

SEMINARI 2. Comparacions i Distribucions (Respostes)

1. OBJECTIUS

La part 1 d'aquest seminari introdueix les geometries de *ggplot* que ens permeten visualitzar comparacions de dades categòriques (nominals i ordinals). La part 2, introdueix les geometries de *ggplot* que ens permeten visualitzar distribucions. A més, el seminari té com objectiu també, introduir algunes eines que ens permetran fer una mínima edició dels nostres gràfics.

2. PART 1. Comparacions mitjançant els diagrames de barres

Continuem amb el conjunt de dades *mtcars* del seminari 1 que conté informació de 32 cotxes. Com el conjunt de dades és petit, poseu `view(mtcars)` per veure el *dataframe* en forma de taula. La funció `view()` ens permet obtenir informació sobre el conjunt de dades de manera complementària a les dues maneres que vam veure l'altre dia: l'ajuda `?mtcars` o la funció mostrant l'estructura, `str(mtcars)`.

Si obriu R de nou, primer de tot recordeu que heu de tornar a carregar la llibreria *tidyverse*: `library(tidyverse)`.

(Els solucionaris contenen una possible solució, però podria haver més d'una solució)

EXERCICIS:

1.- Veient una sola gràfica, volem saber quants cotxes hi ha de tres grups/nivells diferents segons el nombre de cilindres (4 cilindres, 6 cilindres i 8 cilindres). Feu un gràfic de barres on l'alçada de les barres sigui proporcional al nombre de cotxes de cada grup/nivell. Quina informació us dona la gràfica?

- a) Poseu la etiqueta als eixos utilitzant `xlab()` i `ylab()`
- b) Pinteu les barres segons el nombre de cilindres utilitzant `fill` dins `d aes()`. Ens dona alguna informació extra pintar-les? Per què?
- c) Poseu títol i etiquetes a la llegenda que us ha sortit anteriorment per defecte
- d) Feu les barres més estretes especificant la *width* de les barres. Nota: *width* actua com argument de `geom_bar()`

Hem vist les comandes `geom_bar()` i `geom_col()` a la primera part de la classe. Fixeu-vos que volem que l'alçada del nostre gràfic sigui proporcional al nombre de cotxes d'una categoria, per tant utilitzem `geom_bar()`.

En l'exercici 1 del seminari 1, ja vam veure que si no indicàvem que 'cyl' era una variable 'factor', l'eix x on representàvem el nombre de cilindres ens enganyava mostrant una variable contínua. Aquí ens passarà el mateix si fem:

```
> ggplot(mtcars, aes(cyl)) + geom_bar()
```

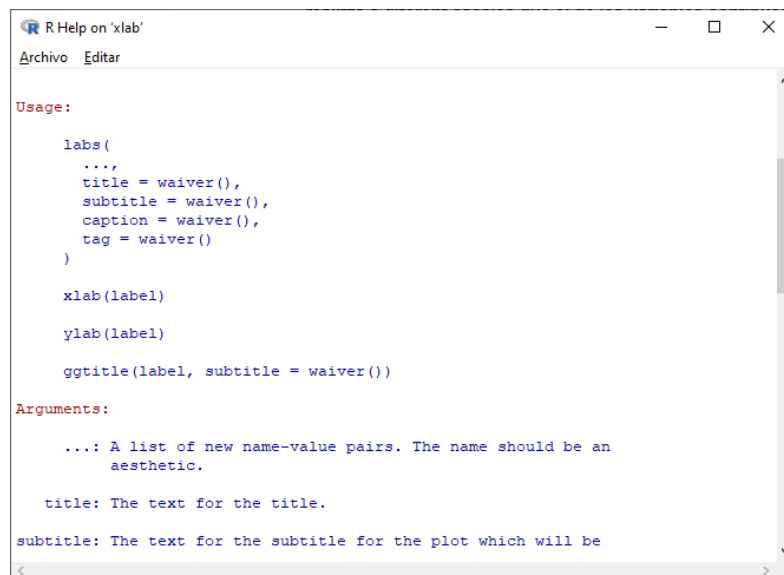
Per tant veiem que hem d'utilitzar *factor* per especificar que 'cyl' és un factor i que no ens passi això. A més avui hem vist que si volem emfatitzar que és una variable categòrica ordinal, com és el cas de 'cyl' podem utilitzar *ordered*. Per simplicitat, fem com vam veure en el seminari 1, i creem ja un dataframe mtcars2 que contingui les variables de mtcars on 'cyl' sigui ara una variable categòrica ordinal:

```
>mtcars2 <- within(mtcars, {
  cyl <- ordered(cyl)
})
> ggplot(mtcars2, aes(cyl)) + geom_bar() #o també:
> ggplot(mtcars2)+ aes(cyl) + geom_bar()
```

a) No hem utilitzat encara xlab i ylab. Demaneu ajuda:

```
>?xlab
```

L'ajuda ens mostra com s'utilitza xlab (ylab s'utilitza igual):



The screenshot shows the R Help window for the 'xlab' function. The window has a title bar 'R Help on 'xlab'' and menu options 'Archivo' and 'Editar'. The main content area displays the following information:

```
Usage:
  labs(
    ...,
    title = waiver(),
    subtitle = waiver(),
    caption = waiver(),
    tag = waiver()
  )

  xlab(label)

  ylab(label)

  ggtitle(label, subtitle = waiver())

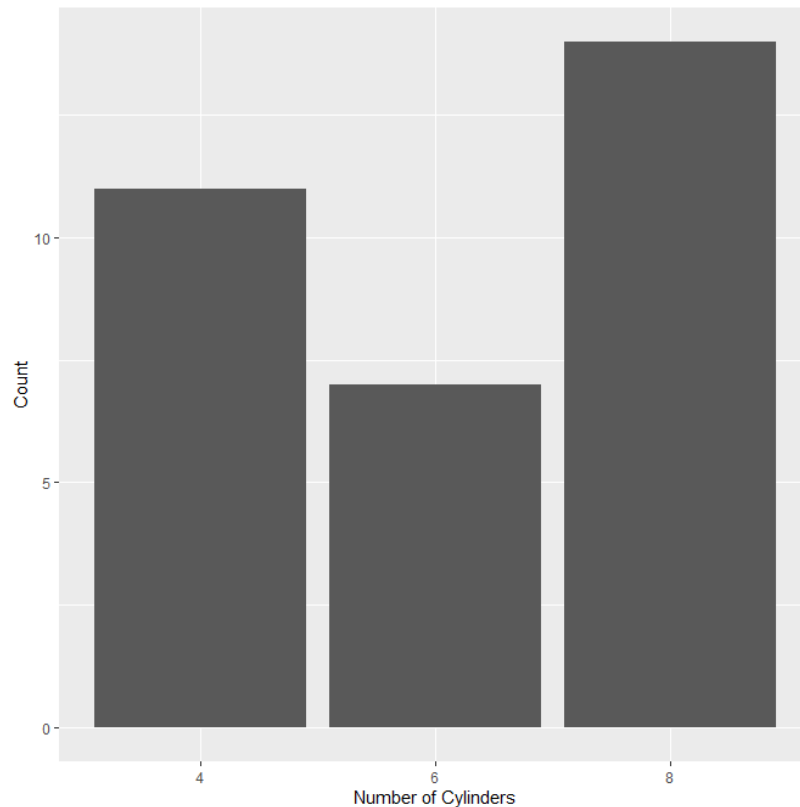
Arguments:
  ...: A list of new name-value pairs. The name should be an
       aesthetic.

  title: The text for the title.

  subtitle: The text for the subtitle for the plot which will be
```

Assignem a una variable g el que teníem i sumem les etiquetes dels eixos:

```
> g<- ggplot(mtcars2)+ aes(cyl) + geom_bar()
> g+xlab("Number of Cylinders")+ ylab("Count")
```



Aquesta gràfica, ens indica d'una manera prou visual que el grup amb més cotxes és el grup dels cotxes que tenen 8 cilindres. De fet en tenim aproximadament el doble que de 6 cilindres, el grup del qual en tenim menys.

NOTA: També podríem fer-ho com vam veure l'altre dia amb *scale*, però ho em fet amb *label* per l'enunciat, si ho féssim seria:

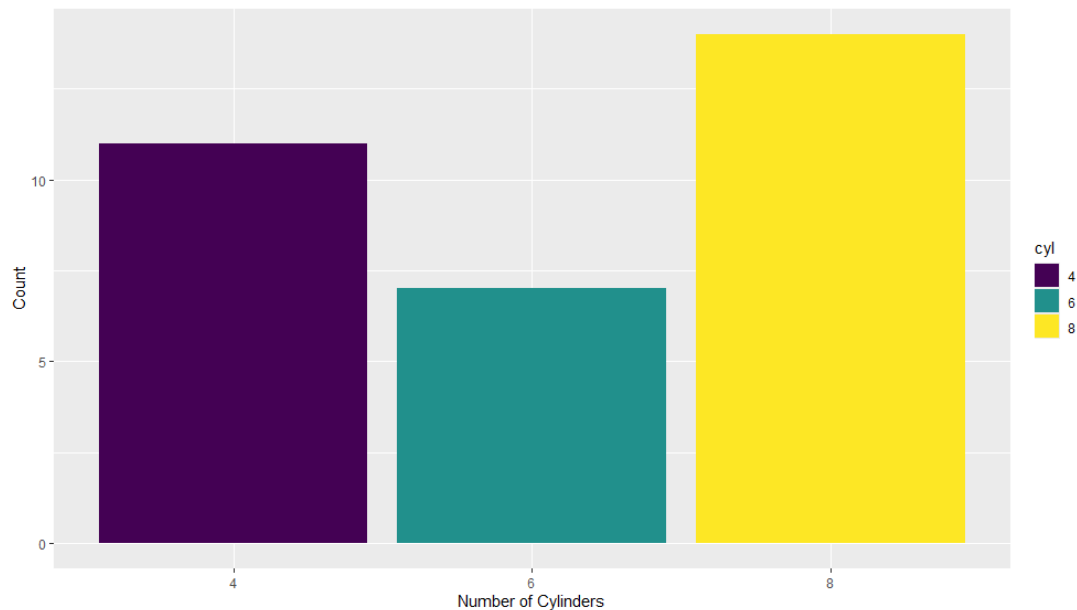
```
>g+ scale_x_discrete("Number of  
Cylindres")+scale_y_continuous("Count")  
  
> g # per mostrar
```

Finalment, també podríem utilitzar *labs*. Veieu:

<https://ggplot2.tidyverse.org/reference/labs.html>

b) Ja vam veure en el 1er seminari que per pintar una gràfica segons un grup de variables necessitàvem fer un mapeig del color en *aes()*, on especificaríem el color segons el grup de variables que volíem. Aquí ens diuen que utilitzem *fill* enlloc de *color*, però ho fem de la mateixa manera (en *aes()*). Nota: Recordeu utilitzar factor amb 'cyl' per a que l'escala de color no sigui contínua.

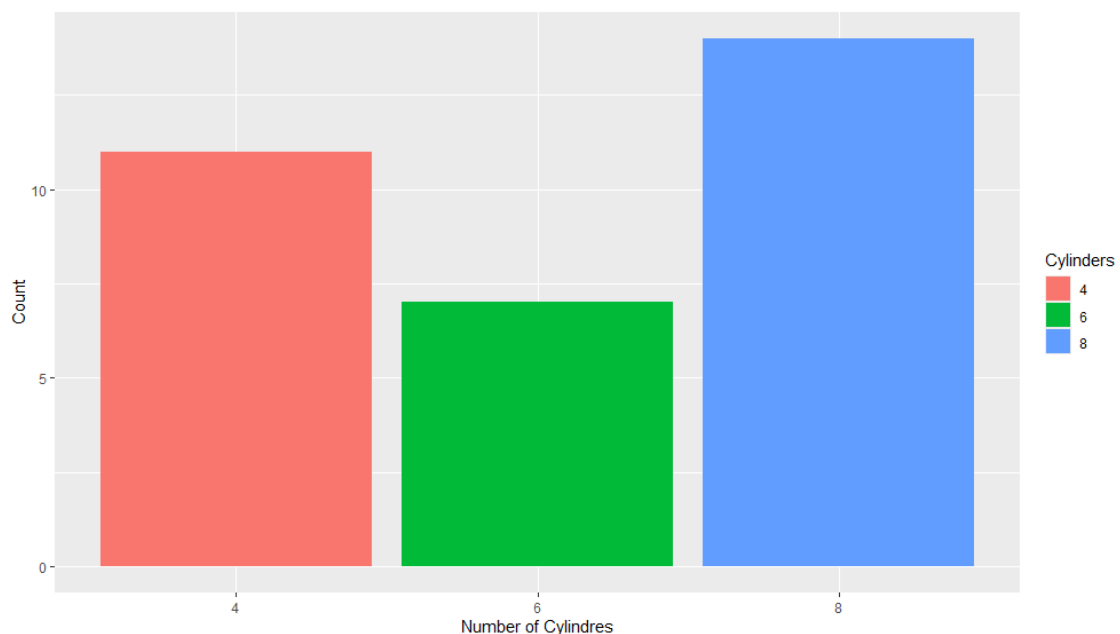
```
> b<-ggplot(mtcars2, aes(cyl, fill=cyl)) + geom_bar() + xlab("Number  
of Cylinders") + ylab("Count")  
  
> b
```



El color NO ens aporta cap informació nova. L'eix x ja ens estava dividint les dades en els tres grups que volíem.

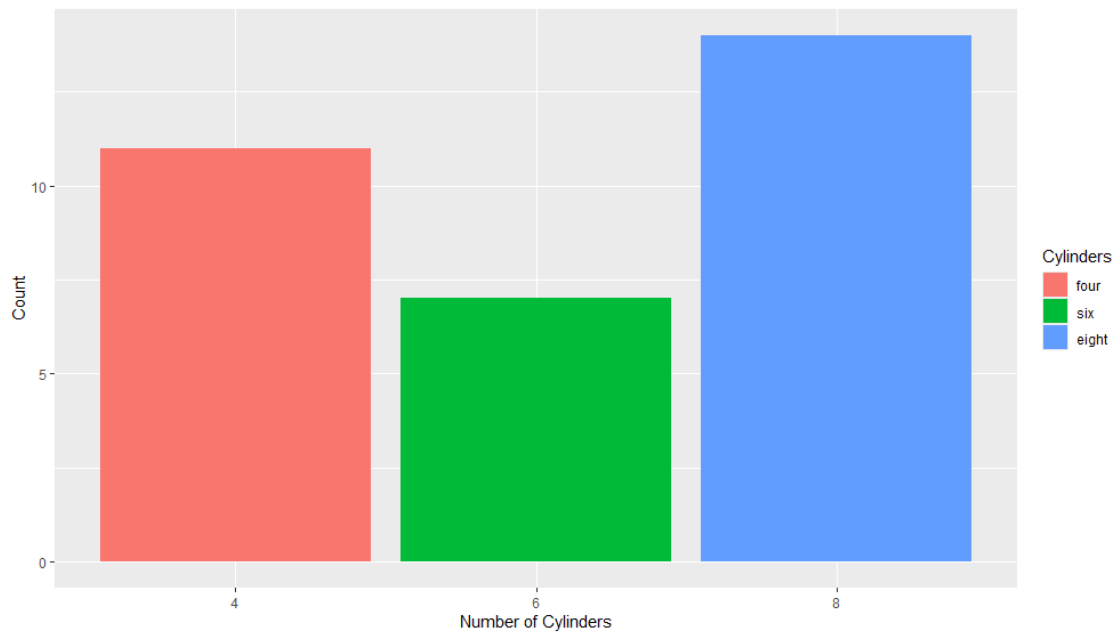
c) Podem posar un títol i etiquetes a la llegenda com l'altre dia, amb `scale_fill_discrete`

```
> b<-ggplot(mtcars2, aes(cyl, fill=cyl)) + geom_bar()
+scale_x_discrete("Number of Cylindres")+scale_y_continuous("Count")
> b+scale_fill_discrete("Cylinders")
```



També podem posar etiquetes personalitzades a la llegenda:

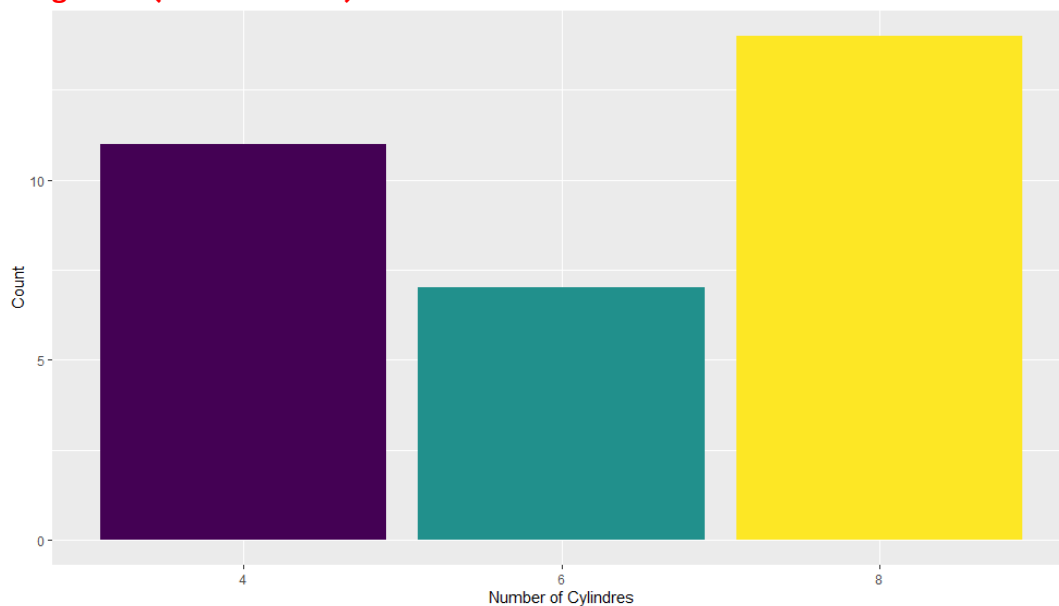
```
> b+scale_fill_discrete("Cylinders", labels=c('four','six','eight'))
```



NOTA: També podríem utilitzar `labs`. Veieu l'enllaç que ja citàvem en l'apartat (a): <https://ggplot2.tidyverse.org/reference/labs.html>

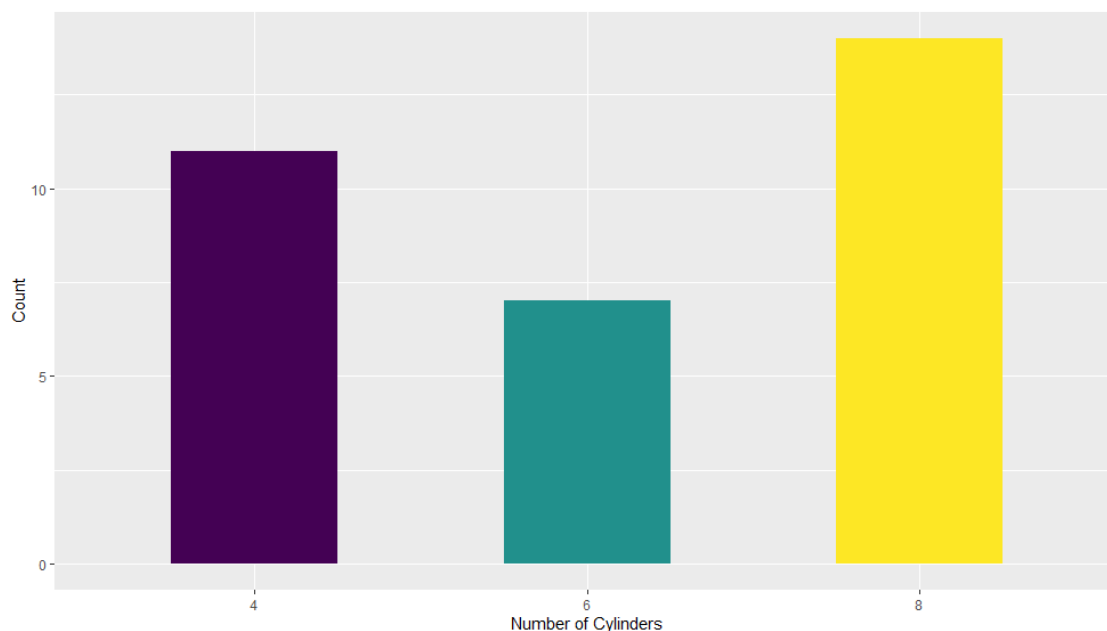
NOTA2: De fet si hi pensem les etiquetes de la llegenda NO ens donen informació extra pel que podríem treure-la posant:

```
>b+guides(fill="none")
```



d) Exemple amb `width=0.5`

```
> ggplot(mtcars2, aes(cyl, fill=cyl)) + geom_bar(width=0.5) +  
xlab("Number of Cylinders") + ylab("Count") +guides(fill="none")
```



Ens permet fer barres més amples o estretes segons la mida posem. L'amplada de les barres ha de ser proporcional a l'eix x.

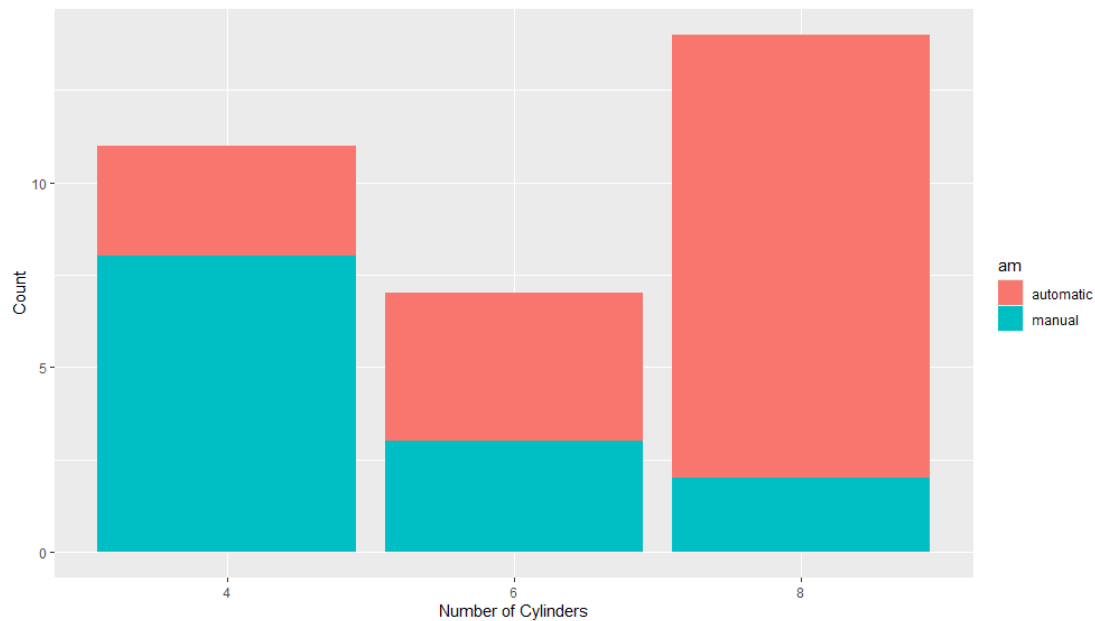
NOTA: Si haguéssim provat de fer una amplada de 1.5, les barres s'haurien sobreexposat i R ens tornaria un *warning* que ens ho indicaria.

2.- Feu un gràfic on pugueu veure la proporció de vehicles amb transmissió manual versus automàtica segons el nombre de cilindres. Què observeu?

Utilitzeu l'argument *dodge* de manera que cada tipus de transmissió aparegui en l'eix x, una al costat de l'altra d'acord amb el nombre de cilindres. Per fer això últim especificarem l'argument *posició* en `geom_bar(position="dodge")`. Observeu més coses?

Com sempre, abans de fer la gràfica, primer mireu com són les variables 'cyl' i 'am'. Pregunteu-vos, per exemple: són contínues/discretes?, divideixen les dades en subgrups?, es poden categoritzar amb *factor* o no?. La variable 'cyl' ja hem vist que es pot categoritzar utilitzant *factor/ordered* segons la connotació que volem donar. D'altra banda, 'am' és una variable lògica que només pren dos valors (0 o 1), segons transmissió sigui automàtica (0) o manual (1). Per tant també es pot categoritzar amb *factor*.

```
>mtcars2 <- within(mtcars2, {
  am <- factor(am, labels = c("automatic", "manual"))
})
> ggplot(mtcars2, aes(cyl, fill=am))+ geom_bar()+xlab("Number of
Cylinders")+ylab("Count")
```

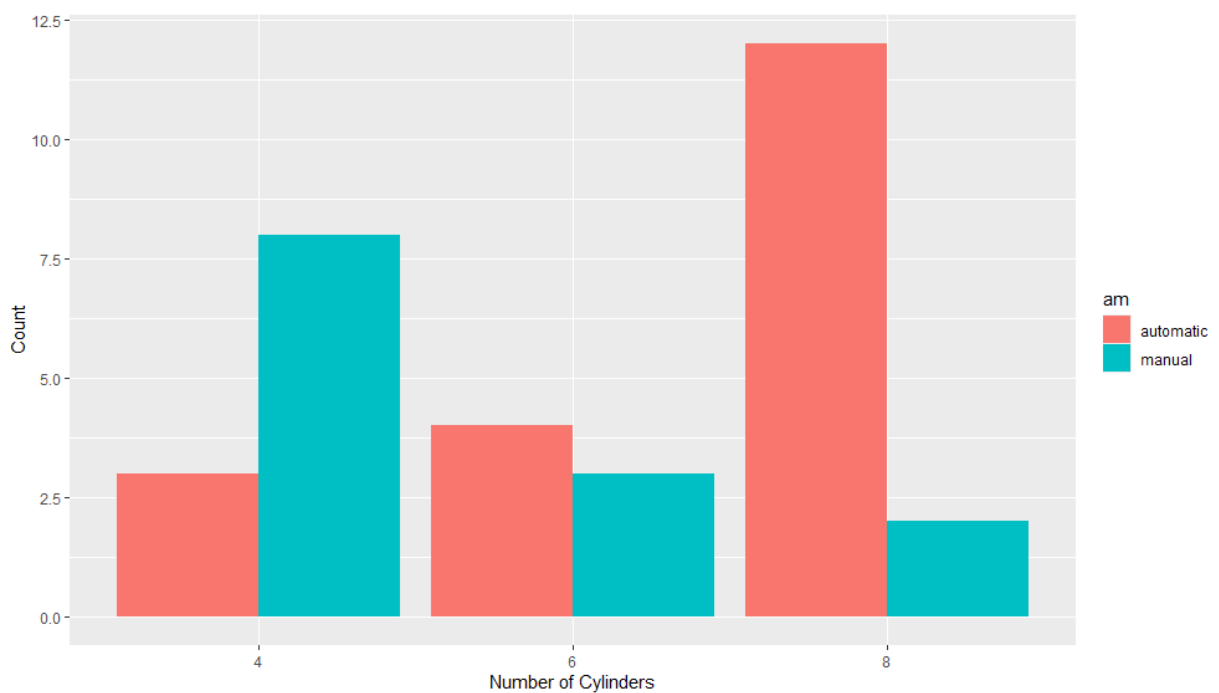


Sembla que la major part dels cotxes de 4 cilindres tenen una transmissió manual, en el cas de 6 cilindres ja tenim algun cotxe més amb transmissió automàtica que manual i en el cas de 8 cilindres, encara més. També veiem globalment que tenim pocs cotxes amb 6 cilindres, i dels que més en tenim són dels de 8 cilindres.

Però, aquesta gràfica no ens mostra la informació que volem d'una forma simple. Fem el que ens diu l'enunciat utilitzant l'argument posició a veure si ens ajuda. Per això, especifiquem, com ens diu l'enunciat, **geom_bar(position="dodge")**:

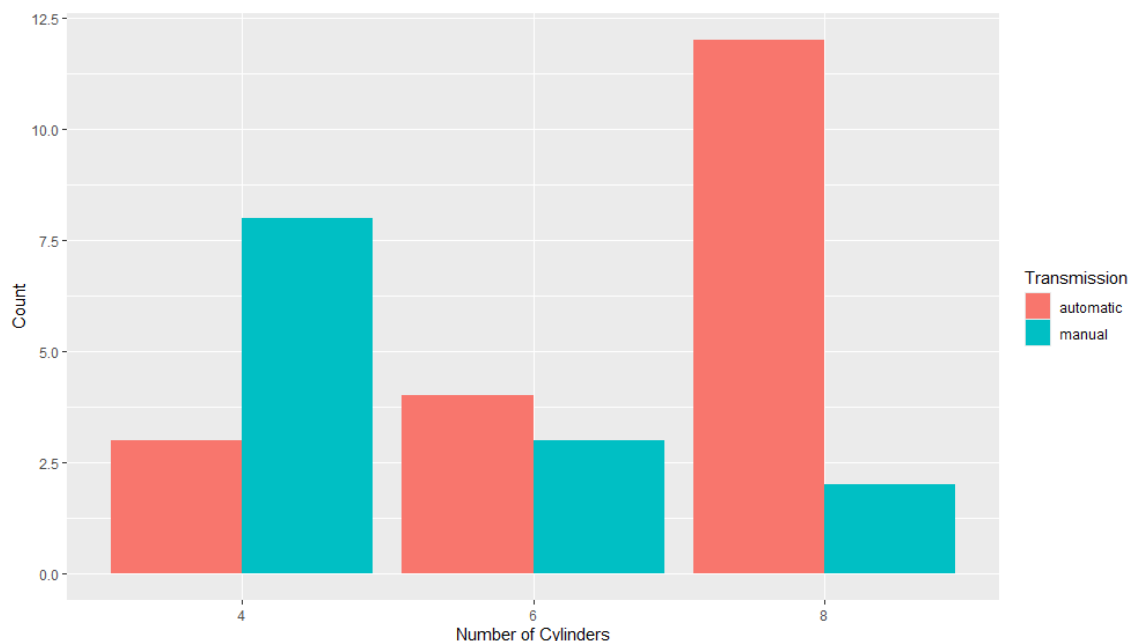
```
> g<-ggplot(mtcars2, aes(cyl, fill=am))+  
geom_bar(position="dodge")+xlab("Number of Cylinders")+ylab("Count")
```

```
>g
```



I podem posar títol a la llegenda per fer-ho més fàcil d'interpretar:

```
> g+scale_fill_discrete("Transmission")
```



Com ens deia l'enunciat: la transmissió apareix en l'eix x, una al costat de l'altra d'acord amb el nombre de cilindres de cada cotxe. La gràfica és molt més clara que abans. Ara comparar les proporcions entre cotxes (automàtics o manuals) per un cert nombre de cilindres es pot fer d'una manera molt més clara, i sense perdre la informació global. *Efectivament veiem que la transmissió automàtica (en vermell) creix notablement a mesura que augmentem els cilindres (i exponencialment quan passem de '6' a '8' cilindres). A la vegada, la transmissió manual (en blau), decreix notablement quan els cilindres dels cotxes augmenten. En el cas dels cotxes amb 8 cilindres (columnes més a la dreta), per exemple, la quantitat de cotxes automàtics (vermell) és més de 4 vegades que la de manuals (blau).*

NOTA: Podeu veure més arguments en

https://ggplot2.tidyverse.org/reference/geom_bar.html

Ara que hem vist com comparar algunes proporcions mitjançant diagrames de barres, veiem les distribucions a la part 2 del seminari.

3. PART 2. Distribucions

EXERCICIS:

1.- Mostreu la distribució de la variable 'hp'. Quina informació en podeu extreure?

Avui hem vist just dues maneres de mostrar distribucions, via histogrames, polígons de freqüència i diagrames de caixes (boxplots). Hem vist que els histogrames i els polígons de freqüències eren útils quan volíem mostrar la distribució d'una funció contínua com és el cas de 'hp'.

Si feu:

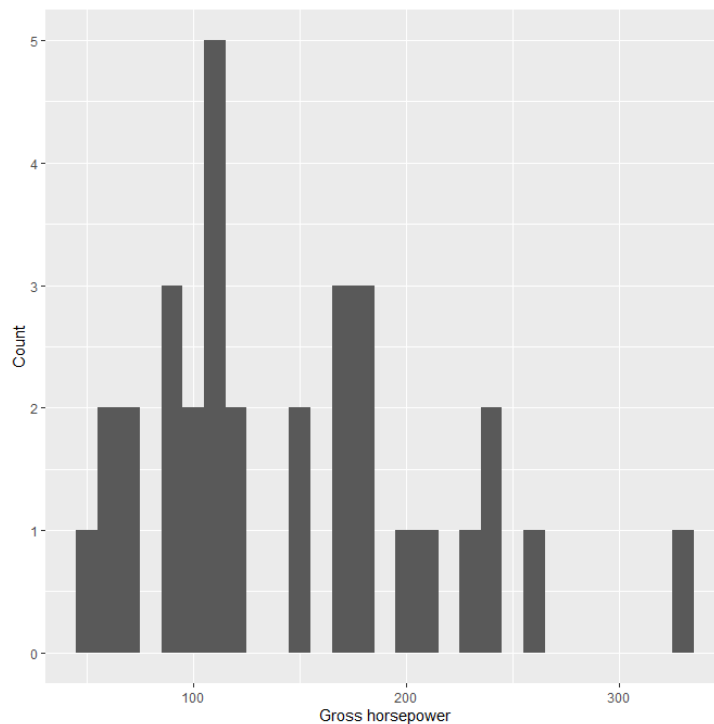
```
> ggplot(mtcars, aes(hp))+geom_histogram()+xlab("Gross Horsepower")+  
ylab("Count")
```


ATENCIÓ: R ens retorna un error:

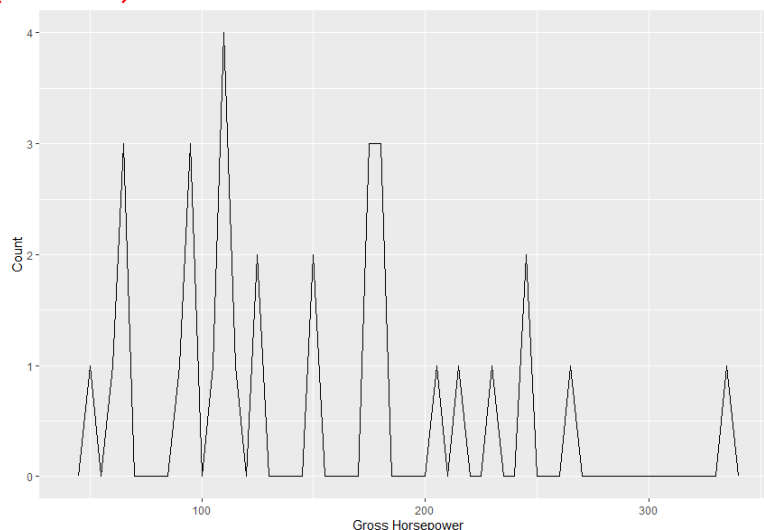
``stat_bin()` using `bins = 30`. Pick better value with `binwidth`.`

Com dèiem abans hem d'especificar l'argument *binwidth*. Exemple:

```
> ggplot(mtcars, aes(hp))+geom_histogram(binwidth=10)+xlab("Gross  
Horsepower")+ylab("Count")
```



```
>ggplot(mtcars, aes(hp))+geom_freqpoly(binwidth=5)+xlab("Gross Horsepo  
wer")+ylab("Count")
```



Sembla que tenim poques dades, veiem que hi ha 5 elements amb una potencia bruta de 100 o cap al voltant de 300. Però en general, tenim quelcom dispers

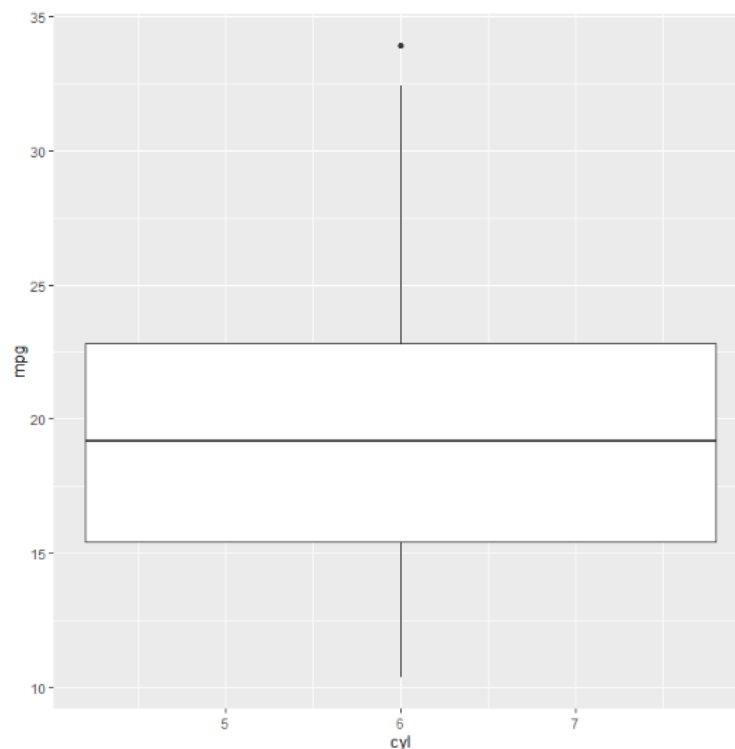
2.- Feu ús de `geom_boxplot()` per fer un gràfic en que l'eix de les x correspongui a la variable 'cyl' (cilindres) i l'eix y a la variable 'mpg' (milles de galó). Un cop

tingueu el gràfic, compareu aquesta gràfica amb la resultant de l'exercici 2 del seminari 1 (Quina informació tenim aquí que no teníem abans?). A més:

- Poseu la llegenda dels eixos utilitzant `xlab()` i `ylab()`
- Gireu els eixos de coordenades, fent que els boxplots quedin en horitzontal utilitzant la comanda `coord_flip()`
- Compareu la gràfica resultant amb la gràfica de l'exercici 2 seminari 1.

Si fem directament:

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



R ens retorna aquesta gràfica i un avis:

Warning message:

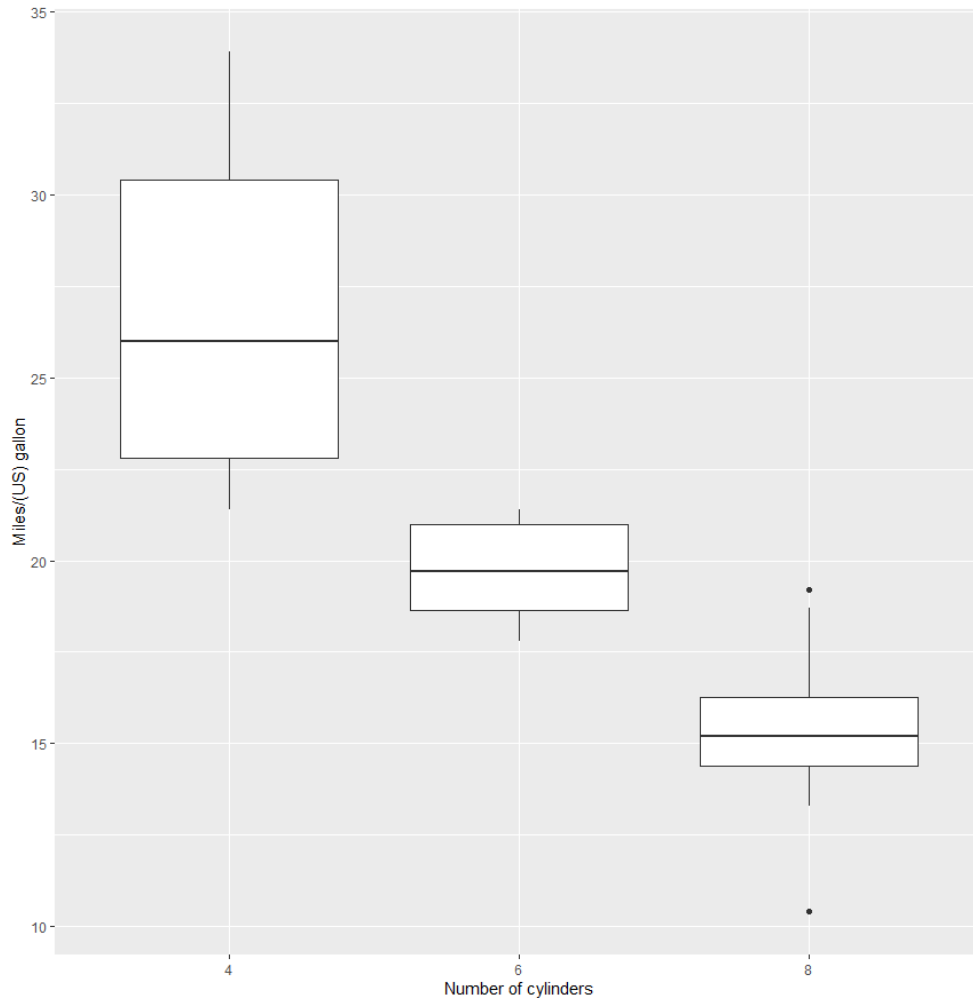
Continuous x aesthetic -- did you forget aes(group=...)?

Com hem vist a classe, quan tenim dues variables, els boxplots són útils quan volem observar la distribució de dues variables, on una de les variables és discreta i l'altra variable és contínua. Ja hem vist en la Part 1 d'aquest seminari, i en el seminari 1, com fer que aquesta variable sigui tractada de forma discreta, **categoritzant-la amb factor/ordered**. Per tant usem directament `mtcars2` (o veiem en la Part 1 com crear-lo):

```
> ggplot(mtcars2, aes(x=cyl, y=mpg)) + geom_boxplot()
```

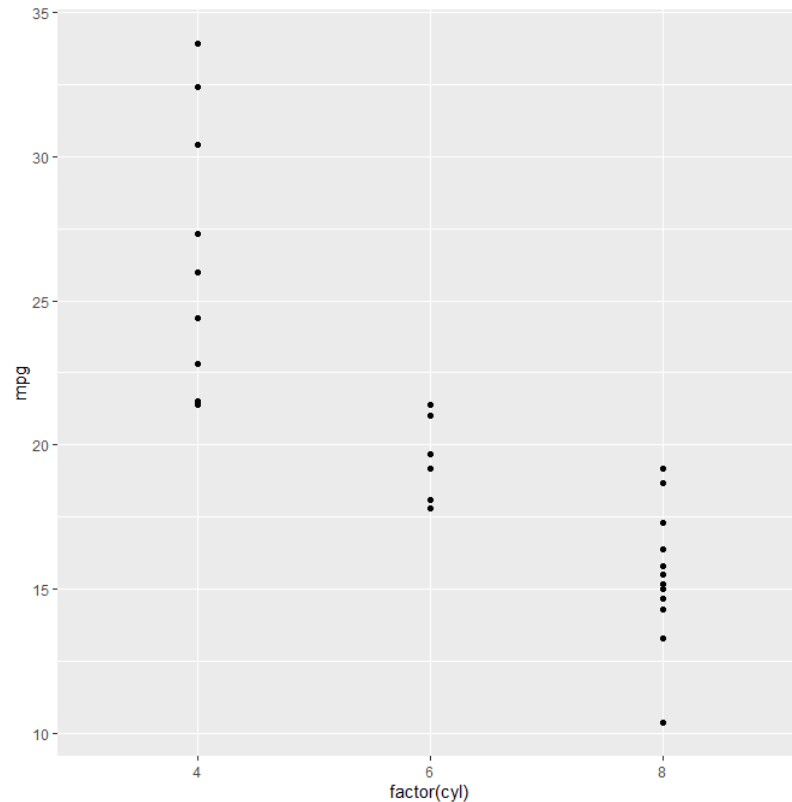
a) Com en l'exercici 1 de la part 1, fem us de `xlab` i `ylab` per posar noms als eixos de la gràfica

```
> ggplot(mtcars2, aes(x=cyl, y=mpg)) + geom_boxplot() +  
xlab('Cylinders') + ylab('Miles/(US) gallon')
```



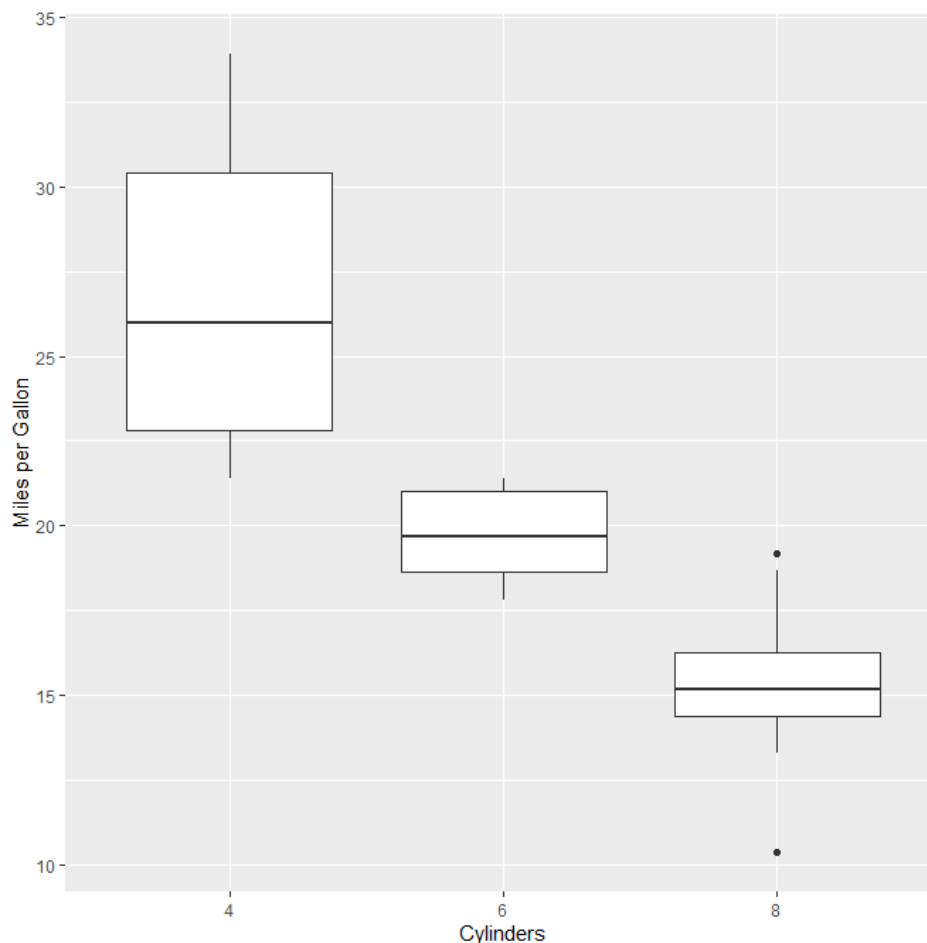
PRIMER FEM l'apartat c

c) Si comparem amb la gràfica de l'exercici 2 del seminari 1 (a sota), tot i que ja podíem intuir alguna informació preliminar sobre la distribució de les nostres dades en la gràfica de punts (scatter plot), no era tant evident com en el boxplot que acabem de fer (i en perdíem informació). Això es deu al fet que la variable cilindres és qualitativa. **Així com en el cas de variables quantitatives els *scatter plots* ens donen molta informació, no ens en donen tanta per a variables qualitatives.**



Gràfica de l'exercici 2 seminari 1

OBSERVACIONS que ens aporta aquest nou gràfic: Ara podem veure per exemple ràpidament que: i) la mediana de milles de galó dels cotxes amb 8 cilindres és al voltant de 15, mentre que la de 6 cilindres és al voltant de 20, i puja fins més de 25mpg en el cas dels cotxes amb 4 cilindres (el qual tindria sentit); ii) tenim outliers en el cas dels cotxes amb 8 cilindres (representats amb dos punts fora del boxplot); iii) els quartils Q1 i Q3 i per tant la distribució (és molt més ampla en termes de 'mpg'), varia més en el cas dels cotxes de 4 cilindres, per exemple; iv) la desviació, casi és nul·la pels cotxes de sis cilindres per exemple (gairebé no tenim bigotis).



b) Com sempre que no hem utilitzat prèviament un comandament fem servir `?coord_flip` per a que l'ajuda de R ens digui com podem utilitzar-lo

```
R Help on 'coord_flip'
Archivo  Editar

limits, then those data points may show up in places such as
the axes, the legend, the plot title, or the plot margins.

Examples:

# Very useful for creating boxplots, and other interval
# geoms in the horizontal instead of vertical position.

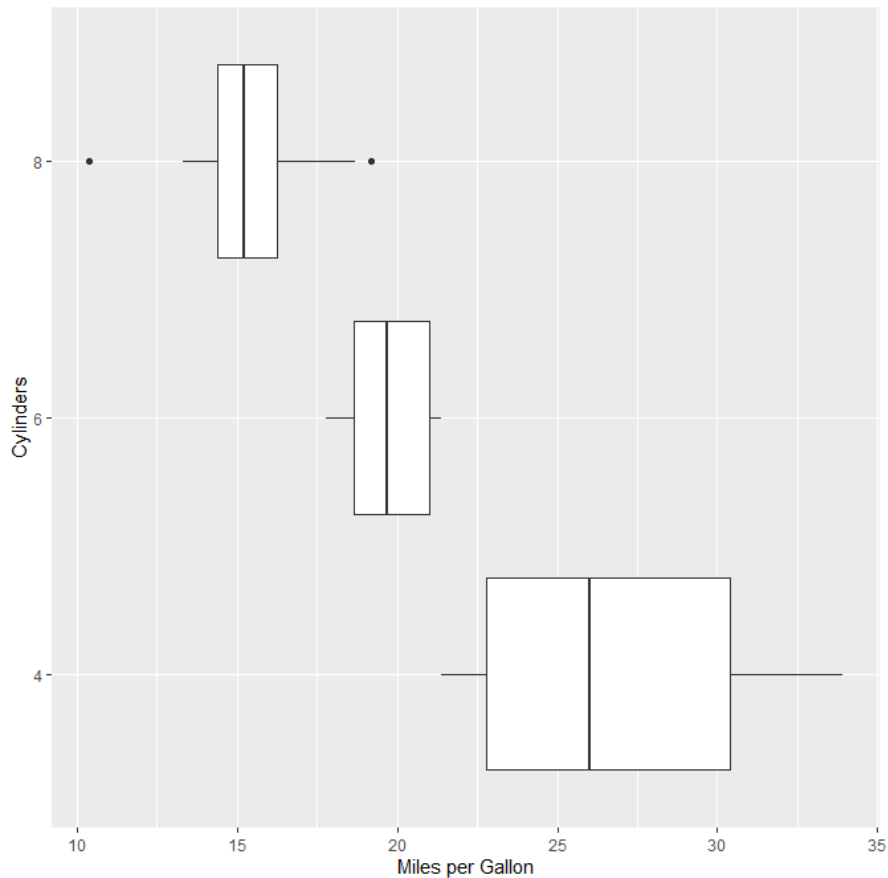
ggplot(diamonds, aes(cut, price)) +
  geom_boxplot() +
  coord_flip()

h <- ggplot(diamonds, aes(carat)) +
  geom_histogram()
h
h + coord_flip()
h + coord_flip() + scale_x_reverse()

# You can also use it to flip line and area plots:
df <- data.frame(x = 1:5, y = (1:5) ^ 2)
ggplot(df, aes(x, y)) +
  geom_area()
last_plot() + coord_flip()
```

Farem:

```
> ggplot(mtcars2,aes(x=cyl,y=mpg)) + geom_boxplot() +  
xlab('Cylinders') + ylab('Miles per Gallon')+coord_flip()
```



Però, com abans, per no haver de repetir cada vegada la línia de codi, podem assignar la nostra gràfica a una variable i afegir comandes a fer mesura, obtenint el mateix resultat. Escriure'm:

```
> my_boxplot <- ggplot(mtcars2,aes(x=cyl,y=mpg))+geom_boxplot()  
+xlab('Cylinders')+ylab('Miles per Gallon')  
> my_boxplot+coord_flip()
```

Escolliu en cada cas la manera de treballar que us sigui més còmoda.

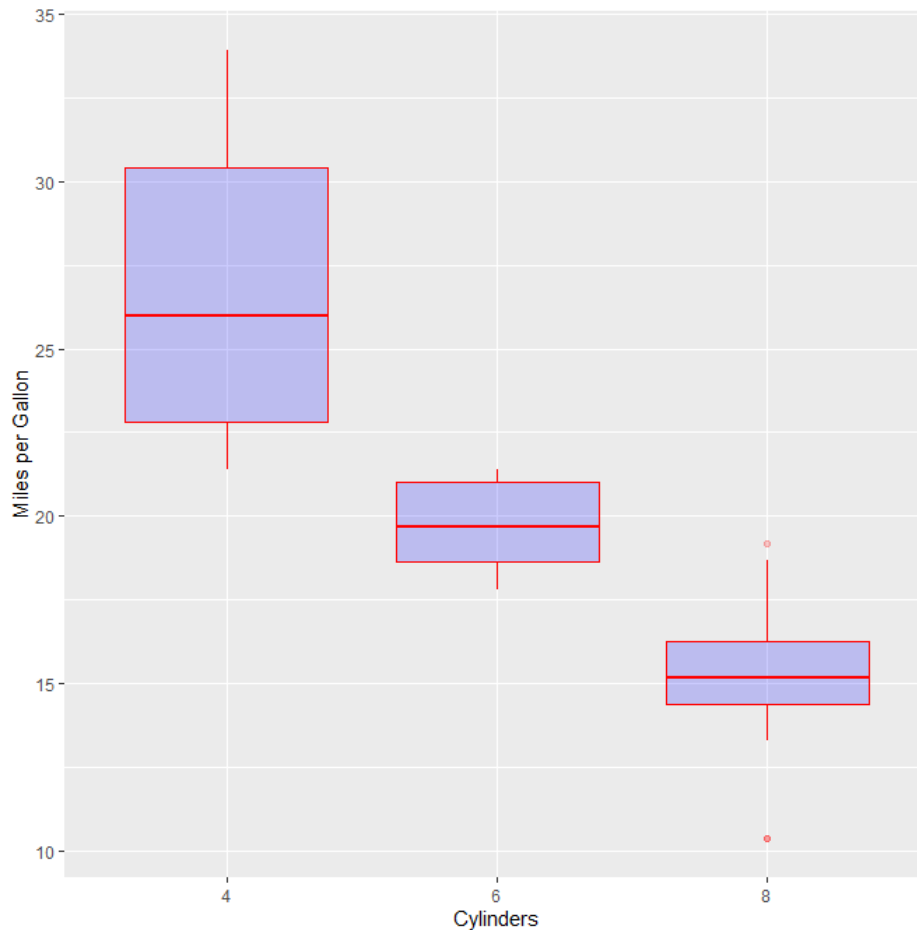
d) Pinte els boxplots de tres maneres diferents:

d1) Escollint un únic color amb fill, color i alpha per als tres grups:

```
geom_boxplot(color="red", fill="blue", alpha=0.2)
```

Fixeu-vos en el resultat de canviar els tres paràmetres *color*, *fill*, *alpha*. Què fa cadascun?

```
> ggplot(mtcars2,aes(x=cyl,y=mpg)) + geom_boxplot(color="red",  
fill="blue", alpha=0.2) + xlab('Cylinders') + ylab('Miles per Gallon')
```



Color canvia el color referent als contorns del boxplot, *fill* el color de dins, i *alpha* ens varia l'opacitat del color amb el que pintem.

!REMARCA: Fixem-nos que estem utilitzant els arguments de `geom_boxplot` (com hem fet abans amb els arguments *width* o *position* en `geom_bar()`). No és que estem fent un mapeig del color per cada grup com fèiem amb `aes()` en el seminari 1 o abans en l'exercici 1 de la part 1.

d2) De manera que el color es trobi en el contorn del boxplot per a cada grup (cotxes amb els mateixos cilindres)

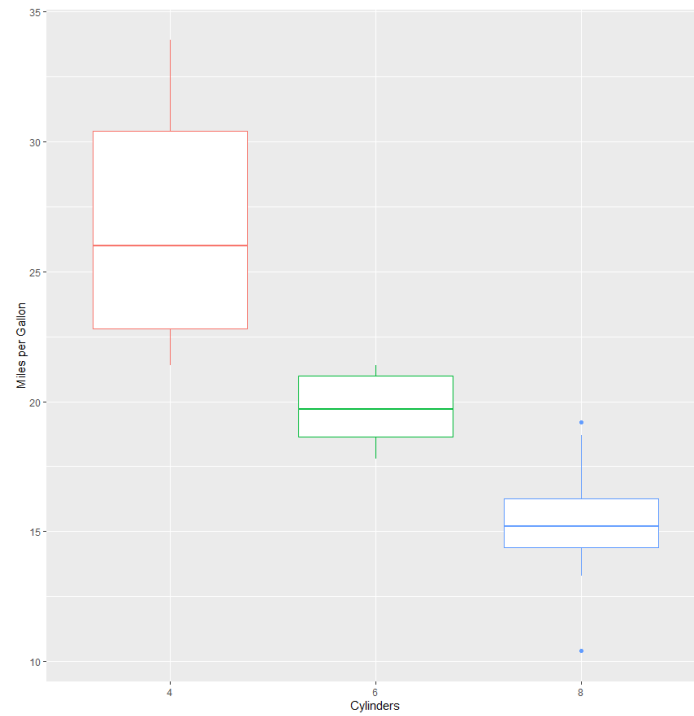
d3) De manera que el color es trobi en l'interior del boxplot per a cada grup (cotxes amb els mateixos cilindres)

Els canvis fets en d2) i d3) ens aporten alguna informació nova?

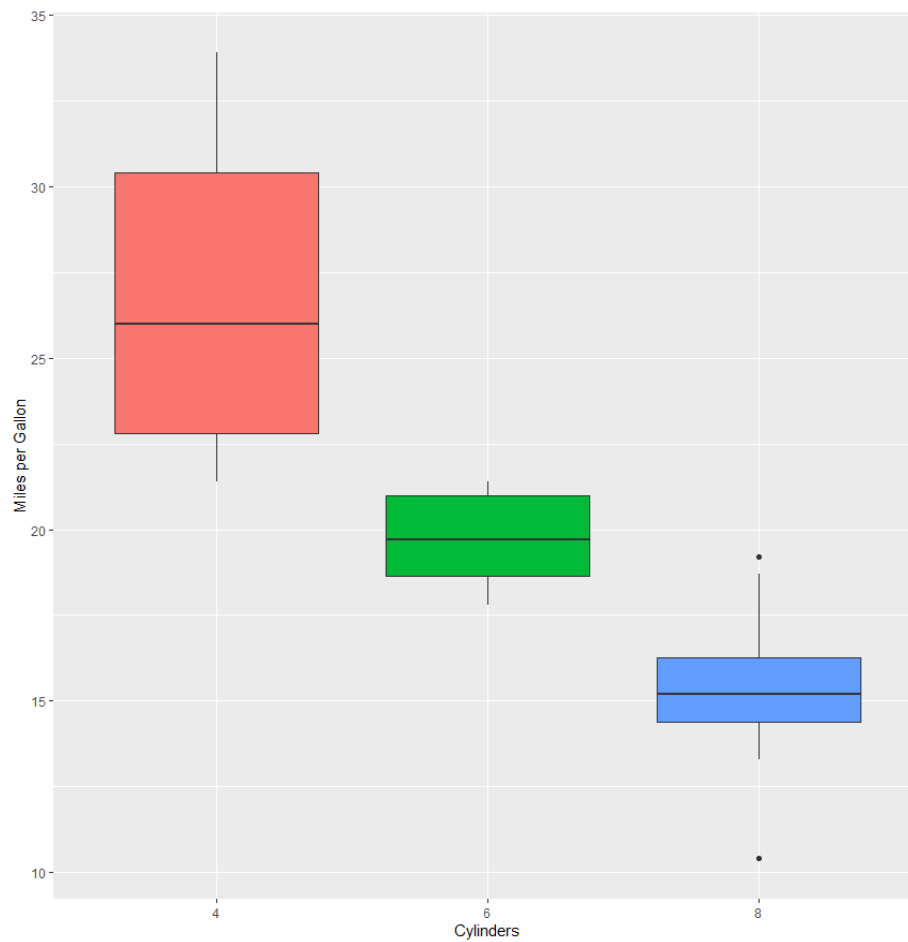
NOTA: Per d2) i d3) al pintar diferenciant per grups recordeu que afegim la informació dins d'`aes()` com ja havíem fet en el seminari 1.

Seguim el que ens diu la nota, especificant dins `aes()` la variable que volem agrupar, tal i com fèiem en el seminari 1.

```
d2) > ggplot(mtcars2,aes(x=cyl,y=mpg, color=cyl)) + geom_boxplot()
+   xlab('Cylinders')      +   ylab('Miles per Gallon')+
guides(color="none")
```



```
d3) > ggplot(mtcars2,aes(x=cyl, y=mpg, fill=cyl)) + geom_boxplot()
+   xlab('Cylinders') + ylab('Miles per Gallon') +
+   guides(fill="none")
```

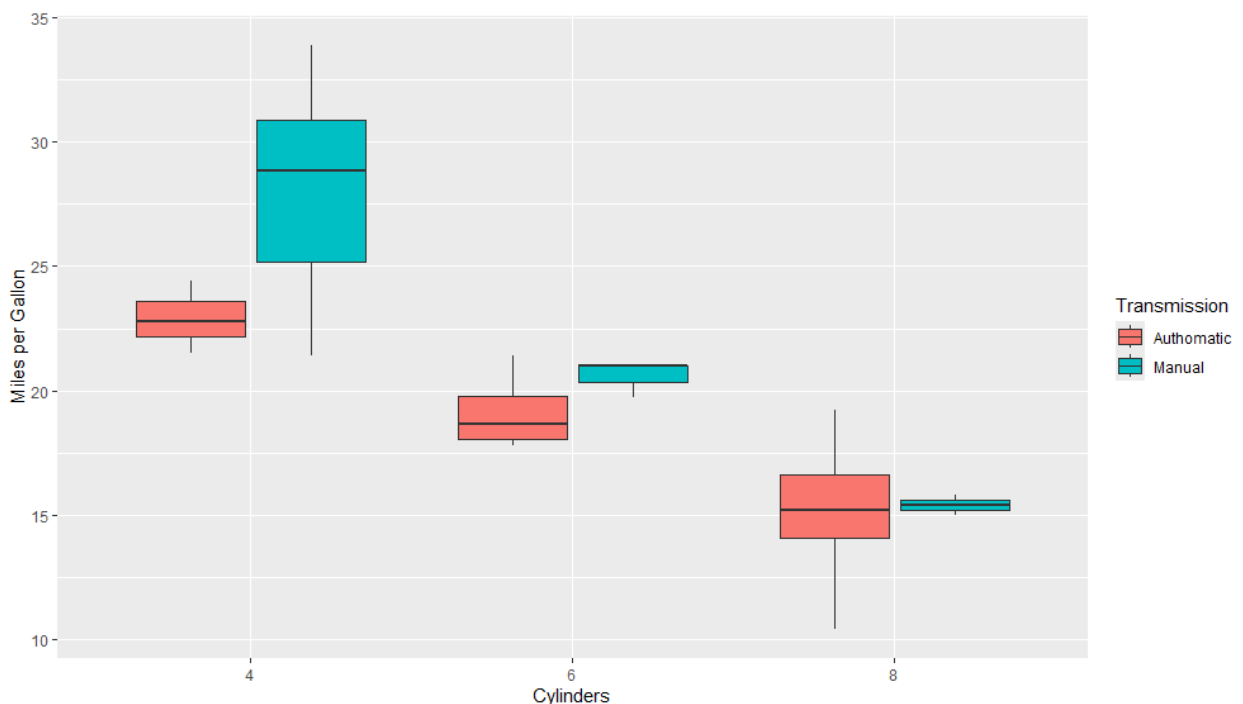


De la mateixa manera que havíem vist en el seminari 1, *pintar els boxplots segons el grup del nombre de cilindres al que pertanyen, no ens aporta cap informació nova respecte el mateix boxplot sense color.*

El mapeig de color per grups via `aes()`, ens podria aportar informació si afegim en el mateix gràfic el color als grups generats per una variable discreta com pot ser el tipus transmissió del cotxe (`factor(am)`) o la forma del motor (`factor(vs)`). (Si ho féssim amb `vs`, recordem que hauríem de categoritzar-la com hem fet amb `am`).

Exemple:

```
> ggplot(mtcars2, aes(x=cyl, y=mpg, fill=am)) + geom_boxplot() +  
  xlab('Cylinders') + ylab('Miles per Gallon') +  
  scale_fill_discrete("Transmission")
```



I com sempre, per facilitar la visualització posem títol a la llegenda.

3.- Mostreu en un mateix gràfic un polígon de freqüències i un histograma del temps en quarts de milla (*qsec*). Afegiu títol i etiquetes als eixos. Poseu un color per cada geometria. Què es mostra? NOTA: Ajusteu els bins.

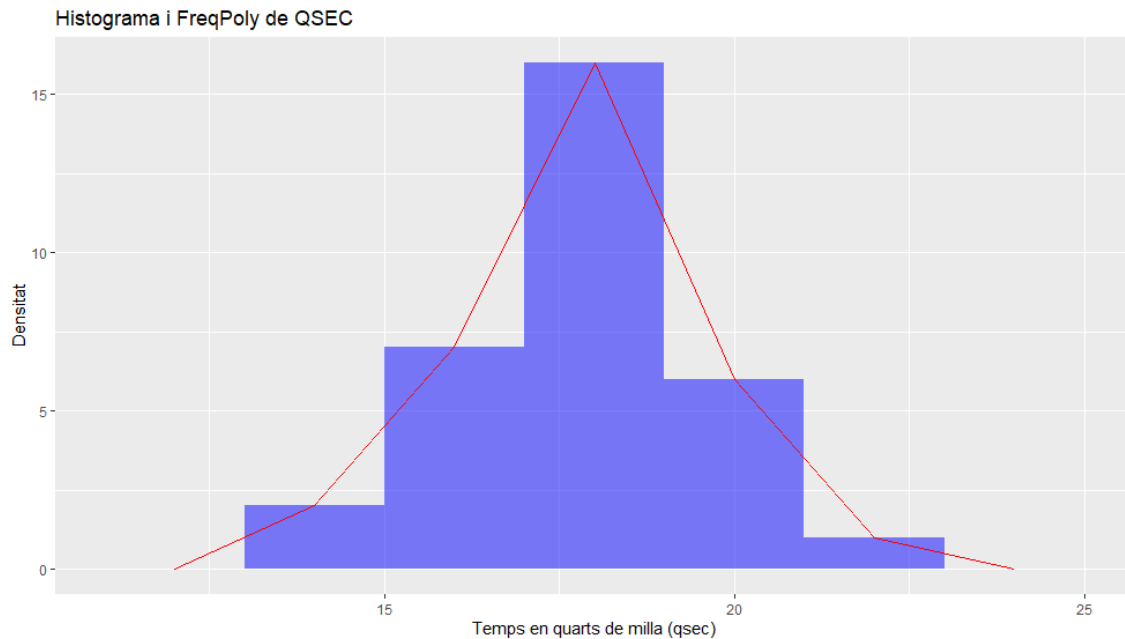
Un cop tingueu el gràfic afegiu la capa `theme_minimal()`, què us fa?

Ja hem vist que podem anant sumant capes, podem usar colors i canals alpha (que ens donen transparència, si volem).

Aquí l'exemple està fet usant `labs` per posar etiquetes als eixos i títol (per veure noves opcions). Podeu veure com s'utilitzen fent `?labs`.

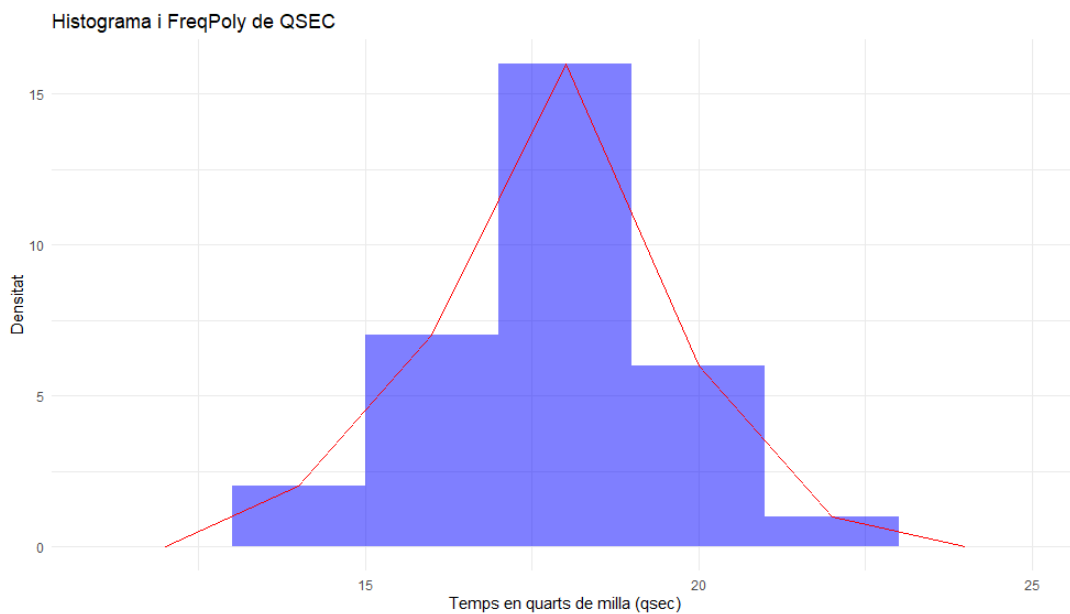
```
> p <- ggplot(mtcars2, aes(x = qsec)) + geom_histogram(binwidth = 2,  
  fill = "blue", alpha = 0.5) + geom_freqpoly(binwidth = 2, color =
```

```
"red") +labs(title = "Histograma i FreqPoly de QSEC", x = "Temps en  
quarts de milla (qsec)",y = "Densitat")
```



Veiem que tot i baixar el binwidth a 2, hi ha poques dades. El que podem dir és que l'histograma mostra una moda clara per exemple al voltant de 17-18 qsec, per exemple.

```
>p + theme_minimal()
```



La funció `theme_minimal()` en `ggplot2` aplica un tema visual minimalista al gràfic. Concretament:

- Elimina els fons de color i els marcs ombrejats.
- Usa una graella lleugera i línies fines per fer el gràfic més net.
- Evita elements innecessaris, fent que el gràfic sigui més llegible i elegant.

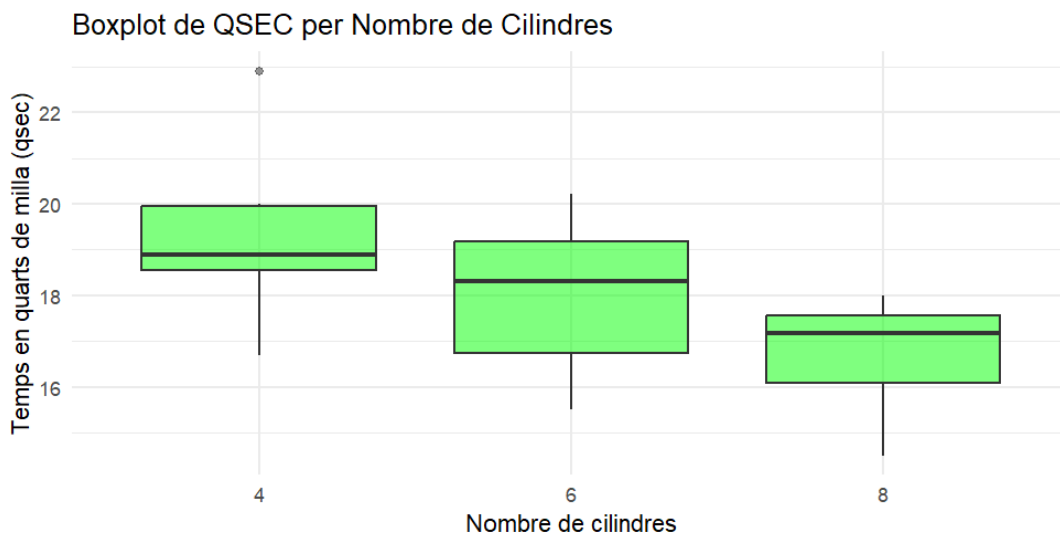
Si vols provar altres estils, també hi ha `theme_classic()`, `theme_light()`, `theme_dark()`, entre d'altres.

4.- Com depèn la distribució del temps en quarts de milla (*qsec*) segons la cilindrada? Feu una imatge que permeti visualitzar-ho. Imagineu que heu de posar aquesta gràfica en una web corporativa de tons verds, poseu-hi un color verd per tal que estèticament quedi més en el context de la pàgina web.

Ara tenim una variable contínua i una variable discreta categòrica amb 3 categories, o fem tres histogrames, o si ho volem posar en un gràfic tot fem un boxplot.

Ja hem vist com categoritzar la variable ordinal *cyl* (si no s'ha fet veure PART1). També hem vist en la PART 2 com pintar els boxplots. Per tant, ajuntant tot el que hem après podríem posar, per exemple:

```
>ggplot(mtcars2, aes(x = cyl, y = qsec)) + geom_boxplot(fill =  
"green", alpha = 0.5) + labs(title = "Boxplot de QSEC per Nombre de  
Cilindres", x = "Nombre de cilindres", y = "Temps en quarts de milla  
(qsec)") + theme_minimal()
```



Veiem que tenim només un outlier pel cas de cotxes de 4 cilindres, i les distribucions per cada cilindrada varien bastant. Pel cas dels cotxes de 4 cilindres podem dir que el temps en *qsec* té una distribució asimètrica negativa, pel cas de 6 cilindres hi ha una mínima asimetria positiva, tot i que està prou centrada, i, finalment, pel cas de 8 cilindres hi ha una clara asimetria positiva