feito.

Outra cuestión a ter en conta é o nome que se asigna: R crea un gráfico con ese nome, e non pregunta para borrar calquera outro que exista e que xa use ese mesmo nome.

Por esas cuestions habería que abrir e pechar o *device* para cada novo gráfico, cambiando o nome para cada ocasión.

Outra posibilidade é pedirlle que vaia numerando os ficheiros graficos a medida que os crea. Iso faise incluindo no nome do ficheiro a expresión *%d*:

```
png(file = "myplot%d.png")
barplot(c(3,1,1,4)) #realiza o gráfico myplot1.png
hist(rnorm(100))#realiza o gráfico myplot2.png
plot(rnorm(20),runif(20))#realiza o gráfico myplot3.png
dev.off() #pecha o "device", agora os gráficos apareceran onda sempre
```

Gráficos: ggplot2

```
###########ggplot2
#comézase cargando o paquete na memoria:
library(ggplot2)
datos=read.csv2("exemplo.csv")
```

O concepto que manexa é construir o gráfico sumando elementos. Hai varios tipos de elementos, pero aquí comentaremos só 5

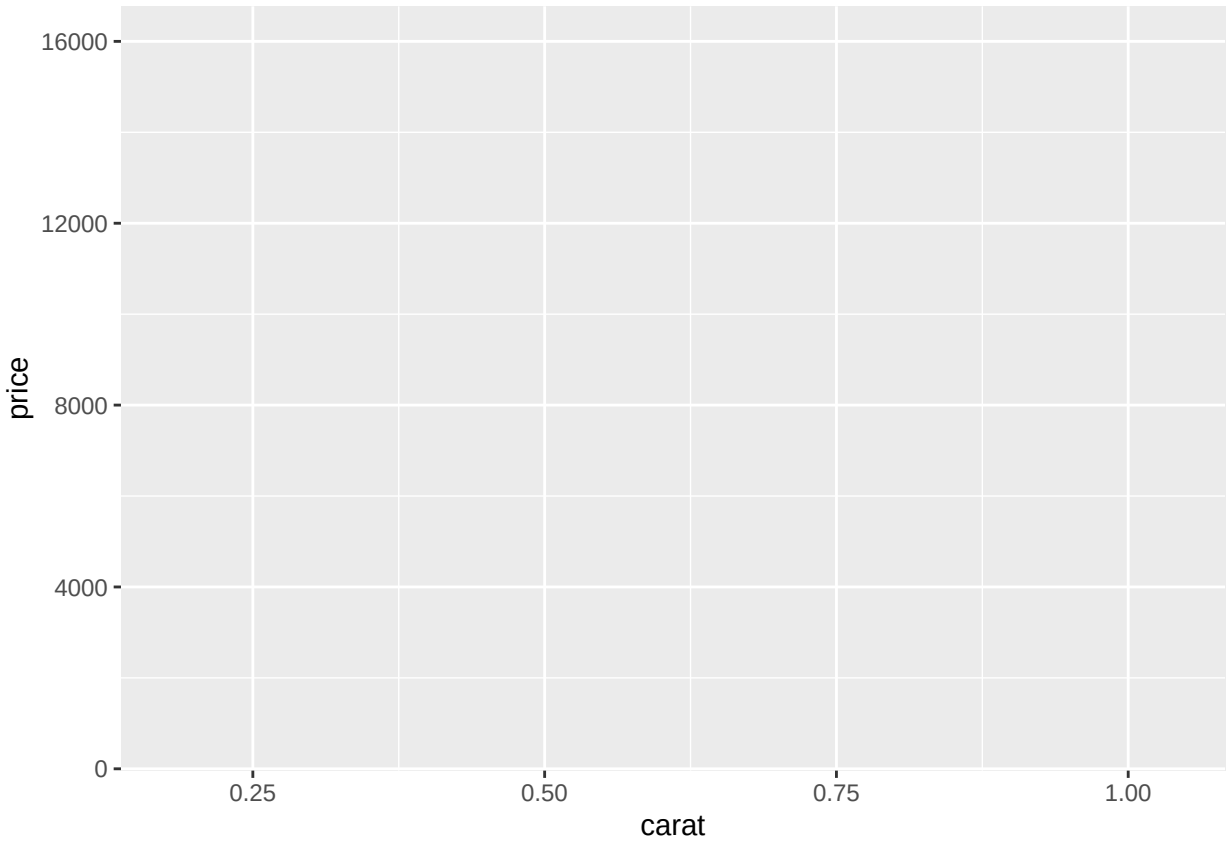
- datos:

o data.frame coas variables que queremos representar graficamente. **É obrigatorio que teñan estructura de data.frame**

```
ggplot(datos) #con isto non fai nada, posto que non se lle dixo que se quere facer
```

- esteticas: aes() indica diferentes aspectos que apareceran no grafico, entre eles as variables que imos representar

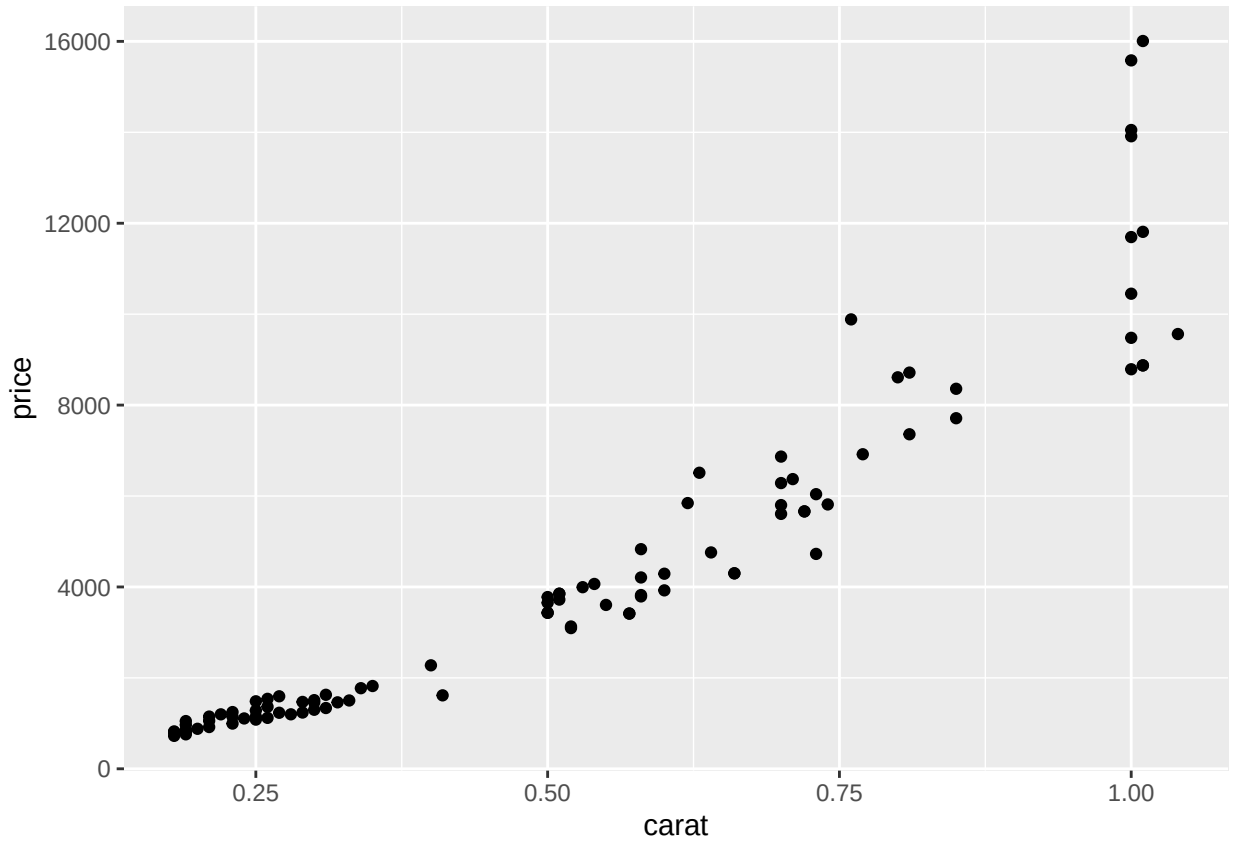
```
ggplot(datos,aes(x=carat,y=price))
```



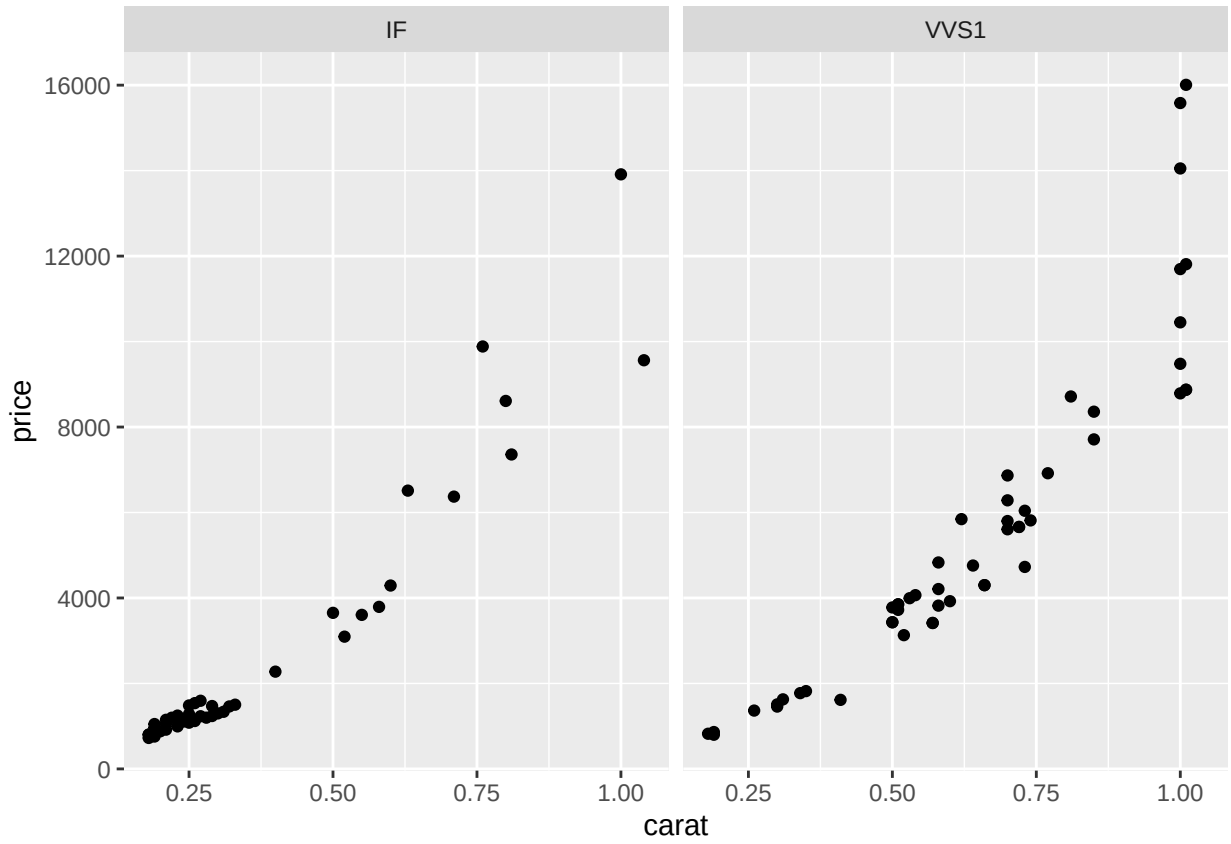
#capas:

precisamente a parte que di que grafico se emprega

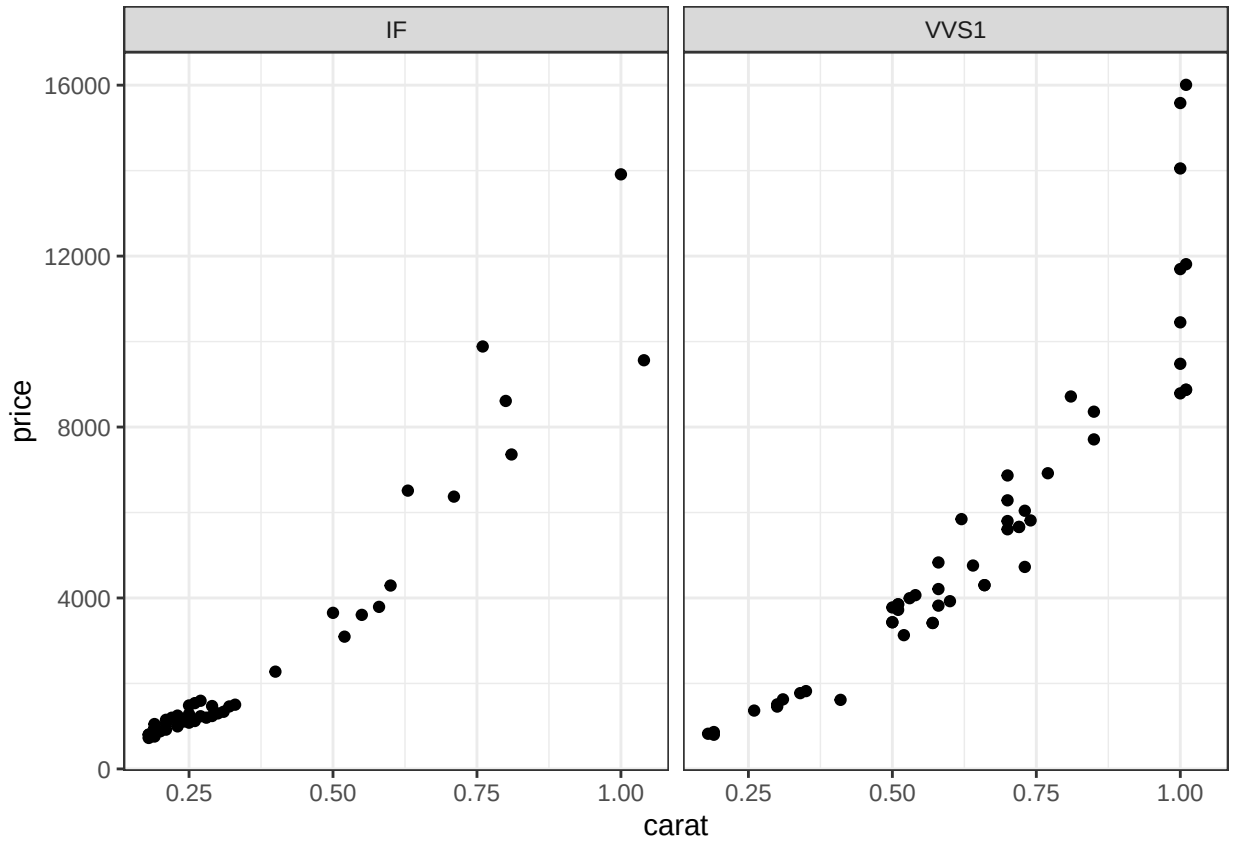
```
ggplot(datos,aes(x=carat,y=price))+geom_point()#geom_point é a capa para un diagrama de dispersion
```



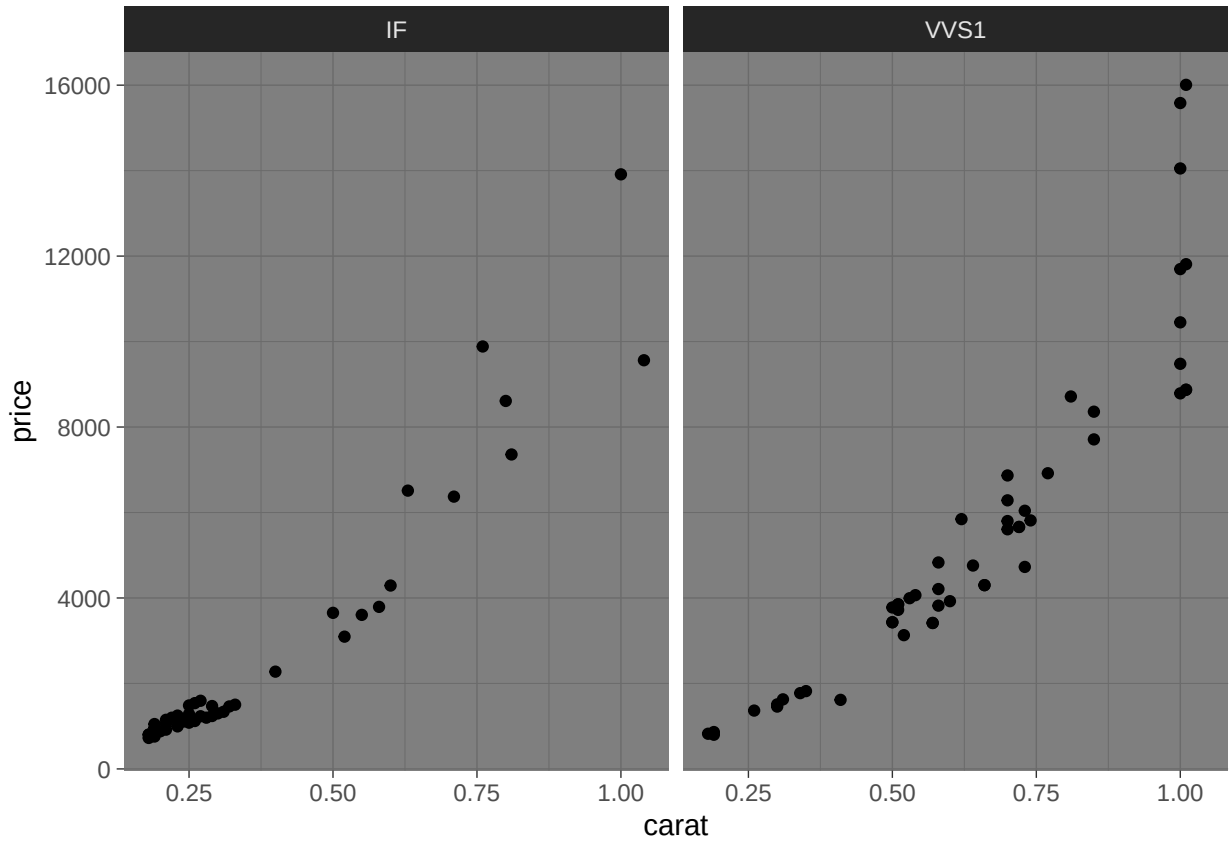

```
#facetas: permite facer os graficos separados polos niveis de 1 ou mais atributos
ggplot(datos,aes(x=carat,y=price))+geom_point()+facet_grid(~clarity)
```



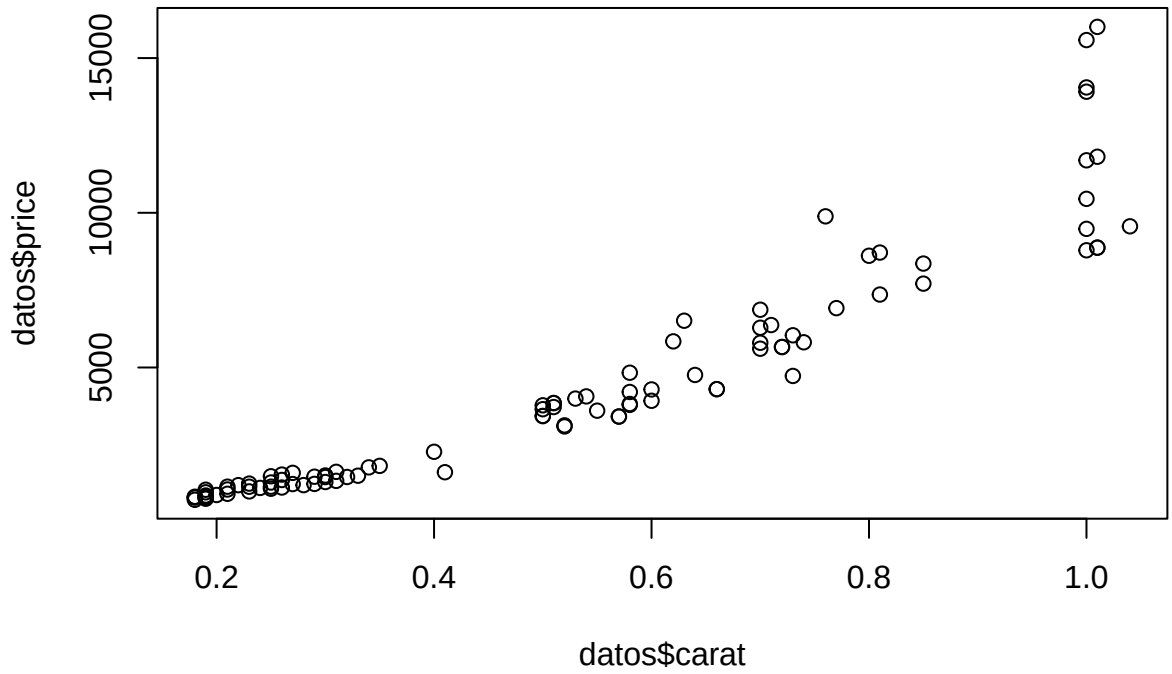

```
#temas: permiten variar o aspecto dos gráficos,usando motivos predefinidos.
#tedes exemplos na chuleta.6.ggplot2.r
ggplot(datos,aes(x=carat,y=price))+geom_point()+facet_grid(~clarity)+theme_bw()
```




```
ggplot(datos,aes(x=carat,y=price))+geom_point()+facet_grid(~clarity)+theme_dark()
```

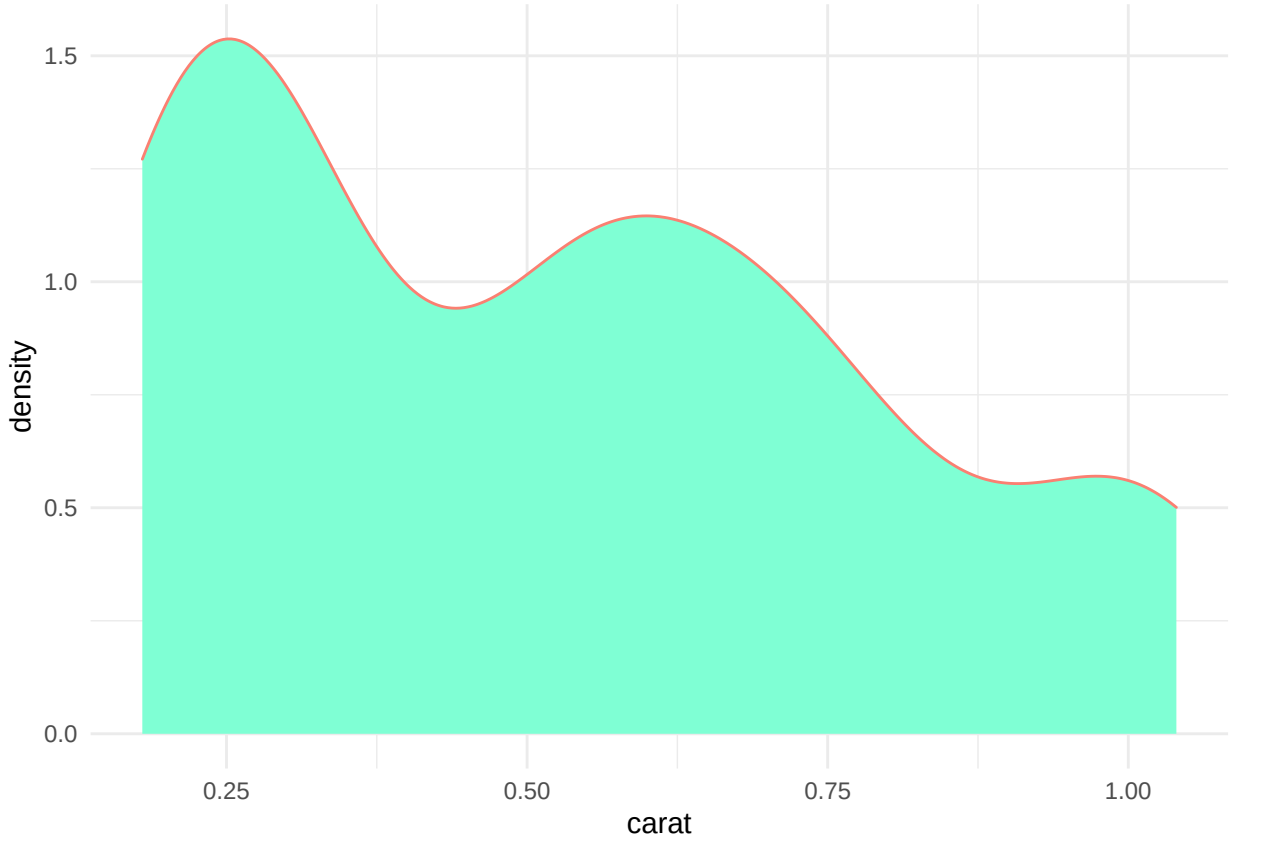



```
# ¢omparación co plot básico
plot(datos$carat,datos$price)
```

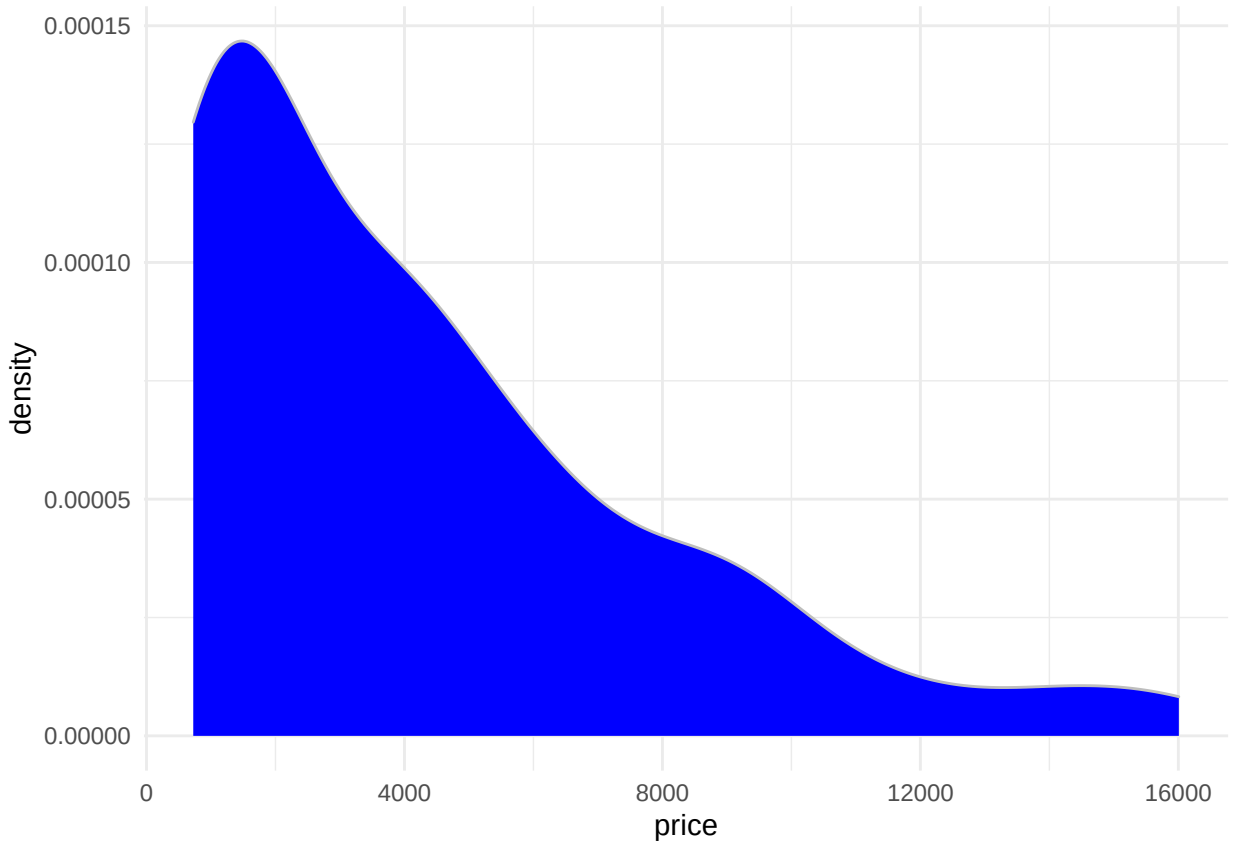



```
#densidade
#necesita so unha variable, continua.
ggplot(datos,aes(x=carat)) +
```

```
  geom_density(fill="aquamarine", color="salmon")+
  theme_minimal()
```



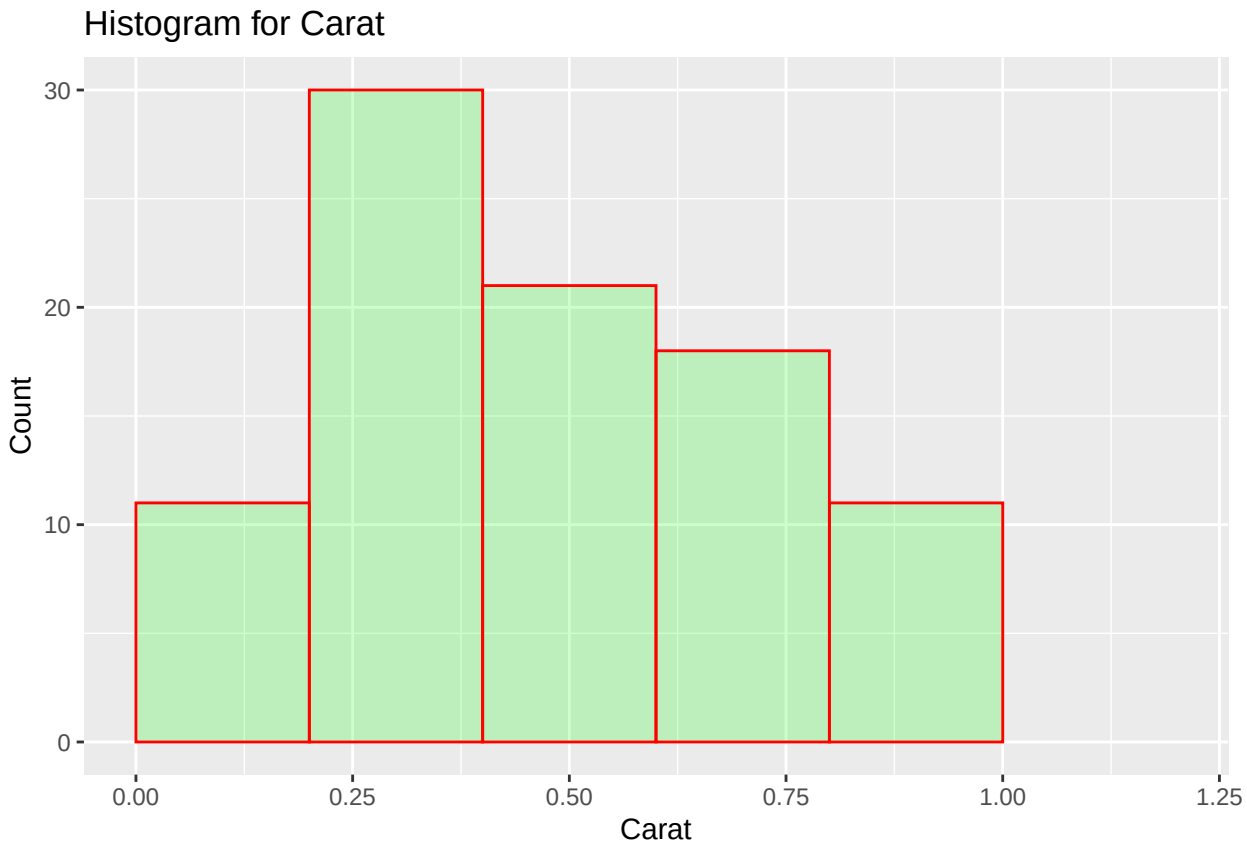

```
ggplot(datos,aes(x=price)) +
  geom_density(fill="blue", color="gray")+
  theme_minimal()
```




```
#HISTOGRAMA
# breaks=seq(20, 50, by = 2) metido no geom_histogram indica os extremos dos intervalos
#seq(20, 50, by = 2) crea unha sucesion comezando en 20, rematando en 50 e saltando de 2 en 2
seq(20, 50, by = 2)
```

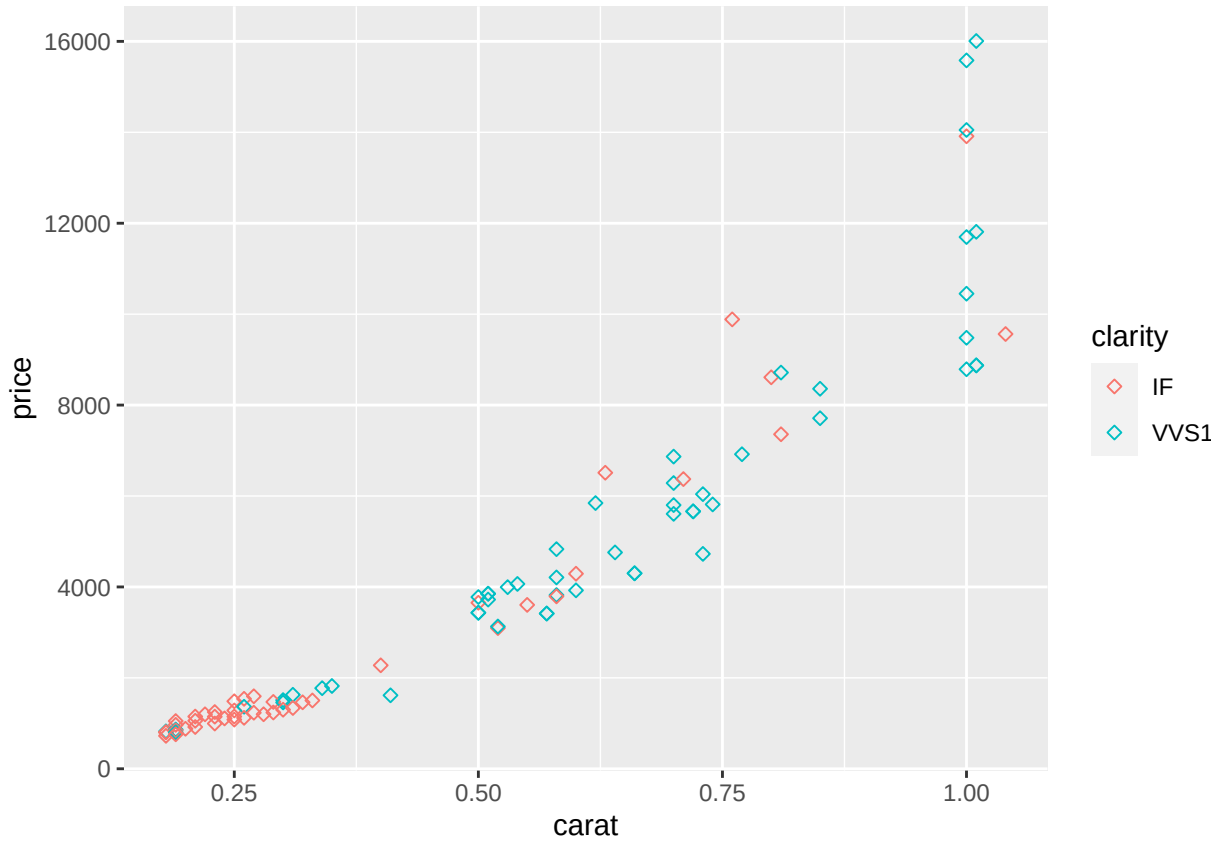
```
##  [1] 20 22 24 26 28 30 32 34 36 38 40 42 44 46 48 50
```

```
#se a liña remata en + continua na liña seguinte:
ggplot(datos, aes(carat)) +
  geom_histogram(breaks=seq(0, 1.2, by = 0.2),
                 col="red",
                 fill="green",
                 alpha = .2) +
  labs(title="Histogram for Carat") +
  labs(x="Carat", y="Count") +
  xlim(c(0,1.2)) +
  ylim(c(0,30))
```

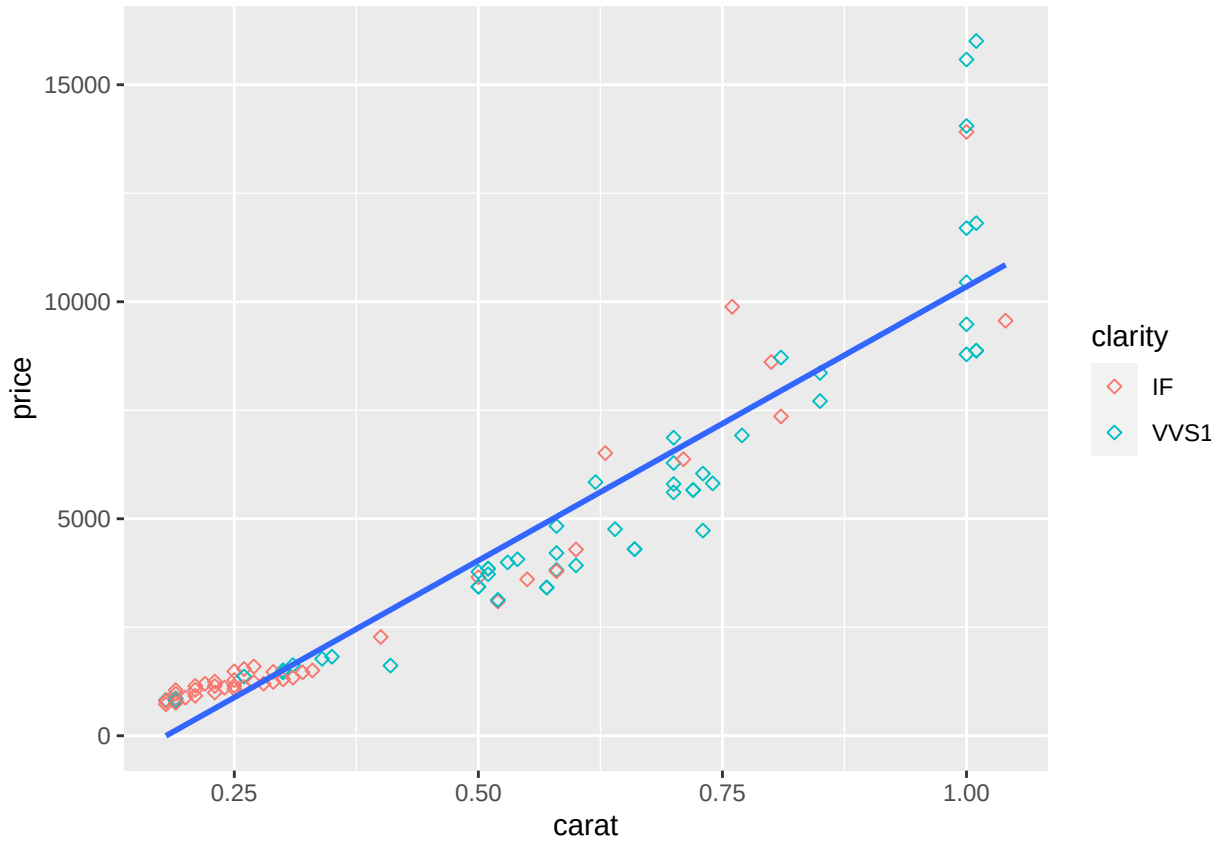



```
#xlim() e ylim() indican entre que valores se debuxan os eixes x e y
```

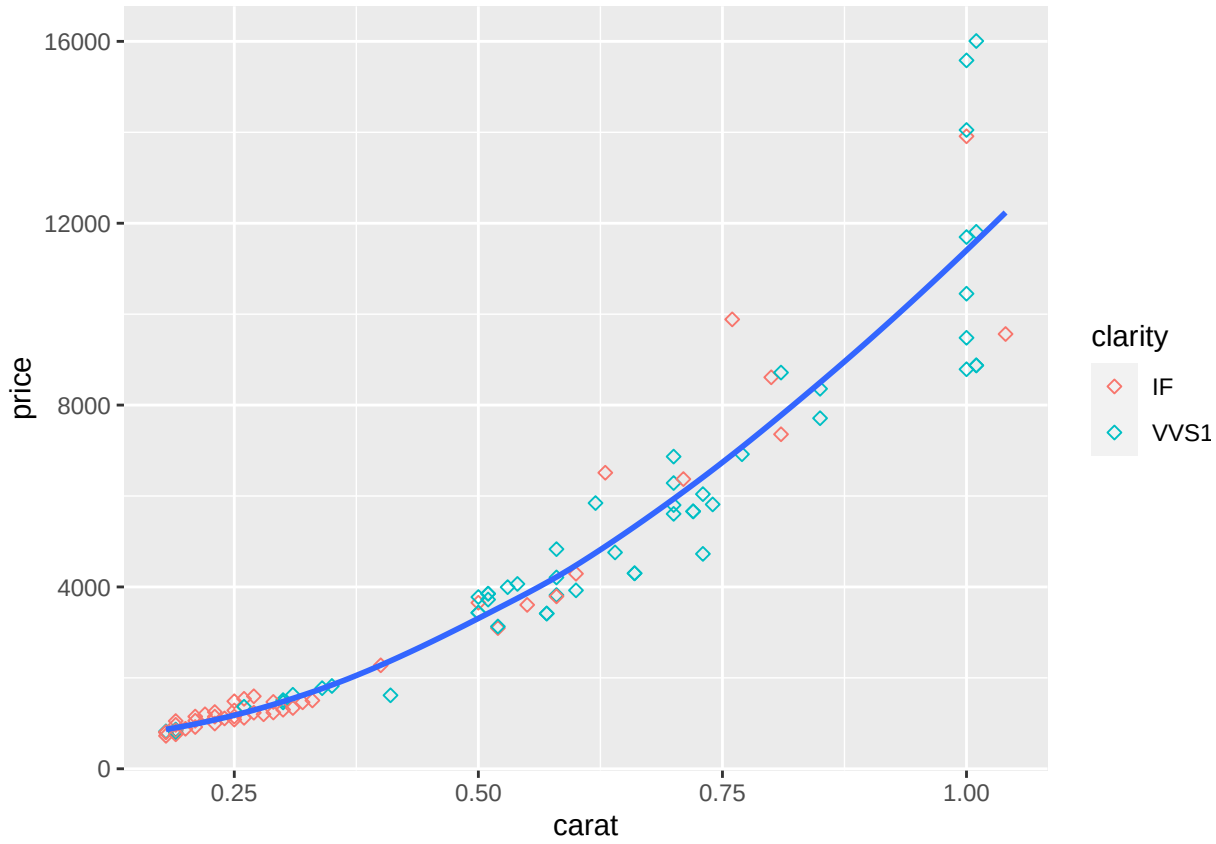
```
## Diagrama de dispersión con recta de regresión e regresión suavizada:
#as graficas creadas en ggplt2 poden ser gardadas como un obxecto e non se ve a súa representación.
#na seguinte liña gardase a grafica realizada e chámaselle p:
p=ggplot(data=datos,aes(x = carat, y = price))+
  geom_point(aes(colour=clarity),shape = 5, fill = "white", size = 1.4)

#Agora pode verse executando p:
p
```



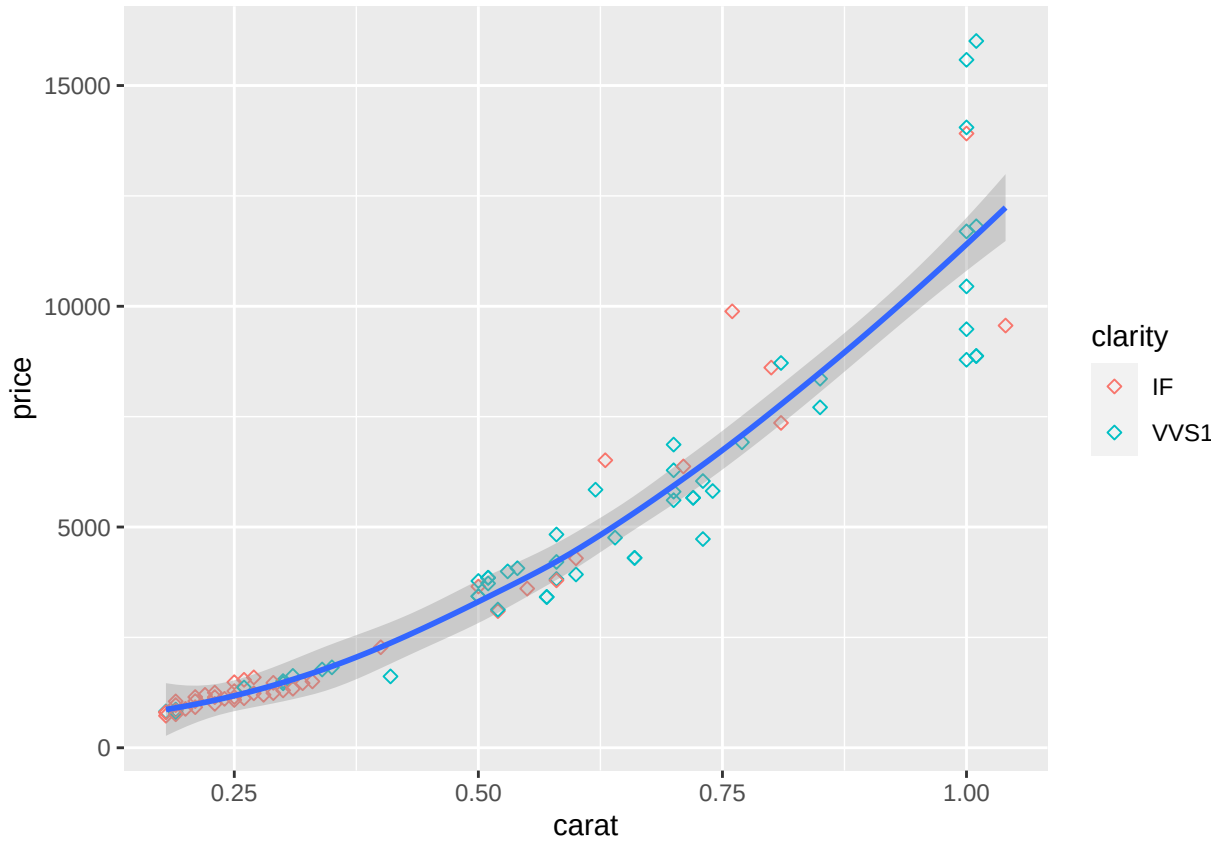

```
#e pódenselle engadir cousas a ver como queda, por exemplo unha regresión liñal
p+   geom_smooth(method=lm,  se=FALSE)
```




```
#ou unha curva de regresión non parámetrica
p+   geom_smooth(method=loess, se=FALSE)
```




```
#cun intervalo de confianza para os valores
p+   geom_smooth(method=loess)
```


15000
10000
5000
0
price
0.25
0.50
0.75
1.00
carat
clarity
IF
VVS1