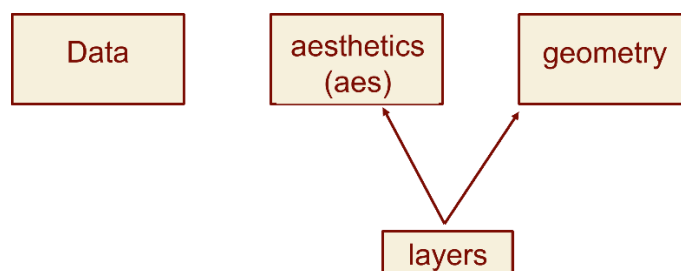


SEMINARI 1. *R / ggplot*. Introducció (Respostes)

1. OBJECTIUS

Aquest seminari serveix per familiaritzar-se amb l'ús de ggplot2 i els seus passos successius.



Si no l'heu instal·lat encara, instal·leu i carregueu la llibreria tidyverse.

```
> install.packages("tidyverse")  
> library(tidyverse)
```

NOTA: En aquest seminari, l'únic objectiu és familiaritzar-se amb l'ús i l'estructura de ggplot i veure les diferents “point shapes” segons el tipus de dades que tenim. Treballarem només amb `geom_point()`. Per tant, les visualitzacions que farem en aquest seminari NO seran les més adequades pel tipus de dades, però això ho anirem veient amb els següents seminaris, on, un cop ja familiaritzats amb les eines, sí que farem un especial èmfasis en aquest segon aspecte.

2. PART 1. Com és el nostre dataset? Quin tipus de variables hi tenim?

El conjunt de dades *mtcars* conté informació de 32 cotxes. És un conjunt de dades petit que conté una varietat de variables contínues i categòriques i ens permetrà familiaritzar-nos amb ggplot2. Podeu utilitzar `str()` per explorar la estructura d'aquest dataset.

Si obriu R de nou, primer de tot recordeu que heu de tornar a carregar la llibreria tidyverse.

```
> library(tidyverse)
```

Podeu utilitzar `str()` per explorar la estructura d'aquest *dataset*.

```
> str(mtcars)
'data.frame': 32 obs. of 11 variables:
 $ mpg : num 21 21 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 ...
 $ cyl : num 6 6 4 6 8 6 8 4 4 6 ...
 $ disp: num 160 160 108 258 360 ...
 $ hp : num 110 110 93 110 175 105 245 62 95 123 ...
 $ drat: num 3.9 3.9 3.85 3.08 3.15 2.76 3.21 3.69 3.92 3.92 ...
 $ wt : num 2.62 2.88 2.32 3.21 3.44 ...
 $ qsec: num 16.5 17 18.6 19.4 17 ...
 $ vs : num 0 0 1 1 0 1 0 1 1 1 ...
 $ am : num 1 1 1 0 0 0 0 0 0 0 ...
 $ gear: num 4 4 4 3 3 3 3 4 4 4 ...
 $ carb: num 4 4 1 1 2 1 4 2 2 4 ...
```

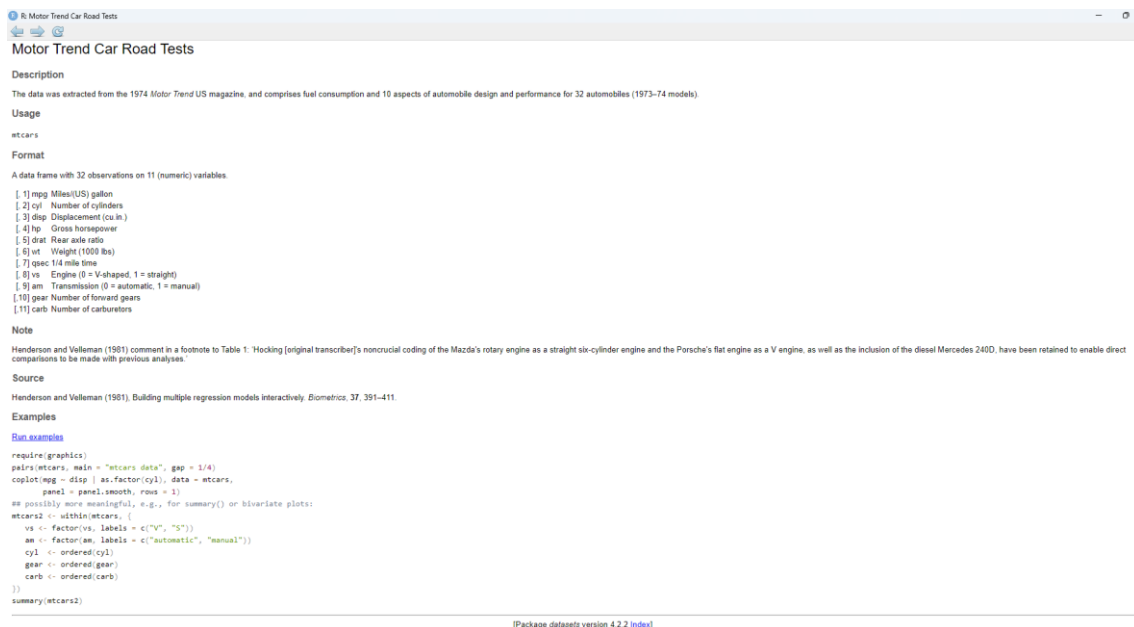
I per saber la definició de cada variable utilitzeu

```
> ?mtcars
```

O:

```
> help (mtcars)
```

Se us obrirà una pantalla nova:



Motor Trend Car Road Tests

Description

The data was extracted from the 1974 *Motor Trend* US magazine, and comprises fuel consumption and 10 aspects of automobile design and performance for 32 automobiles (1973–74 models).

Usage

```
mtcars
```

Format

A data frame with 32 observations on 11 (numeric) variables.

- [1] mpg Miles/(US) gallon
- [2] cyl Number of cylinders
- [3] disp Displacement (cu.in.)
- [4] hp Gross horsepower
- [5] drat Rear axle ratio
- [6] wt Weight (1000 lbs)
- [7] qsec 1/4 mile time
- [8] vs Engine (l = V-shaped, 1 = straight)
- [9] am Transmission (0 = automatic, 1 = manual)
- [10] gear Number of forward gears
- [11] carb Number of carburetors

Note

Henderson and Velleman (1981) comment in a footnote to Table 1: 'Hocking (original transcriber)'s noncrucial coding of the Mazda's rotary engine as a straight six-cylinder engine and the Porsche's flat engine as a V engine, as well as the inclusion of the diesel Mercedes 240D, have been retained to enable direct comparisons to be made with previous analyses.'

Source

Henderson and Velleman (1981), Building multiple regression models interactively. *Biometrics*, 37, 391–411.

Examples

[Run examples](#)

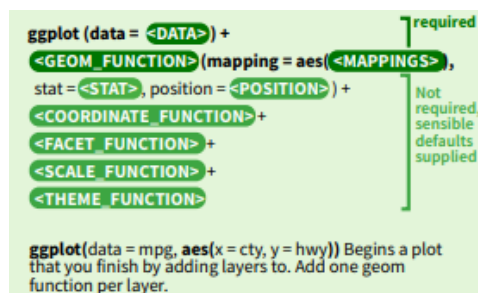
```
require(ggplot2)
pairs(mtcars, main = "mtcars data", gap = 1/4)
coplot(mpg ~ disp | as.factor(cyl), data = mtcars,
       panel = panel.smooth, rows = 1)
# as possibly more meaningful, e.g., for summary() or bivariate plots:
mtcars2 <- within(mtcars, {
  vs <- factor(vs, labels = c("V", "S"))
  am <- factor(am, labels = c("automatic", "manual"))
  cyl <- ordered(cyl)
  gear <- ordered(gear)
  carb <- ordered(carb)
})
summary(mtcars2)
```

[Package datasets version 4.2.2 [Index](#)]

EXERCICIS:

1.- Utilitzeu *ggplot* per dibuixar una gràfica on l'eix x correspongui a la variable 'cyl' (cilindres) i l'eix y a la variable 'mpg' (km de galó). Utilitzeu `geom_point()`.

Hem vist a classe que



```
ggplot(data = <DATA>) +
  <GEOM_FUNCTION>(mapping = aes(<MAPPINGS>),
    stat = <STAT>, position = <POSITION>) +
  <COORDINATE_FUNCTION> +
  <FACET_FUNCTION> +
  <SCALE_FUNCTION> +
  <THEME_FUNCTION>
```

required

Not required, sensible defaults supplied

ggplot(data = mpg, aes(x = cyl, y = hwy)) Begins a plot that you finish by adding layers to. Add one geom function per layer.

Per tant posarem:

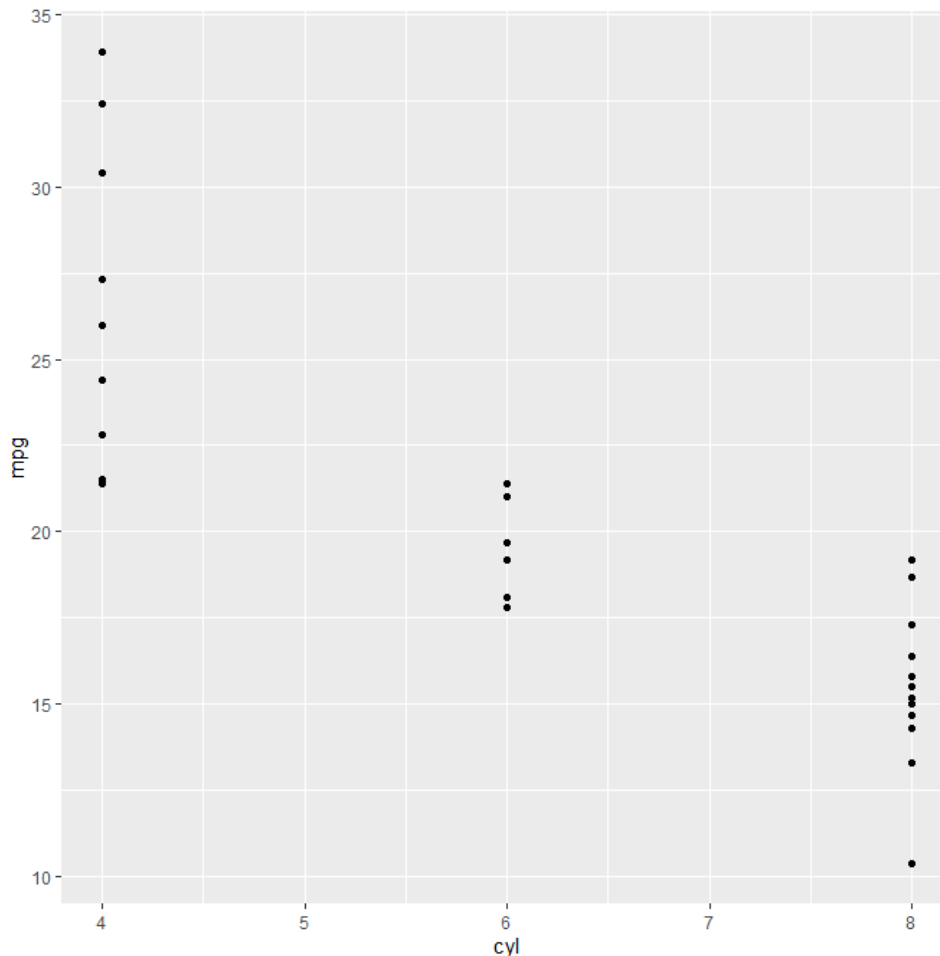
```
> ggplot(data=mtcars, aes(x=cyl, y=mpg))+geom_point()
```

O per simplificar:

```
> ggplot(mtcars, aes(cyl, mpg))+geom_point()
```

```
> ggplot(mtcars) + aes(cyl, mpg)+geom_point()
```

Noteu que al utilitzar `geom_point()`, el *ggplot* tracta la variable 'cyl' com una variable continua. Tenim una gràfica, on dona la impressió que existeix algun automòbil de 5 o 7 cilindres quan no és així.



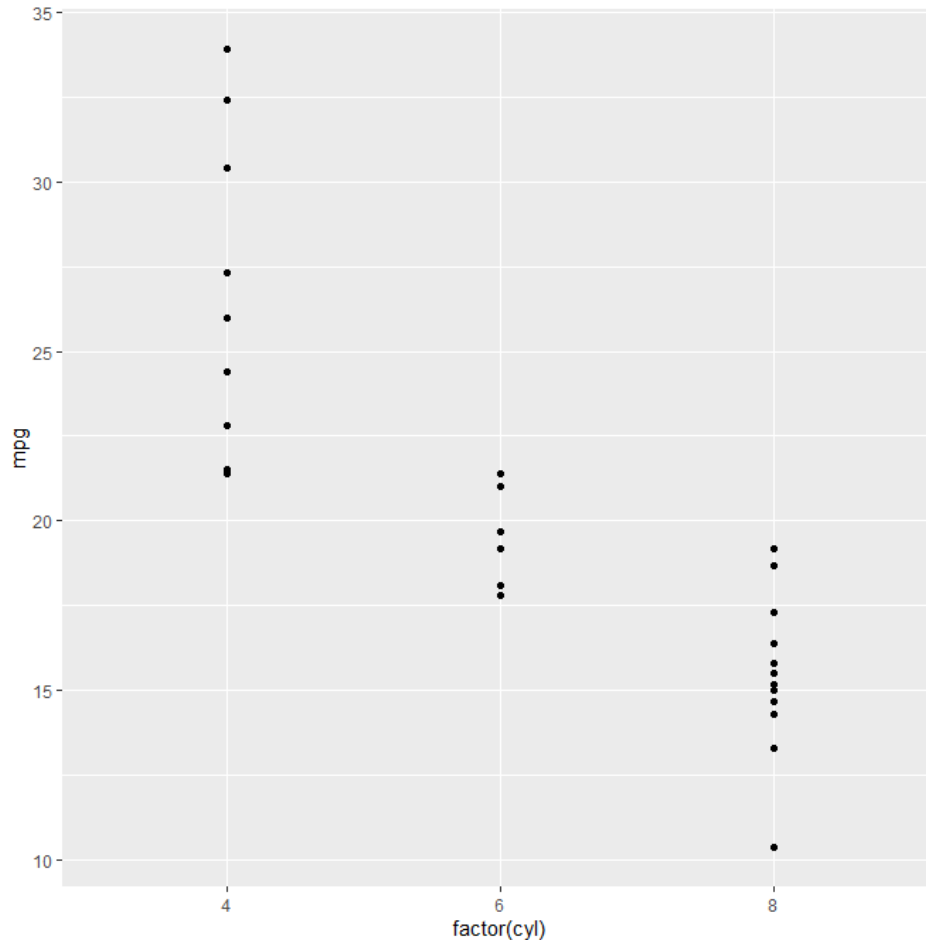
2.- Utilitzeu la funció *ggplot*, però ara categoritzeu la variable 'cyl' ordinal. Per això utilitzeu la funció *factor*. Quina informació podeu extreure'n d'aquesta gràfica?

NOTA: Primer escriviu **?factor** per a que R us digui com especificar que 'cyl' és una variable ordinal que ens està diferenciant en tres grups/nivells de cotxes (els que tenen 4, 6 o 8 cilindres). És el que abans quan veiem els tipus de variables em anomenat factors.

Només hem de canviar *cyl* per *factor(cyl)*

```
> ggplot(mtcars, aes(factor(cyl), mpg))+geom_point()
```

```
> ggplot(mtcars)+aes(factor(cyl), mpg)+geom_point()
```



Ara l'eix x no conté valors com 5 o 7 indicant una certa continuïtat errònia de la variable, sinó només els valors que estaven al data set.

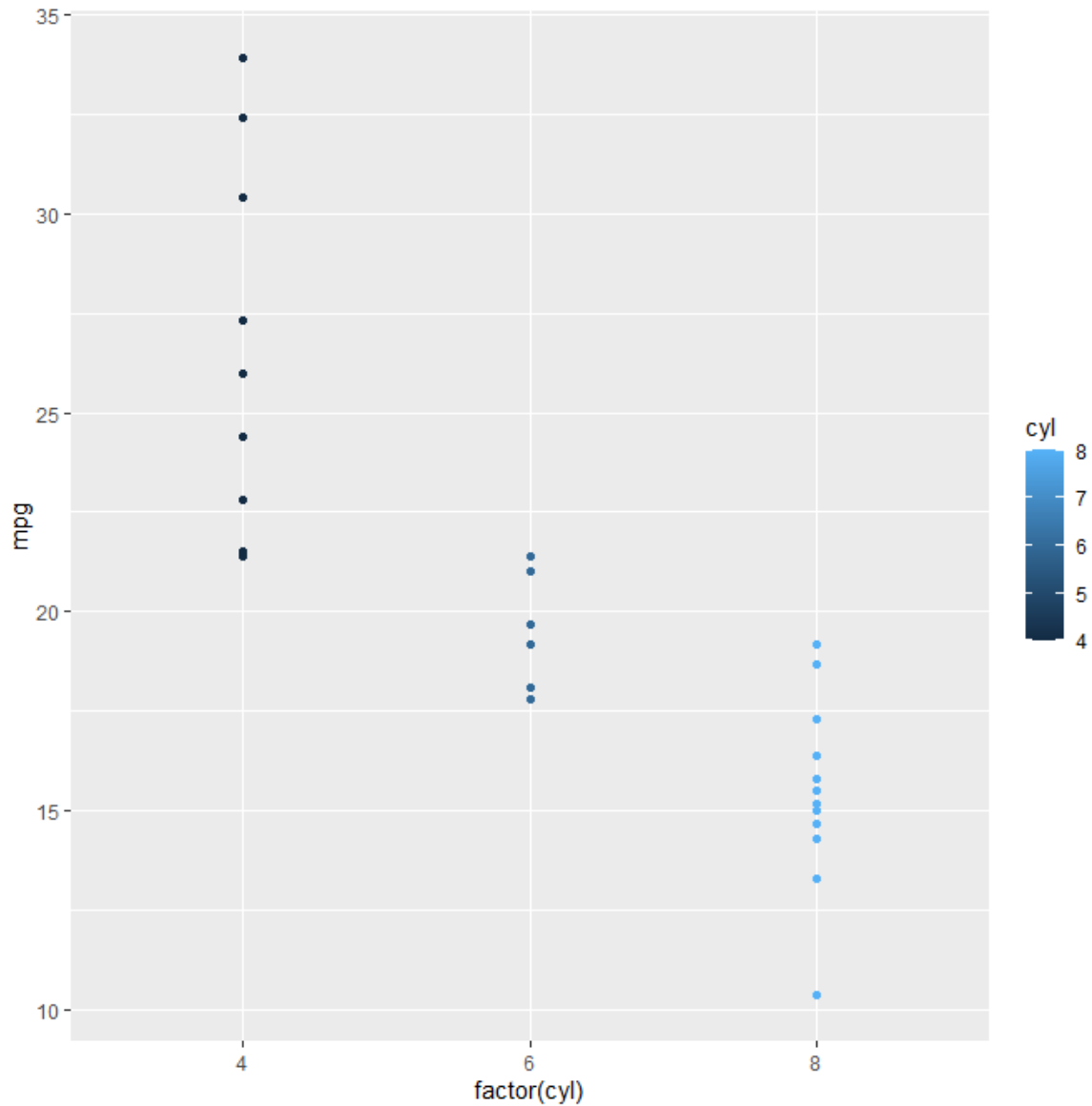
Veiem que els cotxes amb més cilindres són els que menys consumeixen, mentre que els cotxes amb més cilindres són els que més consumeixen.

3.- Afegiu un color segons els cilindres que tingui el cotxe. Ens aporta alguna informació nova? Per què?

El color en ggplot s'afegeix fent un mapeig de la propietat estètica color. Dins d'`aes()`, afegim un argument color igual a la variable 'cyl'. Recordeu que 'cyl' és a la base una variable numèrica i si no diem el contrari, R la tractarà com a variable continua.

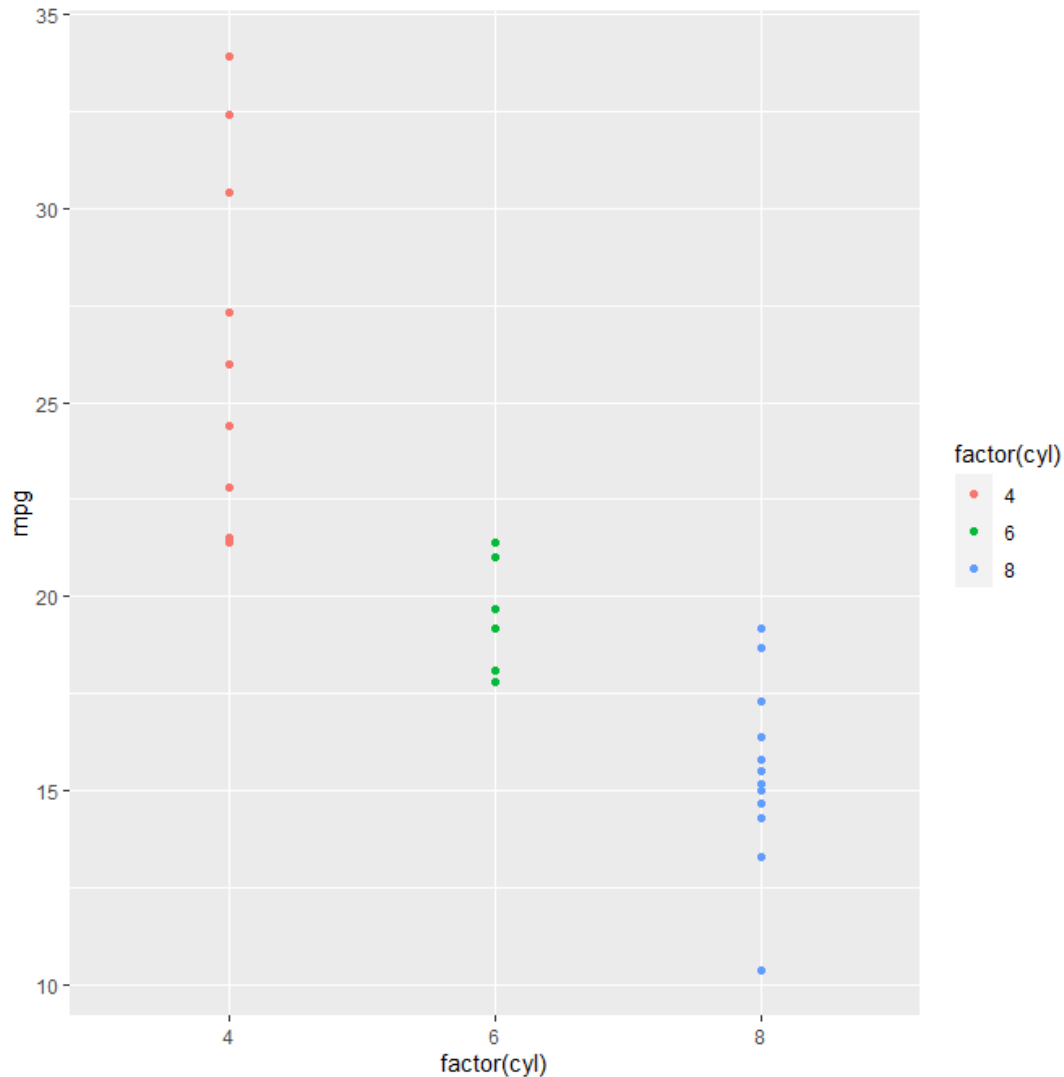
Si fem:

```
>ggplot(mtcars, aes(factor(cyl), mpg, color=cyl))+geom_point()
```



L'escala de color mostra 'cyl' com una variable continua altre cop. Per tant, especifiquem que la variable 'cyl' és un factor com abans (la categoritzem). Això ens permetrà donar un color als cotxes que utilitzen 4 cilindres, diferent del color dels que n'utilitzen 6 i dels que n'utilitzen 8.

```
>ggplot(mtcars, aes(factor(cyl), mpg, color=factor(cyl)))+geom_point()
> ggplot(mtcars)+aes(factor(cyl), mpg, color=factor(cyl))+geom_point()
```



Ara bé, el color no ens ha afegit cap informació nova respecte la gràfica de l'exercici 2 on els grups ja quedaven diferenciats en l'eix x.

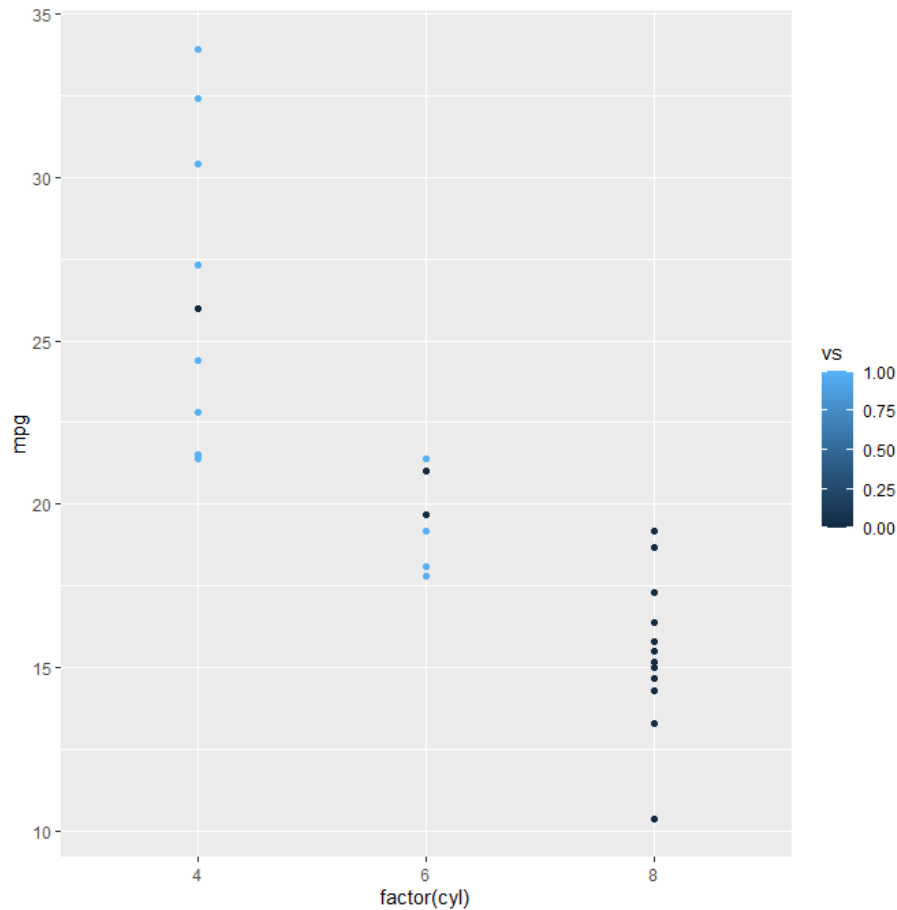
4.- Seguint amb la mateixa gràfica (on l'eix x correspongui a la variable 'cyl' i l'eix y a la variable 'mpg'). Afegiu ara un color al motor del cotxe (*engine*), per això primer recordeu mirar com és la variable 'vs' i feu els ajustos necessaris. Un cop tenim la gràfica, ens aporta alguna informació nova respecte la gràfica de l'exercici 2? Per què? Quina és aquesta informació?

Utilitzeu `?scale_x_discrete` , `?scale_x_continuous` i `?scale_color_discrete` per posar el nom als eixos i a la llegenda de colors amb *scale*

Tot i que no estem mostrant la variable engine (vs), de la mateixa manera que en l'apartat anterior, dins d'`aes()`, podem afegir un argument color igual a la variable vs

Si no mirem com és la variable vs :

```
> ggplot(mtcars, aes(factor(cyl), mpg, color=vs))+geom_point()
> ggplot(mtcars) + aes(factor(cyl), mpg, color=vs)+geom_point()
```



Sembla que 'vs' prengui valors en una escala de 0 a 1. Però fent `?mtcars` veiem que 'vs' té dos valors només que diferencien entre dos grups/nivells segons la forma del motor:

`> ?mtcars`

Files Plots Packages Help Viewer

str Refresh Help Topic

R: Motor Trend Car Road Tests Find in Topic

Usage

```
mtcars
```

Format

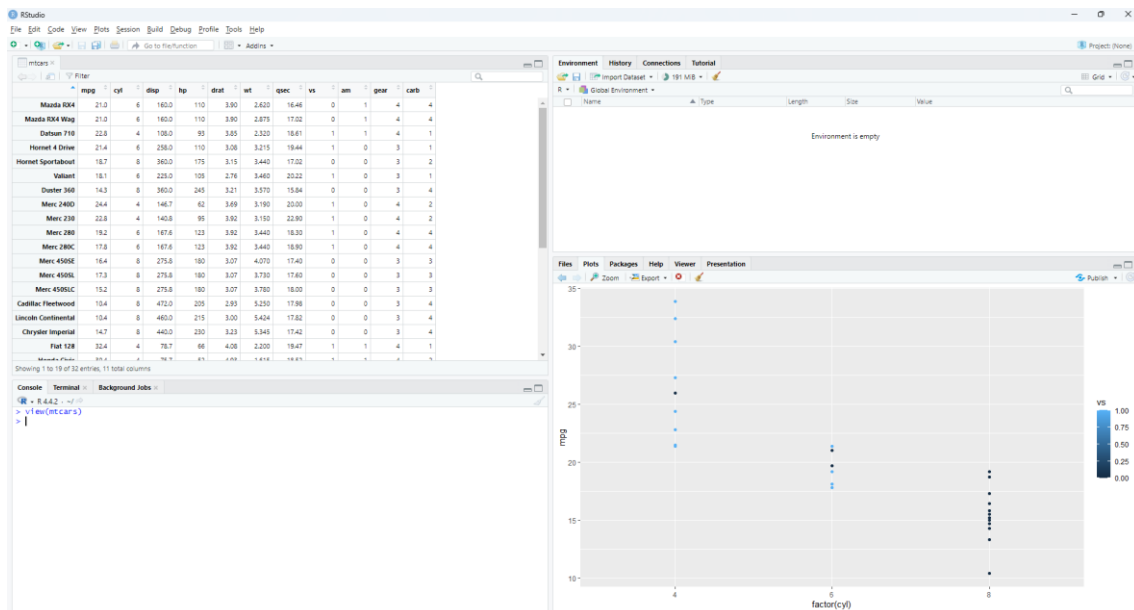
A data frame with 32 observations on 11 (numeric) variables.

- [1] mpg Miles/(US) gallon
- [2] cyl Number of cylinders
- [3] disp Displacement (cu.in.)
- [4] hp Gross horsepower
- [5] drat Rear axle ratio
- [6] wt Weight (1000 lbs)
- [7] qsec 1/4 mile time
- [8] vs Engine (0 = V-shaped, 1 = straight)
- [9] am Transmission (0 = automatic, 1 = manual)
- [10] gear Number of forward gears
- [11] carb Number of carburetors

Note

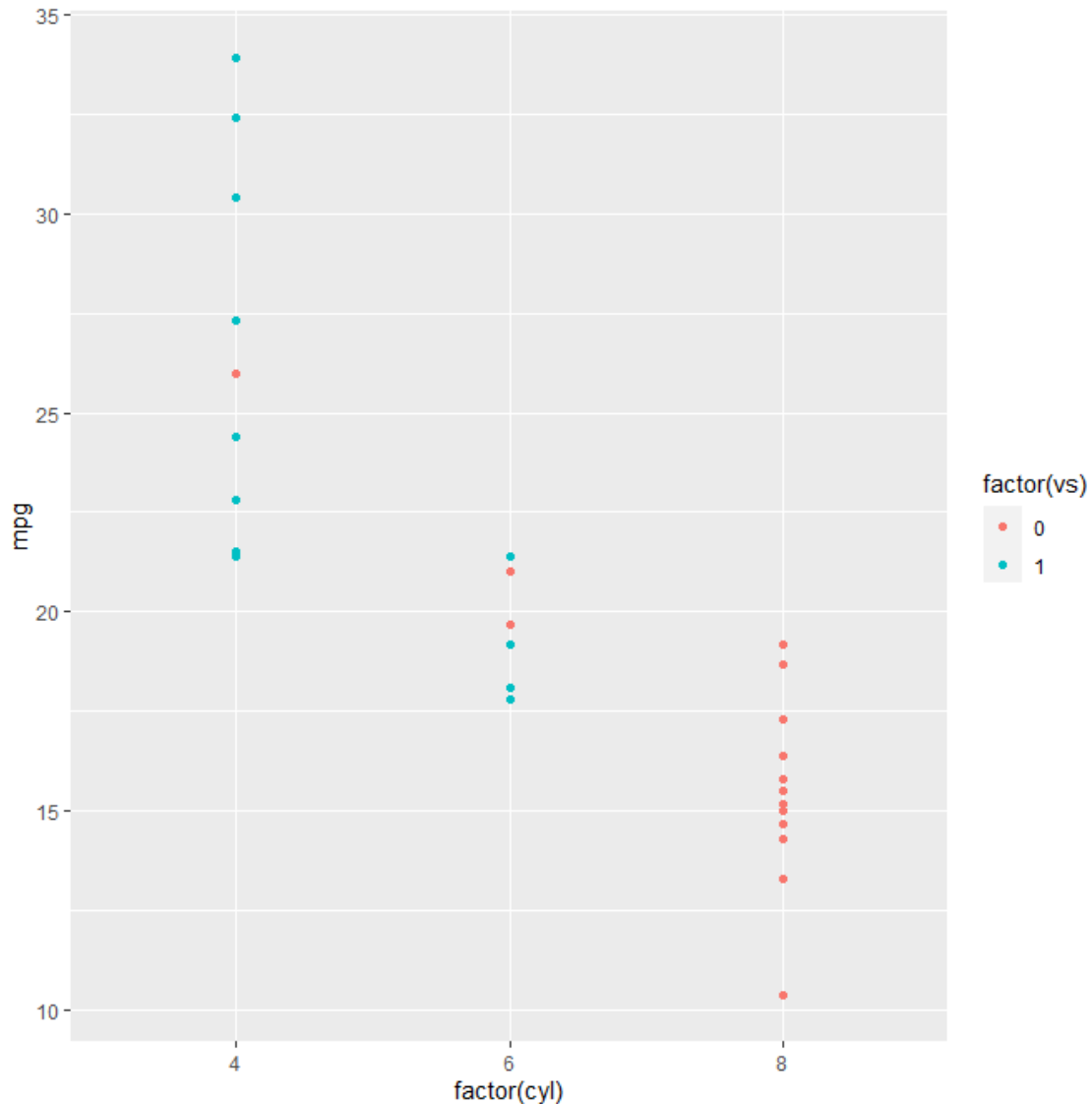
També podem fer us de 'view' per re-assegurar-nos tot visualitzant les dades:

```
> view(mtcars)
```



Per tant 'vs' s'ha de categoritzar també utilitzant factor (o convertir en la variable factor de R):

```
> ggplot(mtcars, aes(factor(cyl), mpg, color=factor(vs)))+geom_point()
```

Posem llegenda als nous colors de la variable factor 'vs':

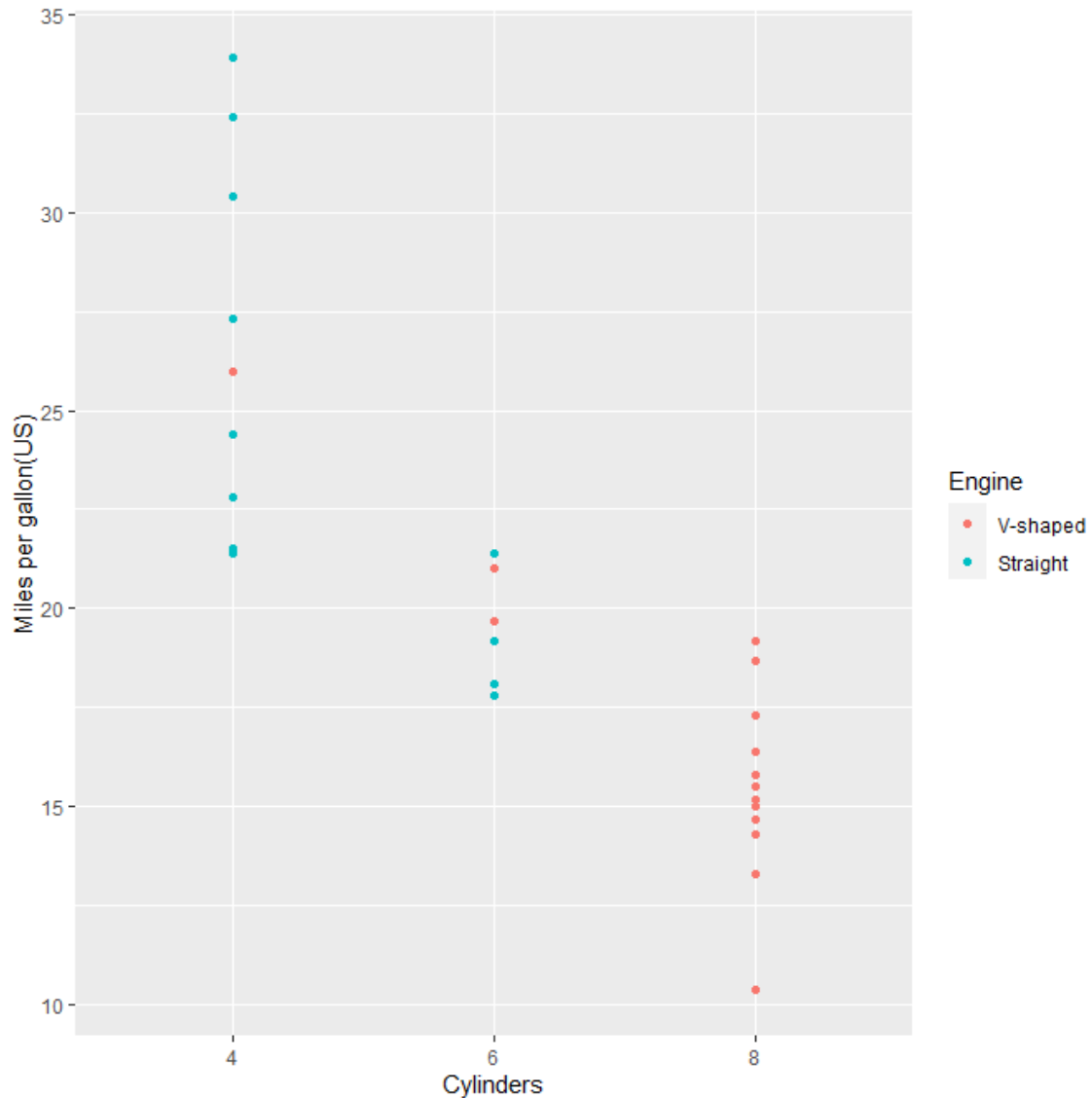
```
> ggplot(mtcars, aes(factor(cyl), mpg, color=factor(vs)))+geom_point()+
scale_color_discrete("Engine", labels = c("V-shaped", "Straight"))
```

I posant llegenda als eixos també (COMPTE, l'eix x és discret):

```
>ggplot(mtcars, aes(factor(cyl), mpg, color=factor(vs)))+geom_point()+
scale_x_discrete("Cylinders")+scale_y_continuous ("Miles per
gallon(US)") +scale_color_discrete("Engine", labels = c("V-
shaped", "Straight"))
```

O el que és el mateix:

```
>ggplot(mtcars, aes(factor(cyl), mpg, color=factor(vs)))+geom_point()+
scale_x_discrete("Cylinders")+scale_y_continuous ("Miles per
gallon(US)") +scale_colour_hue("Engine", labels = c("V-
shaped", "Straight"))
```



Aquí el color sí que ens aporta informació, doncs gràcies al color podem afegir en el mateix gràfic de l'exercici 2 la informació d'un nou factor (nova variable discreta amb dos valors segons la forma del motor). Veiem per exemple que tots els cotxes de 8 cilindres tenen el motor en forma de V.

NOTA: Ara bé, com hem dit al principi, avui hem usat `geom_point()` per simplicitat i per tal de familiaritzar-nos amb ggplot, però en el seminari 2 veurem (amb més profunditat) que *quan dues de les tres variables són qualitatives* (com és el cas de cilindres i motor), *hi ha altres tipus de gràfiques que ens aporten més informació*. Anem-hi pensant amb el que fem a classe de teoria.

D'altra banda, aquest exercici estava fet una mica 'manualment'. Podríeu fer un segon *dataframe* on tot estigués categoritzat i fer directament el dibuix

```
>mtcars2 <- within(mtcars, {
  vs <- factor(vs, labels = c("V", "S"))
  cyl <- factor(cyl)
})
```

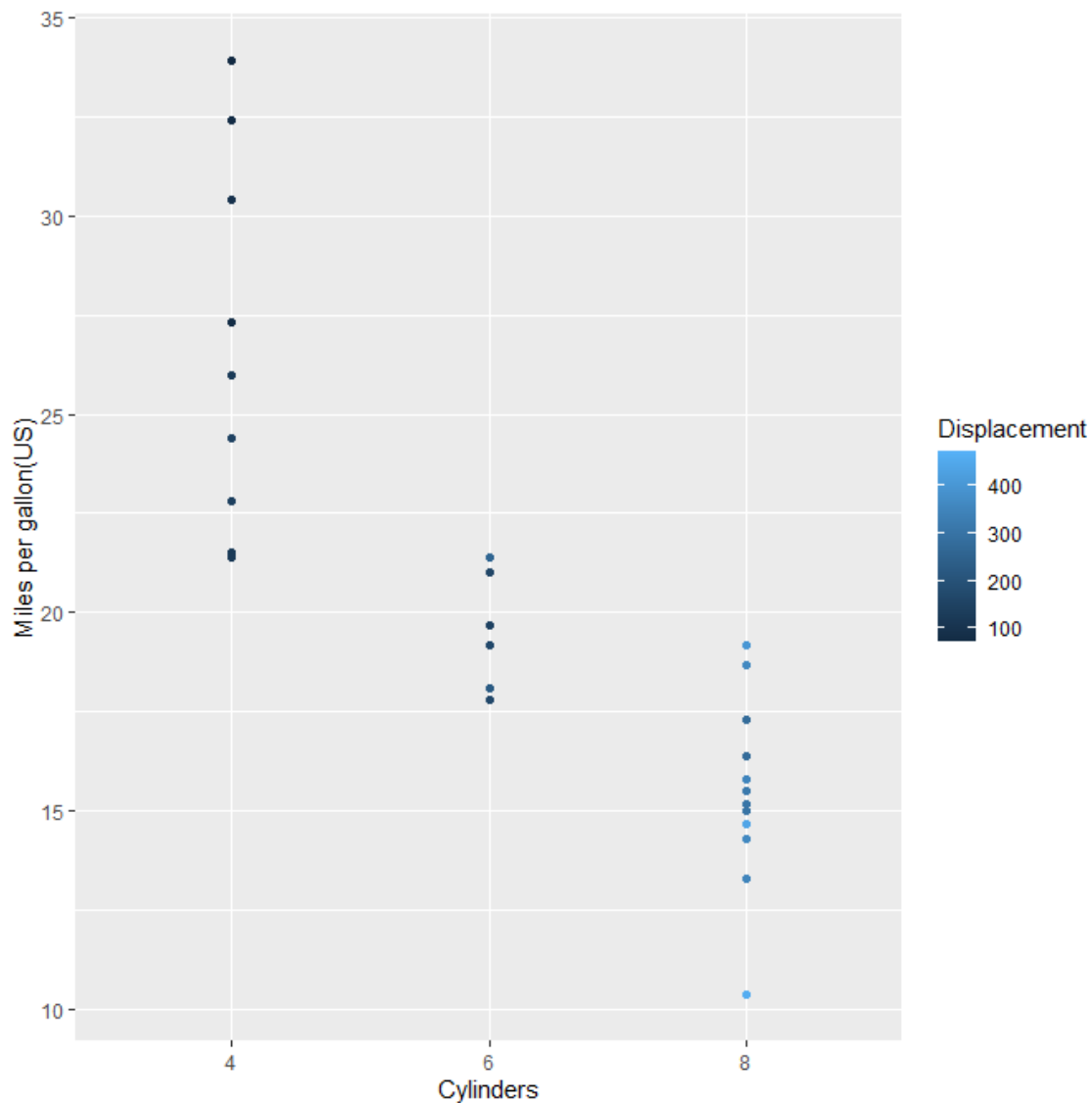
```
> ggplot(mtcars2, aes(cyl, mpg, color=vs))+geom_point()
```

A l'haver fet l'assignació `vs <- factor(vs, labels = c("V", "S"))` al nou dataframe ja no hem de pensar a què es refereixen els valors 0 o 1 de la variable `vs`.

5.- Afegiu ara un color a la variable *Displacement* de cada cotxe i poseu les llegendes adients. És fàcil de veure el que ens aporta aquesta nova informació? Per què? Podeu millorar la visualització de la gràfica d'una manera simple? Quina informació diríeu que en podeu extreure al veure les dades gràficament?

Tot es pot fer molt semblant als apartats anteriors, tenint en compte però que a l'hora de posar el títol a l'escala de color em d'especificar que la variable desplaçament és contínua:

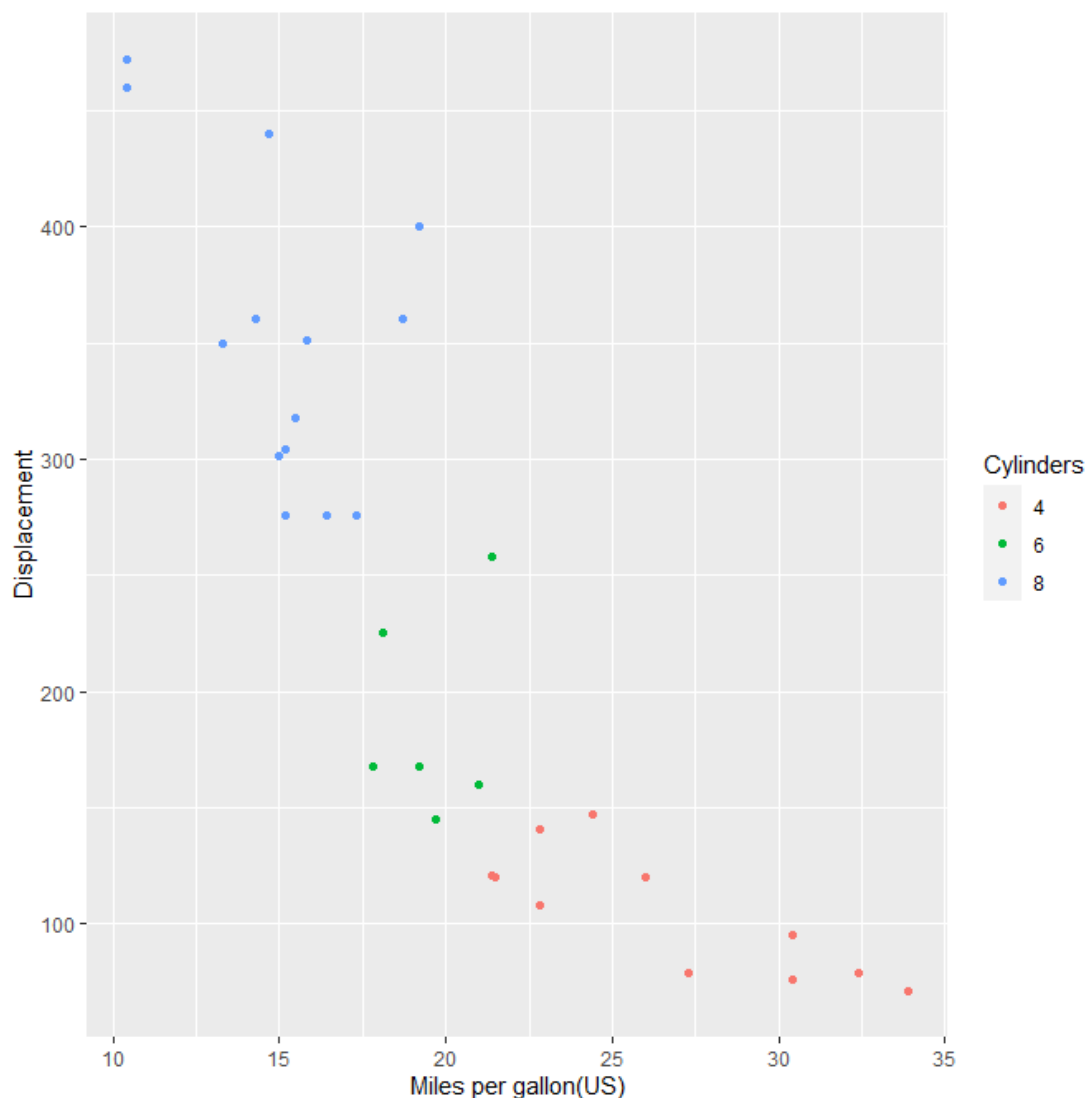
```
> ggplot(mtcars, aes(factor(cyl), mpg, color=disp))+geom_point()+  
scale_x_discrete("Cylinders")+scale_y_continuous("Miles per  
gallon(US)") +scale_color_continuous("Displacement")
```



NOTA: Aquí la variable que afegim és contínua i l'escala de color ha de ser contínua també. Però la veritat es que al visualitzar una escala contínua per aquestes dades, ens costa diferenciar el valor desplaçament en cada punt. Sembla que el color en un gràfic de punts com aquest és més indicat per diferenciar entre variables discretes com en l'exercici 4, però veureu més d'això més endavant (a teoria – en la classe de color- i als seminaris).

Quan volem visualitzar dues variables quantitatives contínues i una variable qualitativa discreta com és aquest cas, **és més adient que l'eix x i y continguin la informació de les variables** contínues i el color s'afegeixi com a variable discreta (on la nostra percepció visual funcionarà millor). Per exemple, podem extreure més informació dibuixant la següent gràfica que no pas l'anterior:

```
> ggplot(mtcars, aes(mpg, disp, color=factor(cyl)))+geom_point()+
scale_x_continuous("Miles per gallon(US)")+
scale_y_continuous("Displacement")+scale_color_discrete("Cylinders")
```



Aquí es veu clarament que quan més cilindres té un cotxe (més volum combinat per tant), aquest fa un desplaçament major amb menys consum. En canvi, quants menys cilindres té el cotxe, aquest necessita un major consum per molt menys desplaçament.

6.- Seguint amb la mateixa gràfica (on l'eix x correspongui a la variable 'cyl' i l'eix y a la variable 'mpg') de l'exercici 4. Intenteu utilitzar *shape* en *aes()* per posar una forma segons cada desplaçament. Què creieu que passa? Podeu utilitzar *shape* amb alguna variable? Quina per exemple?

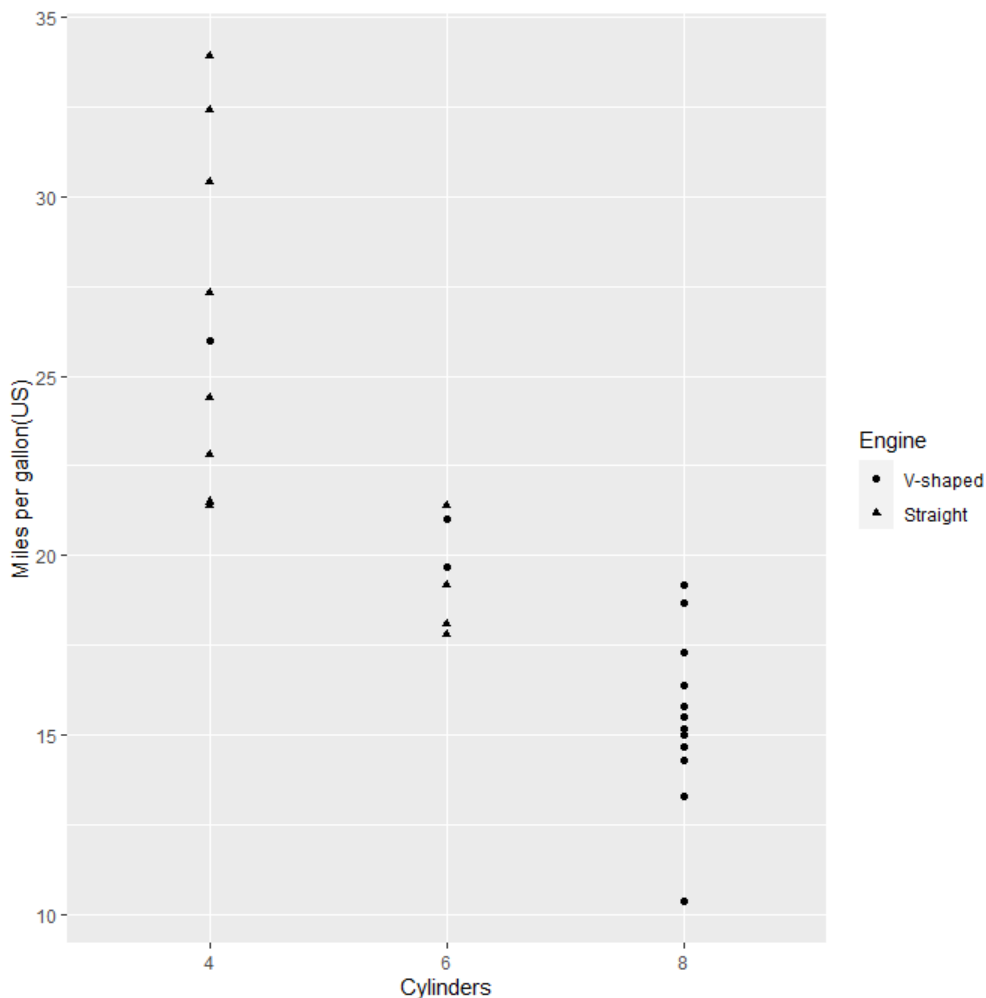
Provem amb la variable *disp*:

```
> ggplot(mtcars, aes(factor(cyl), mpg, shape=disp))+geom_point()
```

Obtenim un error de R, doncs *shape* només té sentit amb variables discretes i la variable *disp* (desplaçament) és contínua. Per tant, necessitem utilitzar *shape* amb una variable discreta.

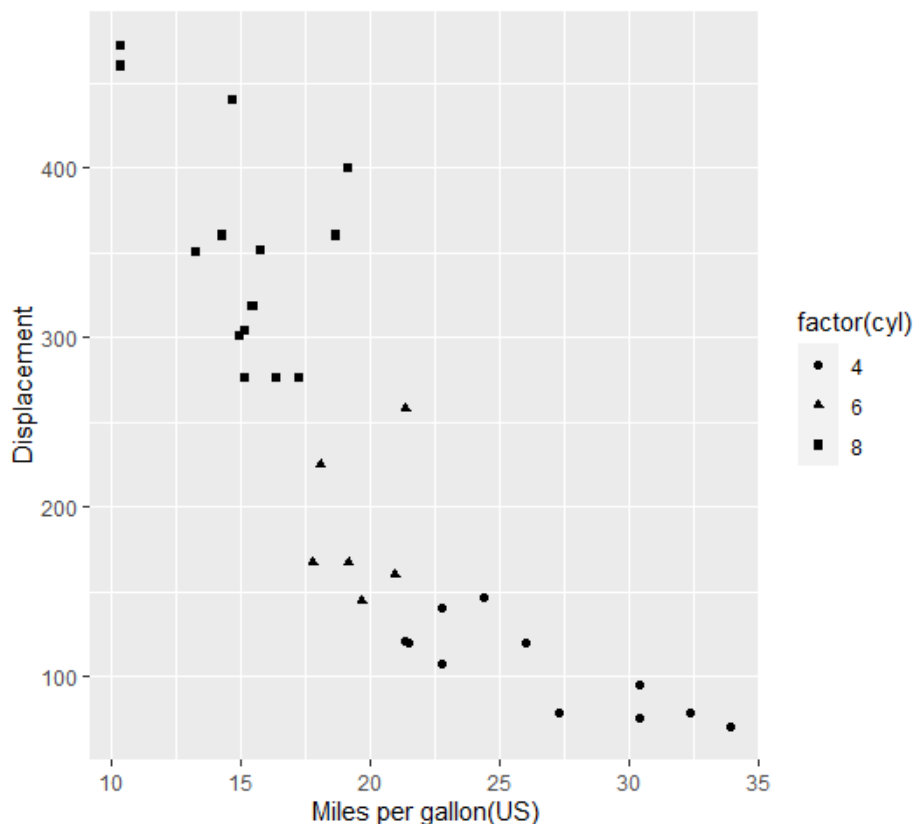
Aquí teniu un exemple com el de l'exercici 4 (tot i que ja hem introduït que *geom_point()* no era la millor forma de visualitzar una variable contínua, versus dues qualitatives) :

```
> ggplot(mtcars, aes(factor(cyl), mpg, shape=factor(vs)))+ geom_point()+
scale_x_discrete("Cylinders")+scale_y_continuous("Miles per
gallon(US)") + scale_shape_discrete("Engine", labels = c("V-
shaped", "Straight"))
```



EXTRA: Tot i que no se us demana a l'enunciat, si hi penseu, un exemple útil per utilitzar *shape* seria reproduir l'exercici anterior canviant el *color* per *shape*. Per exemple, en el cas que volguéssim incloure el gràfic de l'exercici anterior en un document en blanc i negre:

```
> ggplot(mtcars, aes(mpg, disp, shape=factor(cyl)))+geom_point()+
scale_x_continuous("Miles per gallon(US)")+
scale_y_continuous("Displacement")+scale_color_discrete("Cylinders")
```



!! RESUM: Hem vist la importància de saber com és cada variable del dataset per tal de visualitzar-les. El mapeig `aes()` de les propietats estètiques s'anomena "escalatge" i depèn del tipus de variable. El mapeig de les variables discretes es realitza a escales diferents que el de les variables contínues. Per tant avui hem vist:

aes	Discreta	Contínua
Color (color)	Arco iris de colors	Gradient de colors
Forma (shape)	Diferent formes	NO APLICA

També podem experimentar amb altres *aesthetics* i veure per exemple que:

aes	Discreta	Contínua
Talla (size)	Escala discreta de talles	Mapeig lineal entre àrea i el valor
Transparència (alpha)	NO APLICA	Mapeig lineal a la transparència

!! Però també hem vist, gràcies a la comanda `factor()`, que quan una variable numèrica ordinal o lògica ens està diferenciant entre grups/nivells, podem categoritzar-la mitjançant *factor*. Això ens permet veure molt millor la informació que aquestes variables aporten.

Ara que hem vist com d'important és primer familiaritzar-nos amb el nostre dataframe i el tipus de variables que hi tenim per tal d'extreure'n la màxima informació de forma visual, veiem què volem mostrar a la part 2 del seminari.

3. PART 2. Què volem mostrar? Per què és important visualitzar les dades?

Anem a utilitzar el conjunt de dades `anscombe`, del que ja us han parlat a la classe de teoria. Escrivint `anscombe` a R podeu visualitzar-lo. També podeu explorar la seva estructura amb `str(anscombe)`. O accedir a la seva ajuda escrivint `?anscombe`

```
> str(anscombe)
'data.frame': 11 obs. of 8 variables:
 $ x1: num 10 8 13 9 11 14 6 4 12 7 ...
 $ x2: num 10 8 13 9 11 14 6 4 12 7 ...
 $ x3: num 10 8 13 9 11 14 6 4 12 7 ...
 $ x4: num 8 8 8 8 8 8 8 19 8 8 ...
 $ y1: num 8.04 6.95 7.58 8.81 8.33 ...
 $ y2: num 9.14 8.14 8.74 8.77 9.26 8.1 6.13 3.1 9.13 7.26 ...
 $ y3: num 7.46 6.77 12.74 7.11 7.81 ...
 $ y4: num 6.58 5.76 7.71 8.84 8.47 7.04 5.25 12.5 5.56 7.91 ...
> |
```

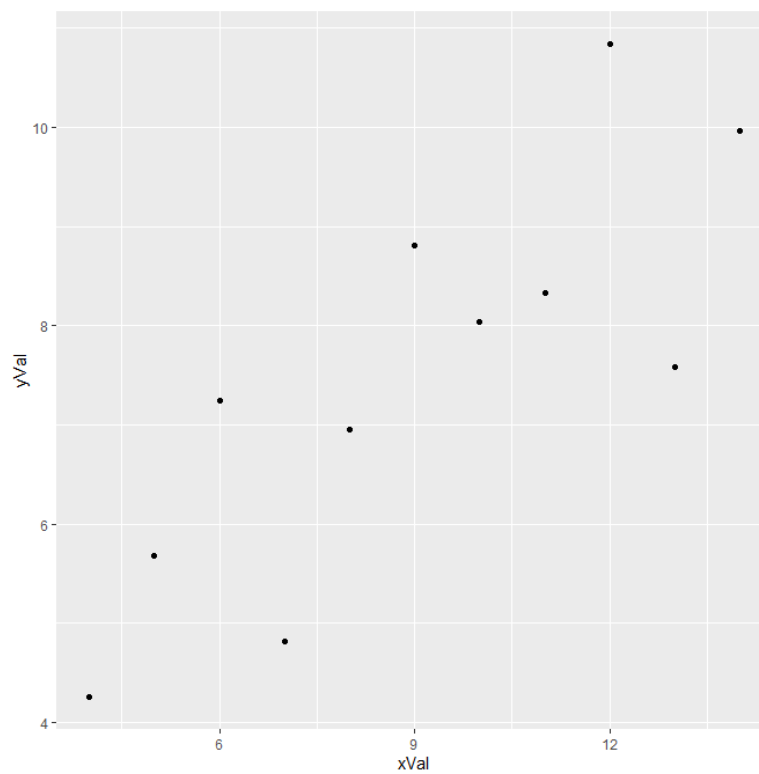
Primer construirem 4 grups de datasets. Els anomenarem `g1data`, ..., `g4data`. I cada un contindrà els valors (x_i, y_i) , amb $i=1, \dots, 4$. Després en veure'm les seves respectives mitjanes i desviació estàndards, omplint la taula següent. Tot seguit plantejarem cada grup `g1data`, ..., `g4data` per separat utilitzant `ggplot()` amb `geom_point()`. Què està passant?

	<code>mean(g1data\$xVal)</code>	<code>mean(g1data\$yVal)</code>	<code>sd(g1data\$xVal)</code>	<code>sd(g1data\$yVal)</code>
<code>g1data</code>	9	7.500909	3.316625	2.031568
<code>g2data</code>	9	7.500909	3.316625	2.031657
<code>g3data</code>	9	7.500909	3.316625	2.030424
<code>g4data</code>	9	7.500909	3.316625	2.030579

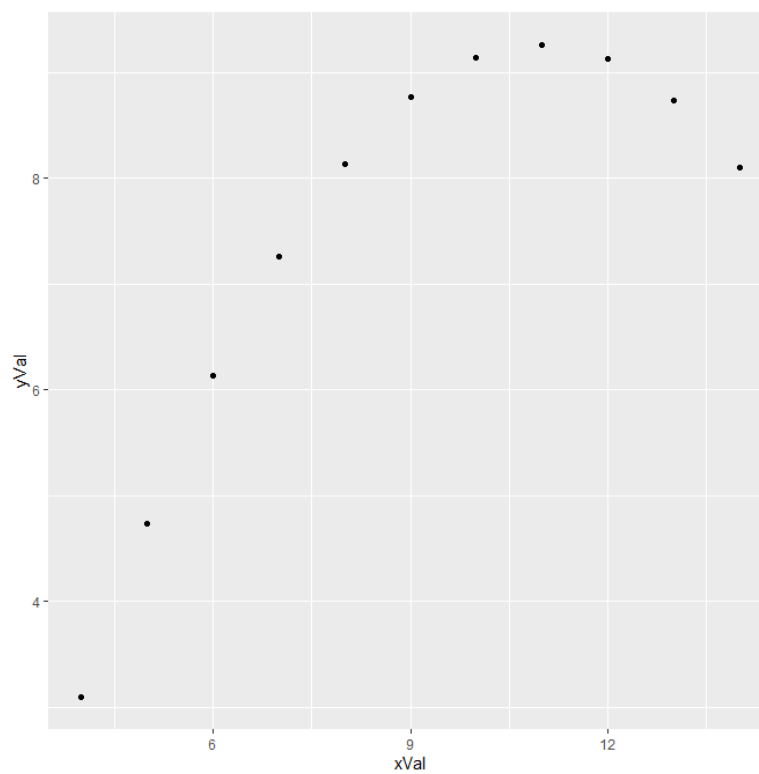
NOTA: `g1data=with(anscombe, data.frame(xVal=c(x1), yVal=c(y1)))` us crearà el primer grup. Per fer les respectives mitjanes farem servir les comandes que hem posat en cada columna de la taula anterior on $i=1, 2, 3, 4$ respectivament:

```
> mean(g1data$xVal)
> mean(g1data$yVal)
> sd(g1data$xVal)
> sd(g1data$yVal)
```

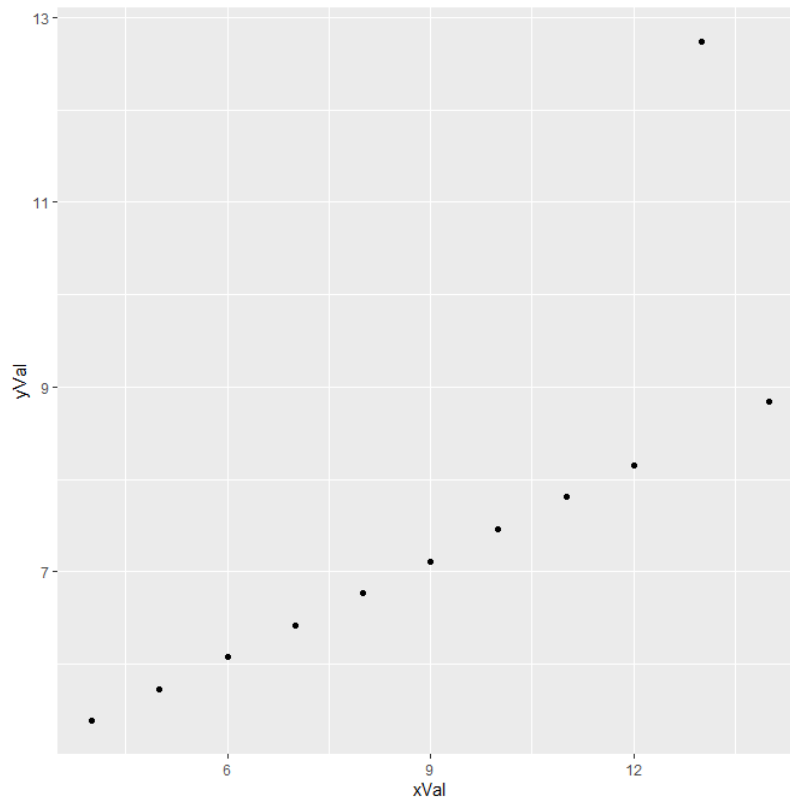
```
> g1data=with(anscombe,data.frame(xVal=c(x1),yVal=c(y1)))
> ggplot(g1data, aes(xVal, yVal))+geom_point()
```



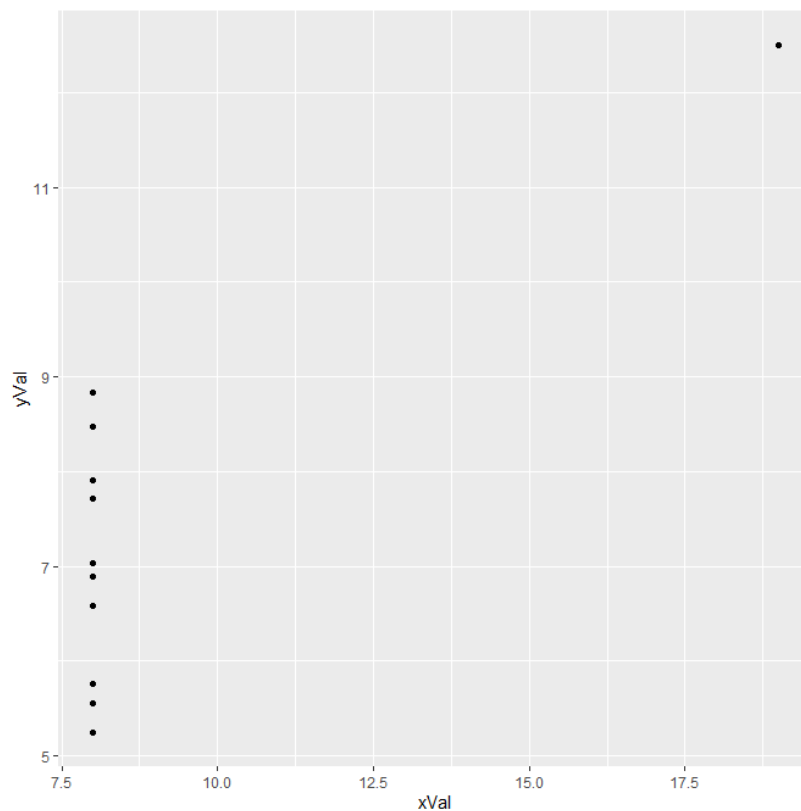
```
> g2data=with(anscombe,data.frame(xVal=c(x2),yVal=c(y2)))
> ggplot(g2data, aes(xVal, yVal))+geom_point()
```




```
> g3data=with(anscombe,data.frame(xVal=c(x3),yVal=c(y3)))
> ggplot(g3data, aes(xVal, yVal))+geom_point()
```



```
> g4data=with(anscombe,data.frame(xVal=c(x4),yVal=c(y4)))
> ggplot(g4data, aes(xVal, yVal))+geom_point()
```



!!RESUM: El dataset 'anscombe' conté quatre conjunts de dades que tenen la mateixa mitjana i la mateixa desviació estàndard (per x i per y) però al visualitzar-les tenen una aparença molt diferent. Per això és important visualitzar les dades que tenim. Si només haguéssim mirat la mitjana o la desviació estàndard del datasets que tenim, haguéssim assumit que els quatre datasets són iguals. La distribució dels punts ens mostra que no. En el pròxim seminari veure'm com mostrar distribucions.

EXTRA: D'una manera més avançada, que encara no hem vist, per crear els grups g1,..., g4 ((xi,yi), amb i=1,..4) d'una manera menys manual i visualitzar-los en un mateix gràfic, escriure'm en R:

```
> anscombedata=with(anscombe,data.frame(xVal=c(x1,x2,x3,x4),
yVal=c(y1,y2,y3,y4), anscombegroup=gl(4,nrow(anscombe))))

> ggplot(anscombedata,aes(x=xVal,y=yVal,group=anscombegroup))+
geom_point()+facet_wrap(~anscombegroup)
```

