Escola d'Enginyeria UAB

UAB

SEMINARI 6. *Advanced Systems II*

1. OBJECTIUS

Aquest seminari està enfocat a Sistemes Avançats II, en ell veure'm de manera pràctica algunes de les eines que vam veure a la classe de teoria de sistemes avançats I.

A la primera part, farem servir una de les tècniques que vam veure per mostras varies proporcions de cop i aprendrem a visualitzar-la : el *treemaps*.

A la segona part, veurem com fer un correlograma que ens permet veure la correlació entre varies variables numèriques.

En la tercera part, finalment, veurem un scatter plot matrix (SPLOM) dels que vam veure també a teoria. Aquest gràfic, com vam veure, ens permet veure també correlacions entre parelles de variables d'un dataframe amb múltiples variables (però no gaire gran).

Recordeu carregar les llibreries necessàries com sempre.

2. PART 1. *Advanced systems II*

En aquesta primera part, com ja hem anunciat a l'inici, veurem una eina que vam veure en teoria, i hem repassat just abans, els *treemaps*. Com sempre mireu les llibreries i paquets que necessiteu abans de començar.

EXERCICIS:

1.- Utilitzant el que es va explicar sobre treemaps en R (inclòs com recordatori en les transparències d'avui), se us demana fer un treemap utilitzant el dataframe mtcars. On:

- **L'àrea de les rajoles/caselles (tiles) i la seva 'etiqueta' vingui donada pel desplaçament (*disp*) de cada cotxe.**
- **El color de cada 'pare' vingui donat pel país d'origen del cotxe. És a dir per la variable que hem creat abans (`mtcars.country`)**
  Feu ús de: `mtcars.country <- c(rep("Japan", 3), rep("US",4), rep("Europe", 7),rep("US",3), "Europe", rep("Japan", 3), rep("US",4), rep("Europe", 3), "US", rep("Europe", 3))`

a) Prepareu el mapeig (*aes*) del vostre ggplot i afegiu la geometria adient que hem vist per fer treemaps

Seguint :

1.1. Visualizing many proportions at once

2. In order to create a treemap, the data must be converted to desired format using treemapify(). The important requirement is, you need to identify in your data:
   - One variable each describes the area of the tiles ('value')
   - One variable for fill color ('parent')
   - One variable that has the tile's label ('id')
   - The parent group ('parent')

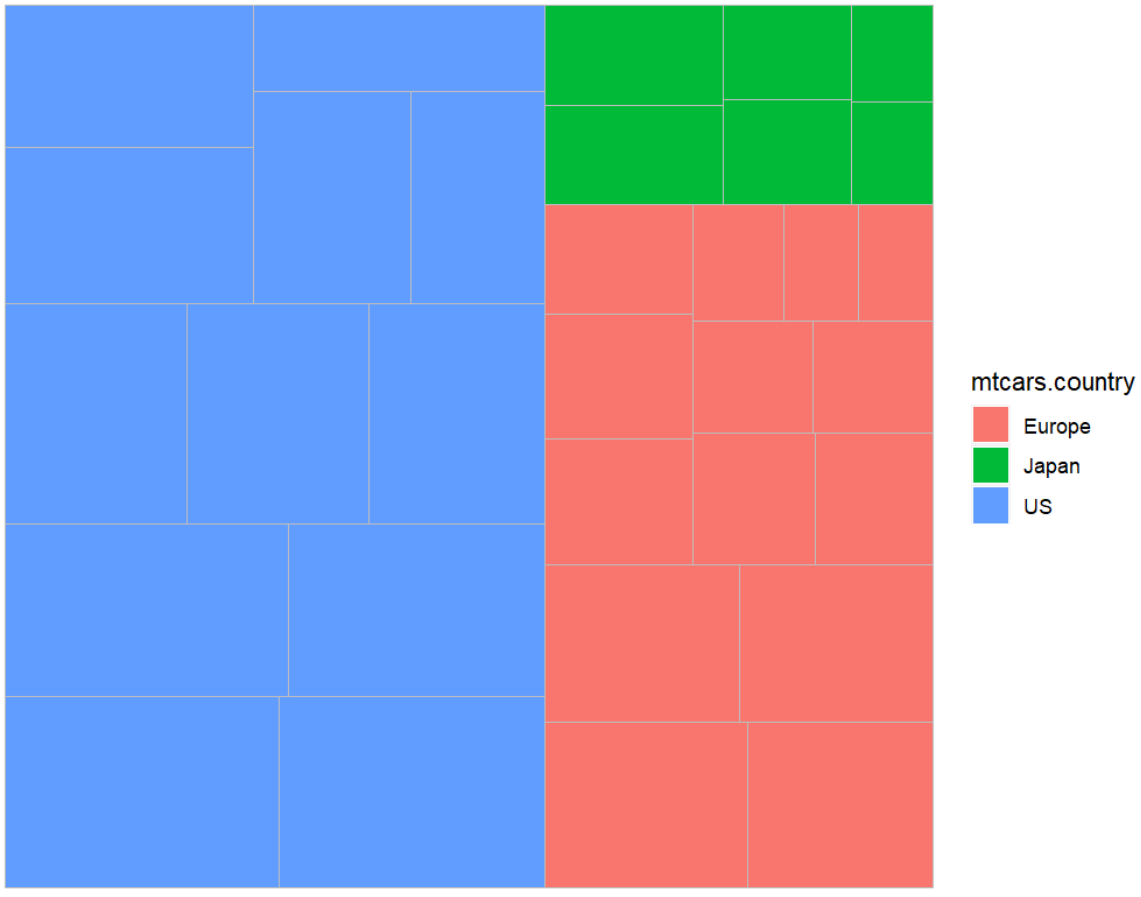
   And map them by using aes:

```
> ggplot(Proglangs, aes(area=value, fill=parent,
label=id, subgroup=parent)) + geom_treemap()
```

3. Use geom_treemap

En el nostre cas tenim:

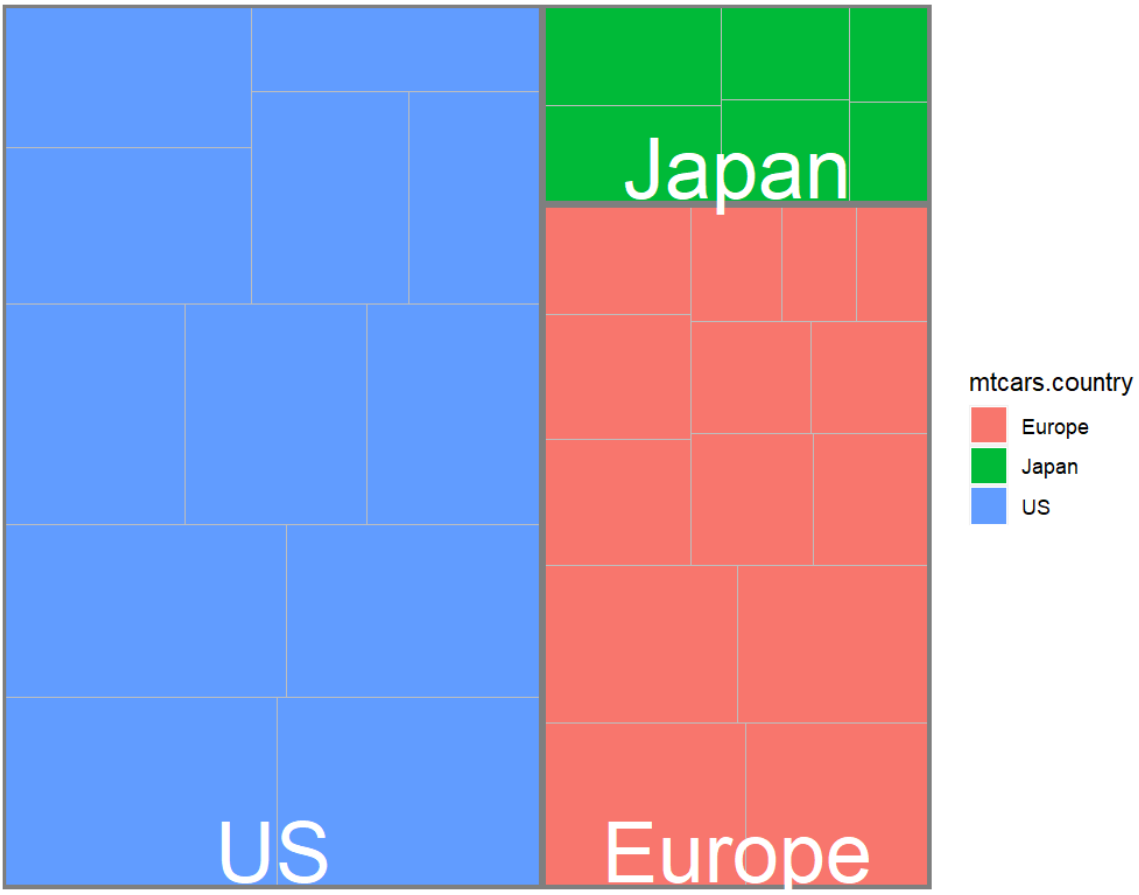
> ggplot(mtcars, **aes(area=disp, fill=mtcars.country, subgroup=mtcars.country))+geom_treemap()**



b) Poseu el contorn de cada subgrup i la etiqueta/text del mateix en blanc

Seguint les diapositives 9,10 i 11 de la primera part de la classe d'avui:

> ggplot(mtcars, aes(area=disp, fill=mtcars.country, subgroup=mtcars.country))+geom_treemap()+ **geom_treemap_subgroup_border() +geom_treemap_subgroup_text(color='white')**



c) Ara que ja teniu l'estructura i etiquetats als 'pares'. Creeu una variable `mtcars.id` seguint:

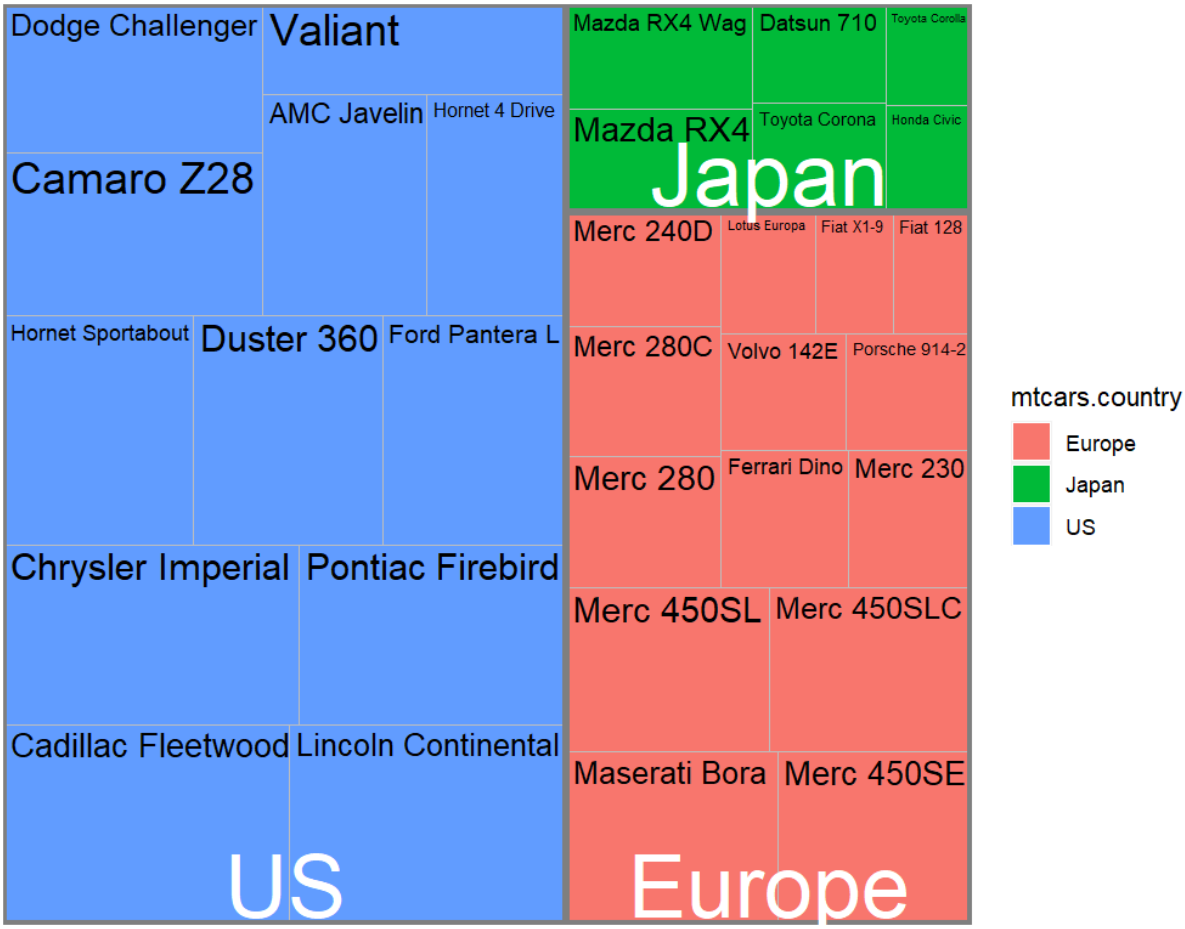
`> mtcars.id=row.names(mtcars)`

Fixeu-vos que us agafa els noms de les marques dels cotxes (files del dataframe)

Afegiu en negre el text d'aquesta variable en el vostre treemap.

Seguint la diapositiva 12 de la primera part de la classe:

> ggplot(mtcars, aes(area=disp, fill=mtcars.country, subgroup=mtcars.country))+geom_treemap()+ geom_treemap_subgroup_border() +geom_treemap_subgroup_text(color='white')+**geom_treemap_text (aes(label=mtcars.id))**



d) Podeu treure la llegenda amb `theme(legend.position = "none")`

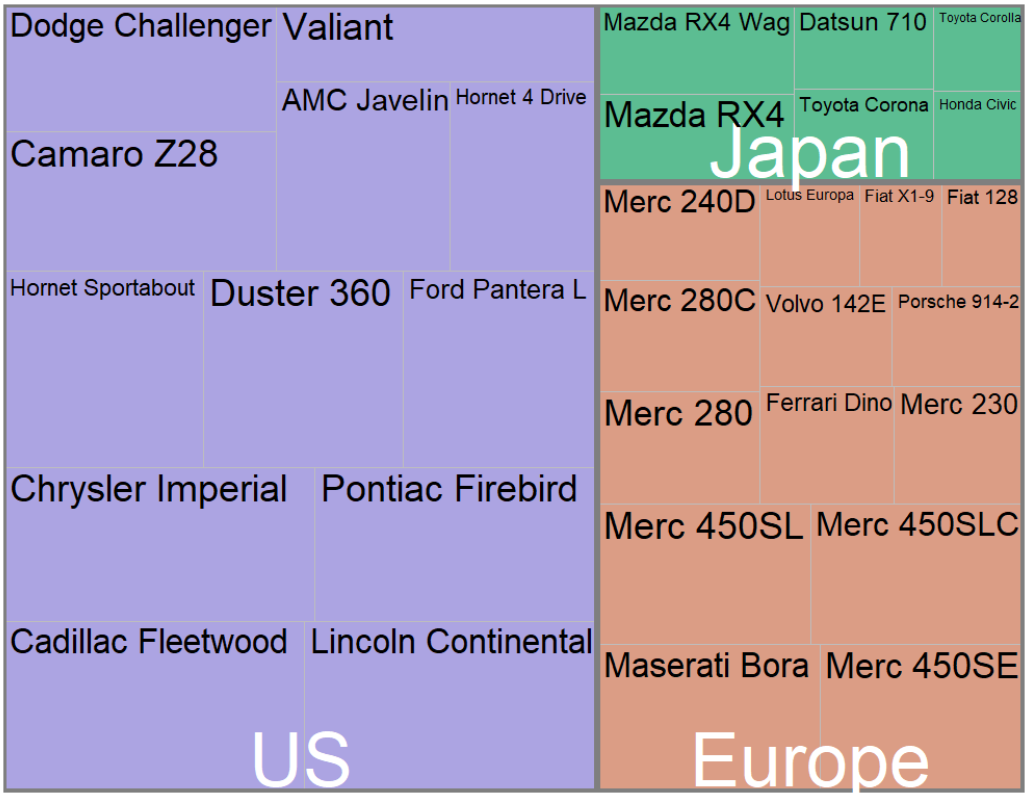
Milloreu l'edició amb colors menys cridaners. Feu ús de

```
? scale_fill_discrete_qualitative
```

Proveu aquestes paletes per exemple, que funcionen bé amb aquest layer: `Pastel 1, Dark 2, Dark 3, Set 2, Set 3, Warm, Cold, Harmonic, Dynamic`

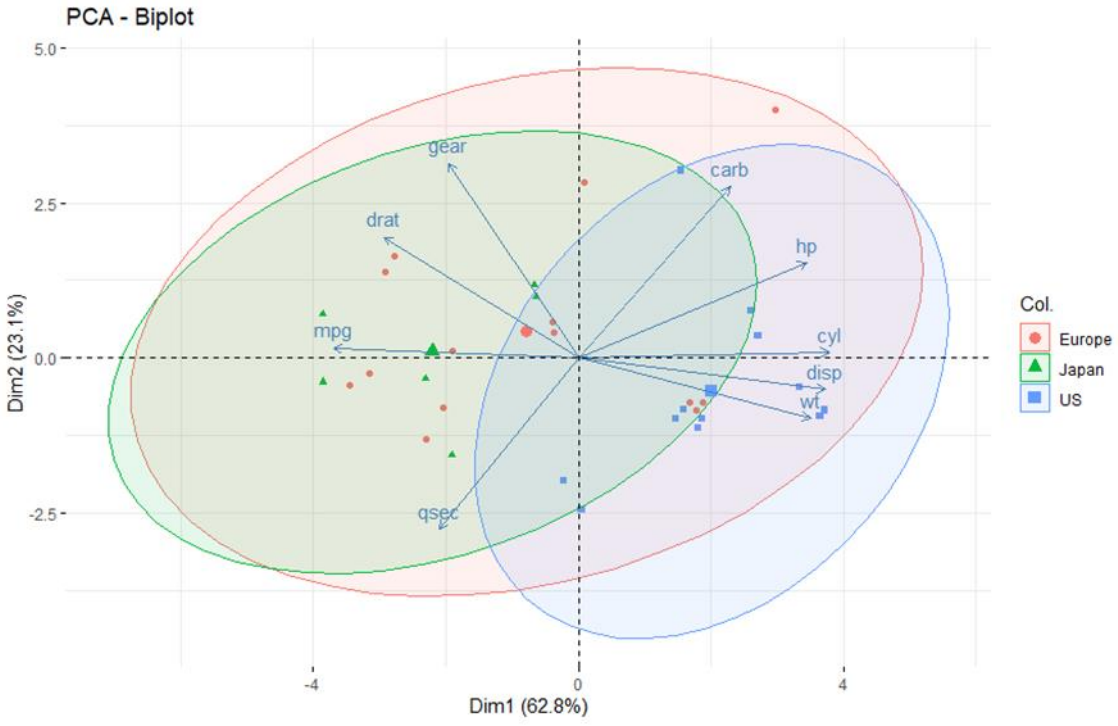
```
> ggplot(mtcars, aes(area=disp, fill=mtcars.country, subgroup=mtcars.country))+geom_treemap()+ geom_treemap_subgroup_border() +geom_treemap_subgroup_text(color='white')+geom_treemap_text (aes(label=mtcars.id))+scale_fill_discrete_qualitative(palette = "Dynamic")+theme(legend.position = "none")
```

Si no us funciona: `scale_fill_discrete_qualitative`, feu ús de `scale_fill_brewer`



També podríeu millorar-lo fent una lletra més petita en algunes etiquetes, posant un títol millor en la llegenda, si la deixeu, etc. La idea del seminari d'avui era aprendre a fer treemaps, però podeu fer proves per personalitzar el vostre gràfic.

e) Quines conclusions en podeu extreure? Observeu alguna equivalència amb el PCA que vam fer en l'exercici pràctic de la última classe de teoria?



Possibles conclusions: Els cotxes japonesos clarament tenen un desplaçament menor. Els de US fan un desplaçament molt major.

Equivalència amb PCA: En el PCA ja havíem vist que els cotxes nord-americans es caracteritzen per tenir valors elevats de *cyl*, *disp* i *wt*. En aquest treemap hem definit l'àrea de les rajoles/caselles segons el desplaçament (*disp*). Podem veure altre cop que els cotxes d'US són els que fan un desplaçament major.

3. PART 2. Correlograma

En aquesta segona part del seminari, anem a veure algunes eines de les que vam veure a teoria per veure relacions entre variables numèriques quan tenim moltes variables. Concretament veurem com fer un Correlograma. Farem servir el *dataframe* de R *mtcars*, però sense modificar. És a dir, el dataframe que conté 32 observacions i 11 variables sobre cotxes, sense NA's. Torneu-vos a familiaritzar amb el dataframe (str(mtcars)).

EXERCICIS:

1.- Volem fer un correlograma utilitzant el *dataframe mtcars* que ens mostri la correlació entre les sis variables:

- **mpg: quilòmetres per galó (EUA)**
- **disp: Desplaçament**
- **hp: potència bruta**
- **drat: relació dels eixos posteriors**
- **wt: pes (1000 lliures)**
- **qsec: 1/milla de temps**

a) Per això comenceu convertint el vostre *dataframe* a una *tibble* de nom *df_tb*

Com vam veure a l'últim seminari, podem fer la transformació utilitzant as.tibble de la llibreria tidyr.

```
> df_tb <- as_tibble(mtcars)
```

```
> df_tb
```

```
> df_tb
# A tibble: 32 x 11
     mpg   cyl  disp    hp  drat    wt  qsec    vs
   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
 1  21       6  160    110  3.9   2.62  16.5     0
 2  21       6  160    110  3.9   2.88  17.0     0
 3  22.8     4  108     93  3.85  2.32  18.6     1
 4  21.4     6  258    110  3.08  3.22  19.4     1
 5  18.7     8  360    175  3.15  3.44  17.0     0
 6  18.1     6  225    105  2.76  3.46  20.2     1
 7  14.3     8  360    245  3.21  3.57  15.8     0
 8  24.4     4  147.    62  3.69  3.19  20       1
 9  22.8     4  141.    95  3.92  3.15  22.9     1
10  19.2     6  168.   123  3.92  3.44  18.3     1
# ... with 22 more rows, and 3 more variables:
#   am <dbl>, gear <dbl>, carb <dbl>
>
```

b) Apliqueu la següent comanda new <-df_tb[, c(1,3,4,5,6,7)]. Què fa aquesta comanda?

> **new <-df_tb[, c(1,3,4,5,6,7)]**

> **new** #al veure una *tibble* veiem la informació essencial

```
> new <-df_tb[, c(1,3,4,5,6,7)]
> new
# A tibble: 32 x 6
     mpg  disp    hp  drat    wt  qsec
   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
 1  21    160    110  3.9   2.62  16.5
 2  21    160    110  3.9   2.88  17.0
 3  22.8  108     93  3.85  2.32  18.6
 4  21.4  258    110  3.08  3.22  19.4
 5  18.7  360    175  3.15  3.44  17.0
 6  18.1  225    105  2.76  3.46  20.2
 7  14.3  360    245  3.21  3.57  15.8
 8  24.4  147.    62  3.69  3.19  20
 9  22.8  141.    95  3.92  3.15  22.9
10  19.2  168.   123  3.92  3.44  18.3
# ... with 22 more rows
> 
```

Aquesta comanda està seleccionant les columnes 1, 3, 4, 5, 6 i 7 de la *tibble* inicial. Ens estem de fet quedant amb les variables quantitatives.

c) Intenteu crear una *tibble* igual que *new* a partir de *df_tb* usant eines de la llibreria dplyr que hem vist en classes anteriors

En l'apartat anterior estem agafant columnes, per tant farem servir select:

> new2 <-**select**(df_tb, 'mpg','disp','hp','drat','wt','qsec')

També es pot posar:

> new2 <-select(df_tb, **c(1,3,4,5,6,7)**)

Ens hem quedat, com volíem, amb les variables: *mpg*, *disp*, *hp*, *drat*, *wt*, *qsec.*

d) Ara que ja tenim les dades preparades, feu la matriu de correlació (que anomenarem *cormat*), arrodonint els valors a dos decimals. Per això utilitzeu les comandes que hem vist avui a la primera part de la classe (diapositives secció 6). Com és aquesta matriu?

> cormat <- **round**(**cor**(new),2)

```
> cormat
       mpg  disp    hp  drat    wt  qsec
mpg   1.00 -0.85 -0.78  0.68 -0.87  0.42
disp -0.85  1.00  0.79 -0.71  0.89 -0.43
hp   -0.78  0.79  1.00 -0.45  0.66 -0.71
drat  0.68 -0.71 -0.45  1.00 -0.71  0.09
wt   -0.87  0.89  0.66 -0.71  1.00 -0.17
qsec  0.42 -0.43 -0.71  0.09 -0.17  1.00
> 
```

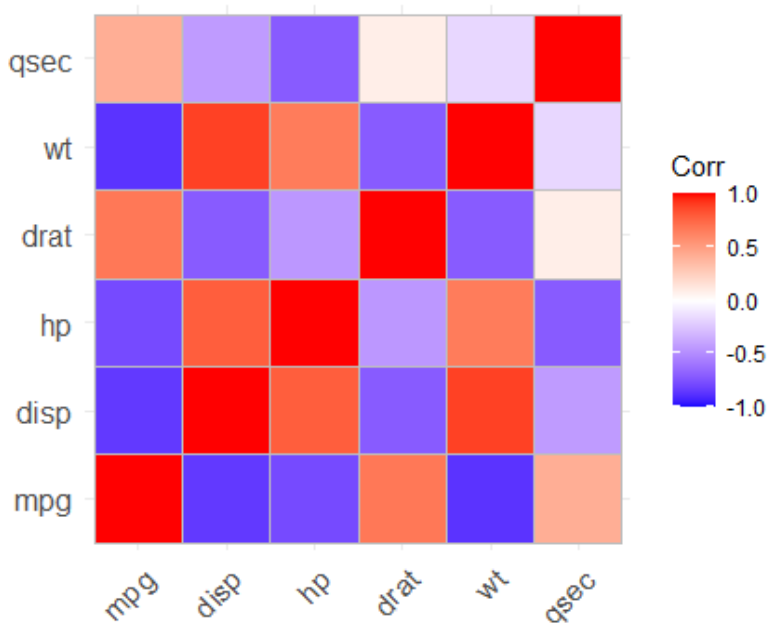
Com bona matriu de correlació, la matriu *cormat* té valors uns a la diagonal (on cada variable es relaciona amb sí mateixa), és simètrica, i els seus valors estan compresos entre -1 i 1. Recordem que com més alt és el valor absolut més correlacionades estan les variables.

e) Si no el teniu instal·lat, instal·leu el paquet que hem vist a la primera part de la classe d'avui (diapositiva 13) per treballar amb matrius de correlació (ignoreu els warnings). Després, carregueu la llibreria necessària (ignorant també el warning), i useu la funció `ggcorrplot()` per visualitzar la vostra matriu de correlació

```
> install.packages("ggcorrplot")

> library(ggcorrplot)

> ggcorrplot(cormat)
```



La funció `ggcorrplot` proporciona una solució per reordenar la matriu de correlació i mostra el nivell de significació al correlograma.

f) Afegiu els coeficients de correlació fent servir l'argument '`lab=TRUE`' de la funció `ggcorrplot()`. Si teniu dubtes feu servir l'ajuda de R (`?ggcorrplot`). Compareu la vostra visualització amb la matriu `cormat` que heu creat en l'apartat (d). Ha fet algun canvi la funció `ggcorrplot()` a l'hora de visualitzar la matriu?

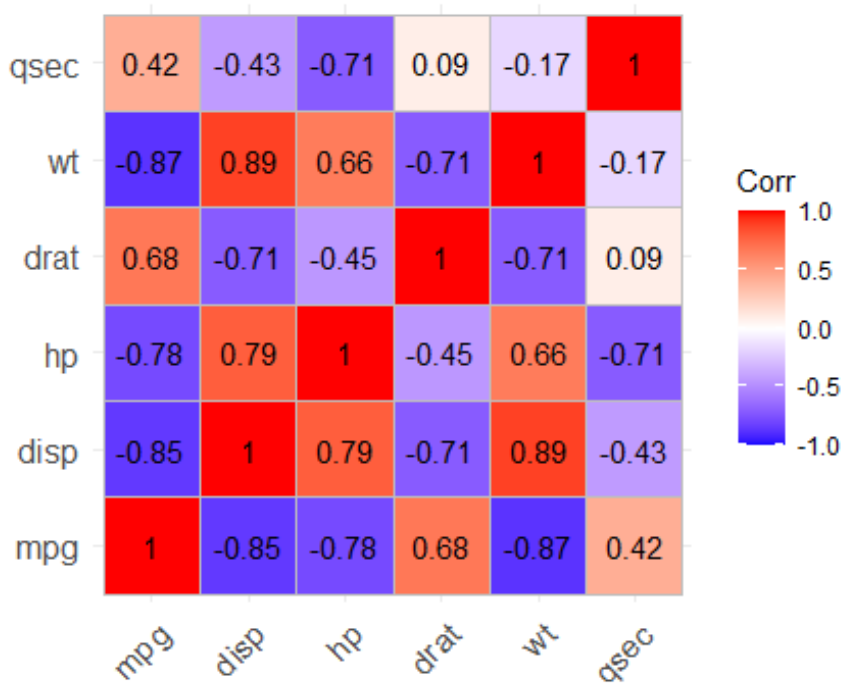
Fent ? ggcorrplot, veiem entre altres els arguments possibles de la funció

lab
logical value. If TRUE, add correlation coefficients on the plot.

R: Visualization of a correlation matrix using ggplot2 — Find in Topic

show.legend	logical, if TRUE the legend is displayed.
legend.title	a character string for the legend title. lower triangular, upper triangular or full matrix.
show.diag	logical, whether display the correlation coefficients on the principal diagonal.
colors	a vector of 3 colors for low, mid and high correlation values.
outline.color	the outline color of square or circle. Default value is "gray".
hc.order	logical value. If TRUE, correlation matrix will be hc.ordered using hclust function.
hc.method	the agglomeration method to be used in hclust (see ?hclust).
lab	logical value. If TRUE, add correlation coefficient on the plot.
lab_col, lab_size	size and color to be used for the correlation coefficient labels. used when lab = TRUE.
p.mat	matrix of p-value. If NULL, arguments sig.level, insig, pch, pch.col, pch.cex is invalid.
sig.level	significant level, if the p-value in p-mat is bigger than sig.level, then the corresponding correlation coefficient is regarded as insignificant.
insig	character, specialized insignificant correlation coefficients, "pch" (default), "blank". If "blank", wipe away the

> ggcorrplot(cormat, **lab=TRUE**)



Fixeu-vos que en la visualització, les files de la matriu a la visualització ens surten (per defecte) '*ordenades a l'inversa de com tenim realment la matriu*' a l'apartat (d):

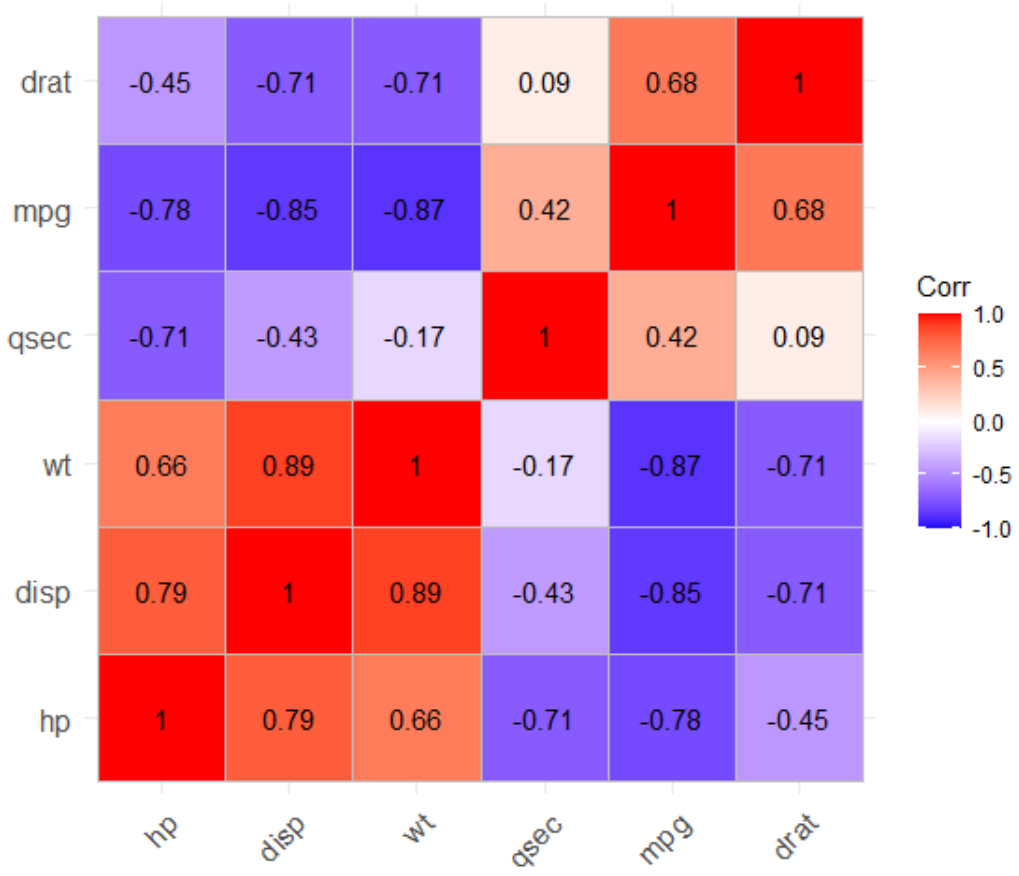
```
> cormat
        mpg  disp    hp  drat    wt  qsec
mpg    1.00 -0.85 -0.78  0.68 -0.87  0.42
disp  -0.85  1.00  0.79 -0.71  0.89 -0.43
hp    -0.78  0.79  1.00 -0.45  0.66 -0.71
drat   0.68 -0.71 -0.45  1.00 -0.71  0.09
wt    -0.87  0.89  0.66 -0.71  1.00 -0.17
qsec   0.42 -0.43 -0.71  0.09 -0.17  1.00
>
```

Per tant els 1's estan ara a l'anti-diagonal de la matriu, on cada variable es relaciona amb si mateixa.

g) Ara afegiu l'argument 'hc.order=TRUE', què passa?

Si afegim l'argument 'hc.order=TRUE':

> ggcorrplot(cormat, lab=TRUE, **hc.order = TRUE**)



Aquest argument fa ús de la funció `hclust()` de R. `hclust()` és una funció d'anàlisi jeràrquica de clústers sobre un conjunt de diferències i mètodes per analitzar-lo.

Al posar 'hc.order=TRUE', se'ns han reordenat a l'eix *x* i *y* les diferents variables per clústers.

h) Per defecte la funció `ggcorrplot()` utilitza quadrats per a visualitzar la matriu de correlació. Feu ús de l'ajuda de R i mostreu cercles enlloc de quadrats. No hi poseu els coeficients de correlació aquest cop (és a dir, no activeu l'argument 'lab=TRUE').

Al fer `?ggcorrplot`:

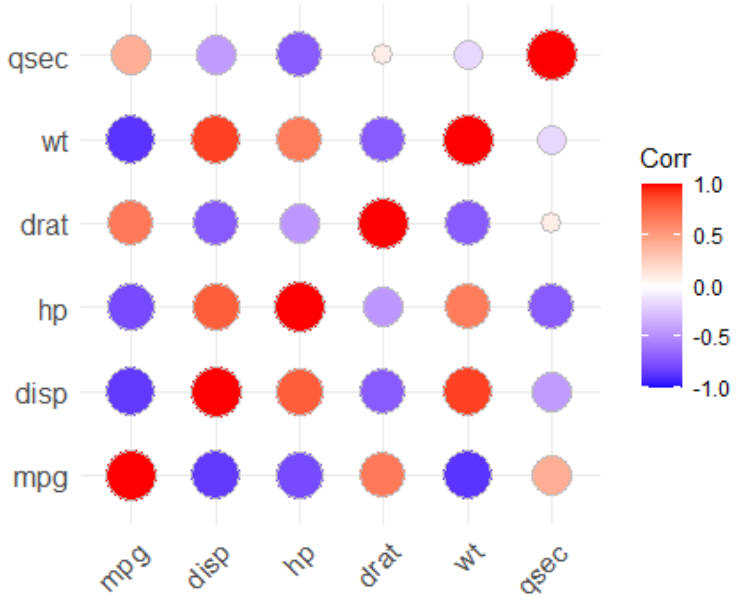
method
character, the visualization method of correlation matrix to be used. Allowed values are "square" (default), "circle".

`cor_pmat(x, ...)`

Arguments

`corr`	the correlation matrix to visualize
`method`	character, the visualization method of correlation matrix to be used. Allowed values are "square" (default), "circle".
`type`	character, "full" (default), "lower" or "upper" display.
`ggtheme`	ggplot2 function or theme object. Default value is 'theme_minimal'. Allowed values are the official ggplot2 themes including theme_gray, theme_bw, theme_minimal, theme_classic, theme_void, ... Theme objects are also allowed (e.g., 'theme_classic()').
`title`	character, title of the graph.
`show.legend`	logical, if TRUE the legend is displayed.
`legend.title`	a character string for the legend title. lower triangular, upper triangular or full matrix.
`show.diag`	logical, whether display the correlation coefficients on the principal diagonal.
`colors`	a vector of 3 colors for low, mid and high correlation values.
`outline.color`	the outline color of square or circle. Default value is "gray".
`hc.order`	logical value. If TRUE, correlation matrix will be hc.ordered using hclust function.

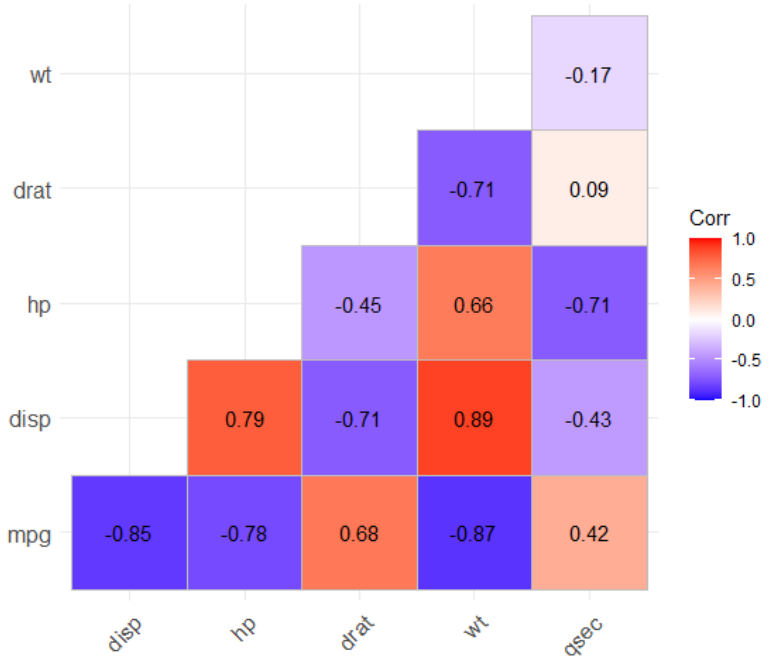
> ggcorrplot(cormat, **method='circle'**)



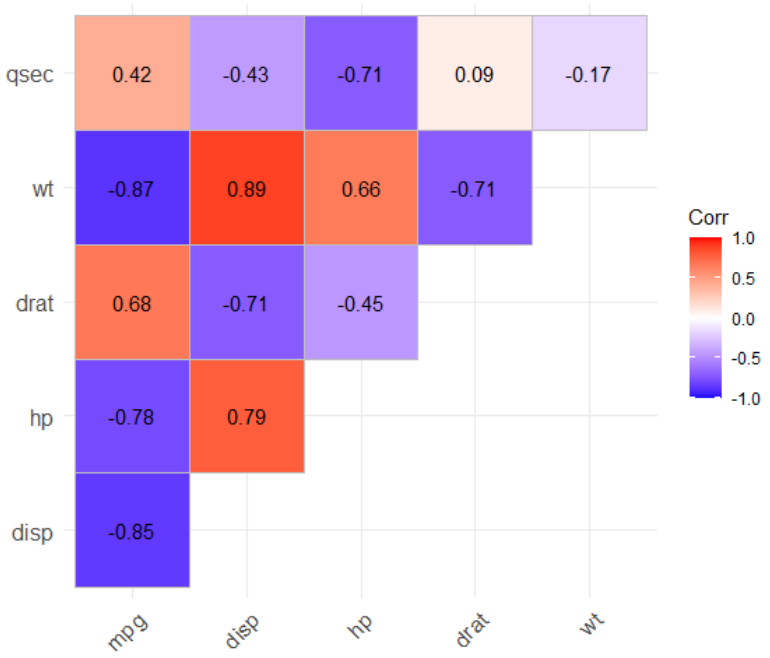
Podem veure que en aquest cas a més els cercles són proporcionals al valor absolut de correlació. Quan més a prop de 1, més grans, quan més a prop de 0, més petits. Per tant, en el cas dels cercles, no ens cal tenir els valors per tenir una intuïció de com de correlacionades estan les nostres variables.

i) Donada la simetria de la matriu i que l'anti-diagonal està formada per 1's, podem mostrar just la part de sobre ("upper") o de sota ("lower") de l'anti-diagonal. Això ens permetrà reduir la "carrega de visualització de la informació" (i serà d'agrair per la gent que vegi la nostra gràfica). Indirectament centrarem la seva atenció en el realment important. Per això podem jugar amb l'argument type de la funció `ggcorrplot()`. Feu altre cop `?ggcorrplot` i introduïu aquest argument a la visualització de l'apartat (f).

> ggcorrplot(cormat, lab=TRUE, **type = "lower"**)



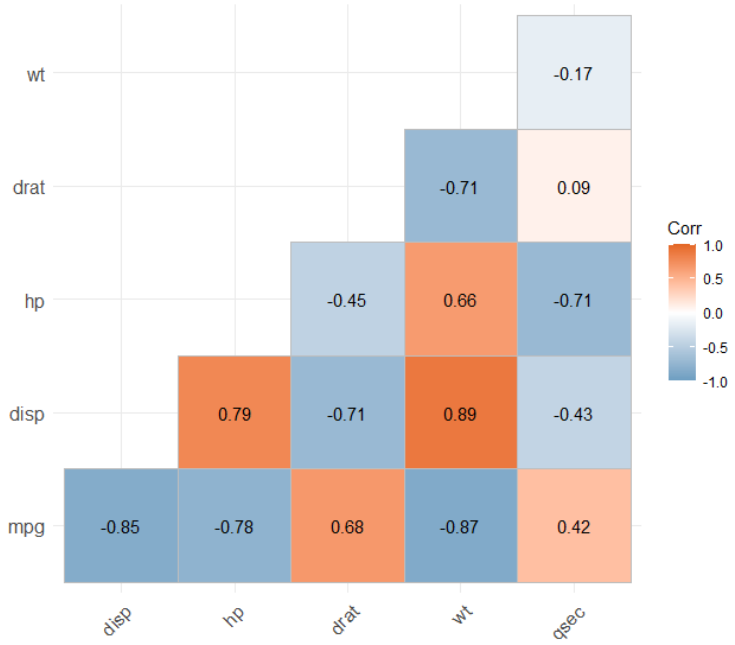
> ggcorrplot(cormat, lab=TRUE, **type = "upper"**)



NOTA: (a) i (b) no són necessaris, podríem haver començat directament en (c), però ho hem fet per practicar amb l'ús de *tibbles*.

Podeu canviar els colors, per exemple per uns amb una tonalitat més suau. Per això fem us de l'argument color:

> ggcorrplot(cormat, lab=TRUE, type="lower", **colors = c("#6D9EC1", "white", "#E46726")**)



Material extra: Si voleu aventurar-vos a canviar colors, podeu fer us de la cheatsheet https://www.nceas.ucsb.edu/sites/default/files/2020-04/colorPaletteCheatsheet.pdf

4. PART 3. Scatter Plot Matrix (SPLOM)

Per aquesta tercera part, treballarem amb el dataframe iris, amb el que ja hem treballat en ocasions anteriors i veurem com es crea un SPLOM.

Iris proporciona les mesures (en cm) de les variables longitud i amplada dels sèpals i dels pètals respectivament per 50 flors de cadascuna de les 3 espècies d'Iris (150 en total). Les espècies d'iris són: la Versicolor, la Virginica i la Setosa.

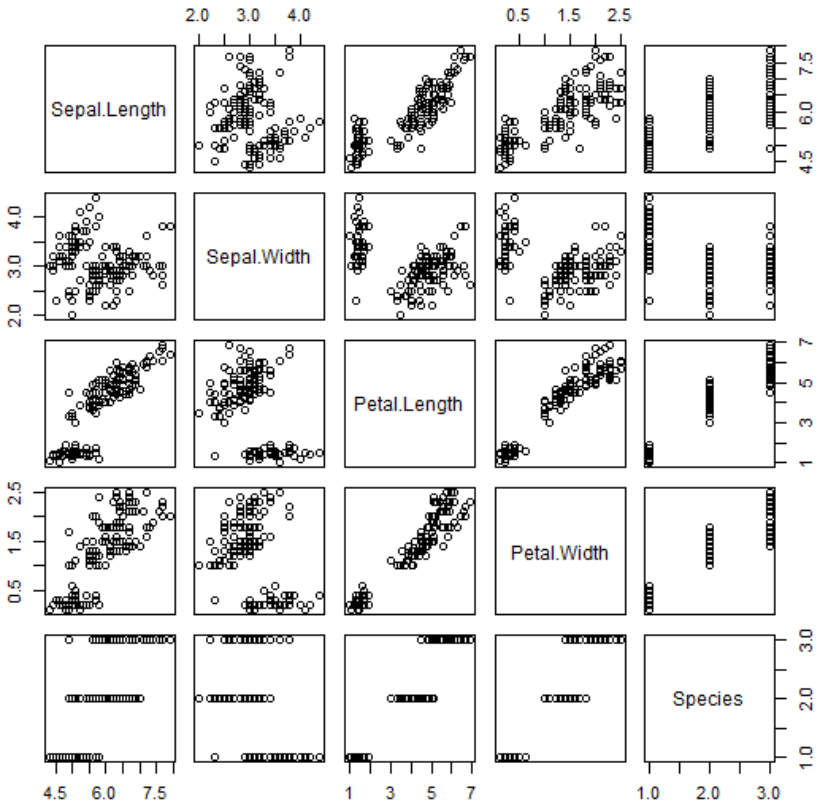
Iris és un dataframe amb 150 observacions (files) i 5 variables (columnes): **Sepal.Lenght**, **Sepal.Width**, **Petal.Length**, **Petal.Width** i **Species**. Els valors de les variables referents a les respectives longituds i amplades (mètriques) estan en centímetres.

EXERCICIS:

1.- Partint del dataframe iris

a) Feu us de la funció `pairs` de R per fer una visualització SPLOM.

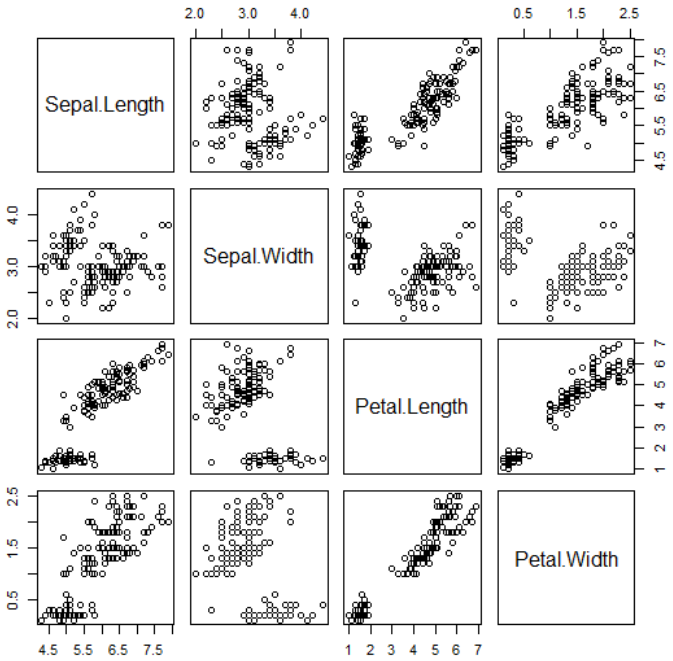
```
> pairs (iris)
```



b) Veient el gràfic de l'apartat (a), eliminaríeu alguna variable? Per què? Si és el cas traieu-la i del SPLOM

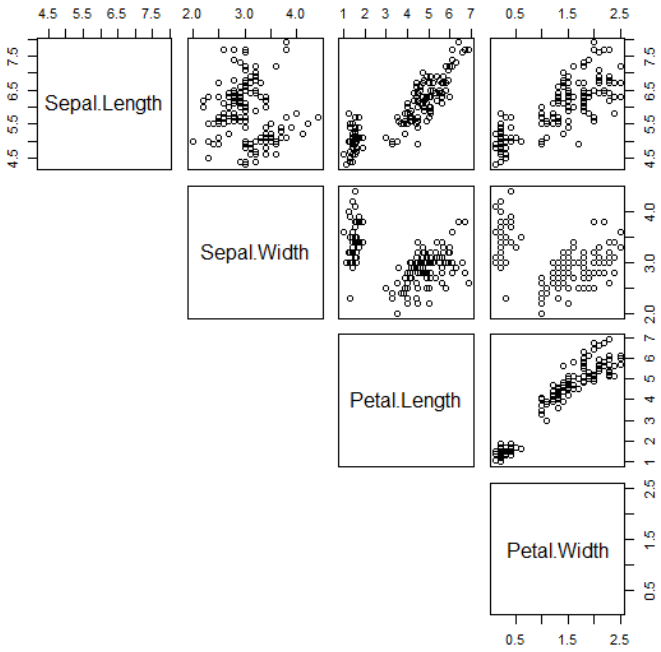
Veiem clarament que fer el scatter plot entre la variable Species (variable categòrica) i la resta de variables de mètriques no ens dóna massa informació. La traiem:

```
> pairs (iris [1:4])
```



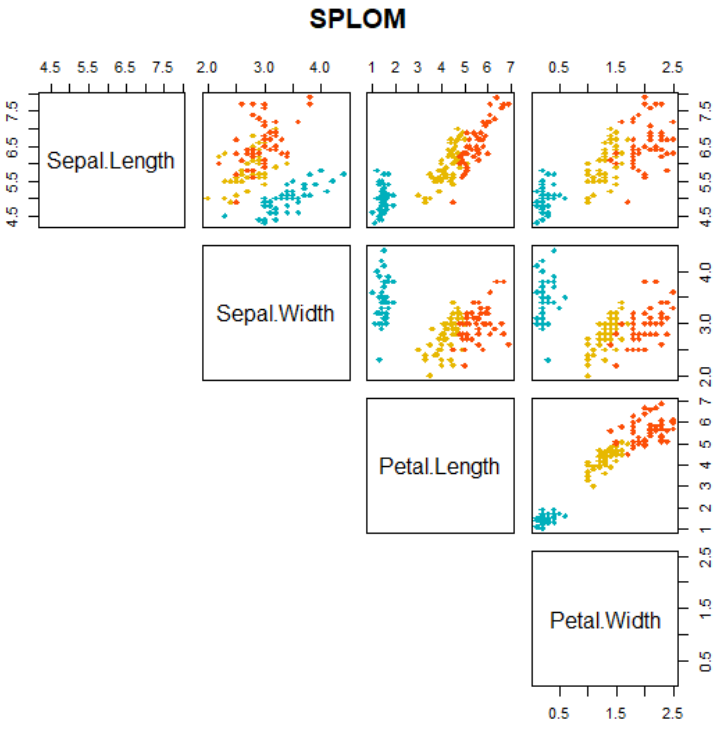
c) Com és una matriu simètrica, elimineu la diagonal inferior del gràfic fent ús de : `lower.panel = NULL`

```
> pairs(iris[1:4], lower.panel = NULL)
```



c) Posa color per especies. Per tal d'afegir aquesta informació. Pots fer ús de `col=c("#00AFBB", "#E7B800", "#FC4E07")[iris$Species]`, o triar altres colors. Pots a més omplir el punt de color, afegint `pch=#` (on # és un número, per exemple , 18 o 19 són bones opcions). Afegeix títol també amb `main="SPLOM"`.

```
> pairs(iris[,1:4],lower.panel=NULL, col = c("#00AFBB", "#E7B800", "#FC4E07")[iris$Species], pch=18, main="SPLOM" )
```



2.- Ara anem a fer un scatter plot matrix (SPLOM) fent us de la funció ggpairs() del paquet GGally de ggplot2. Per això:

a) Com sempre primer instal·leu el paquet necessari, en aquest cas GGally i carregueu la llibreria (necessiteu també la llibreria ggplot2).

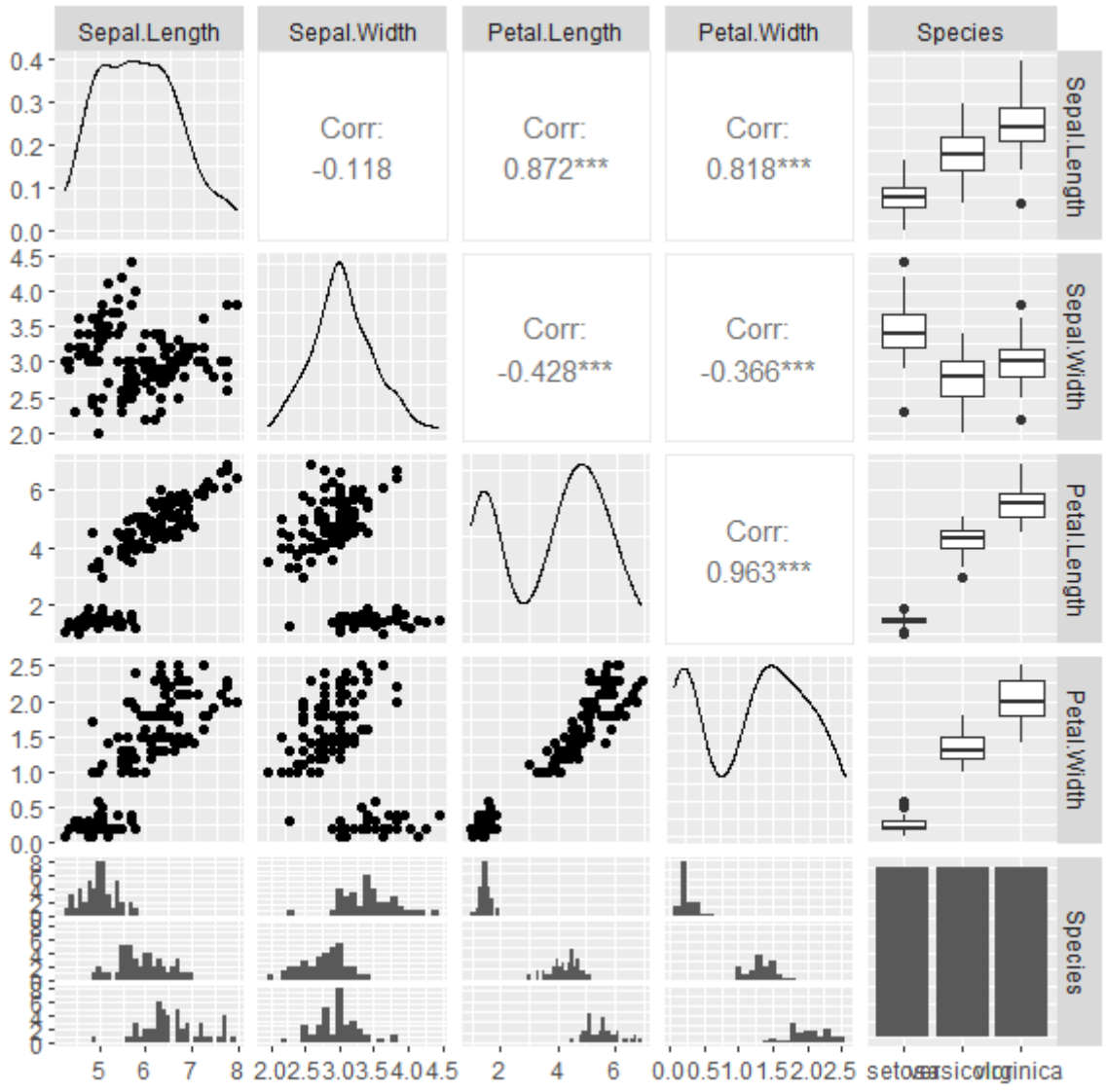
```
> install.packages("GGally")
> library(ggplot2)
> library (GGally)
```

b) Un cop tingueu la llibreria carregada, feu `ggpairs(iris)`. Què observeu en la part esquerra, diagonal i part dreta del gràfic, respectivament?

Els diagrames de dispersió (scatterplot) de cada parell de variables numèriques es dibuixen a la part esquerra de la figura. La correlació de Pearson es mostra a la dreta. La distribució variable està disponible a la diagonal.

b.1) Si us ha sortit un warning referent al binwidth useu:

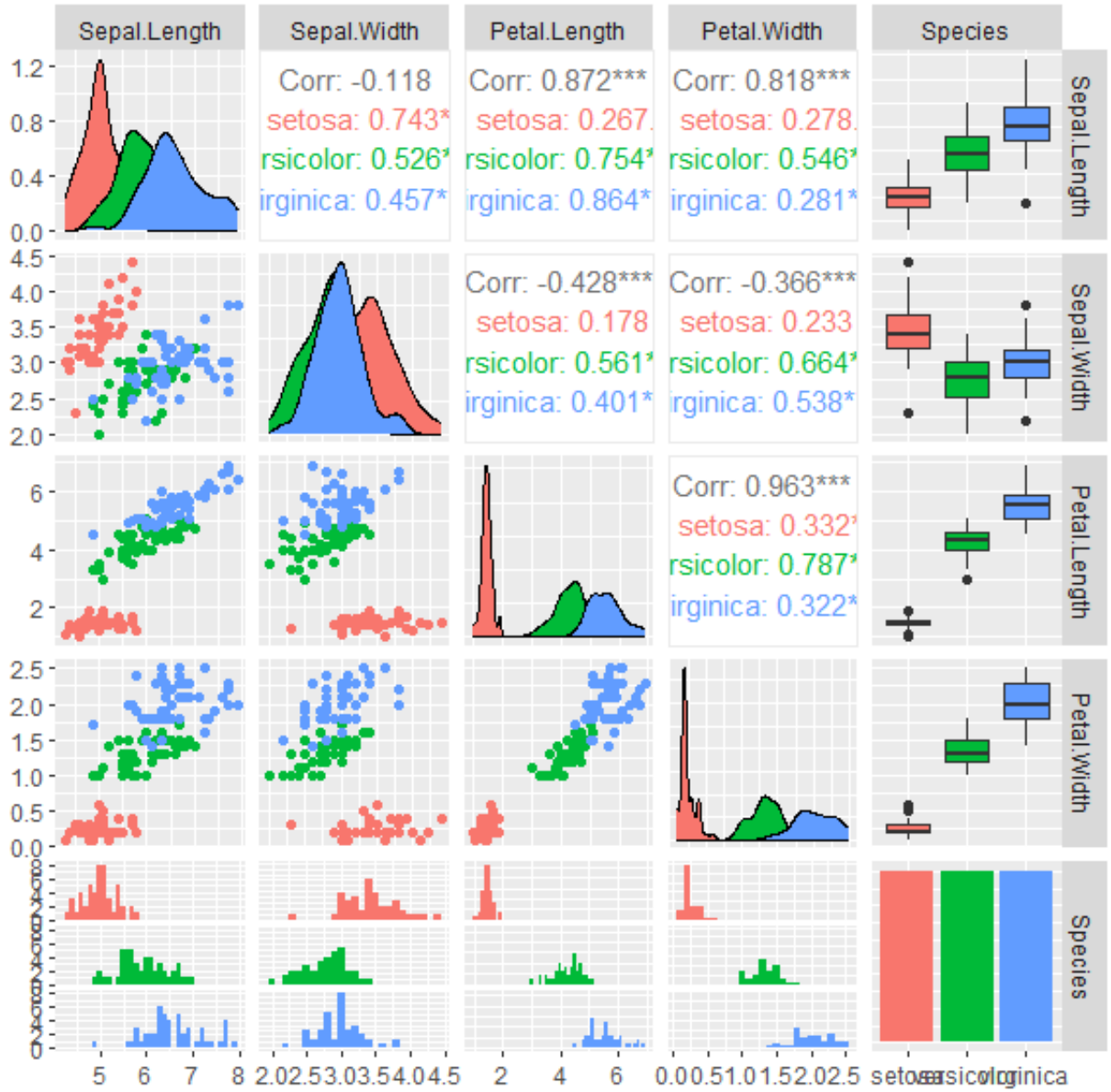
```
>ggpairs(iris, lower=list(combo=wrap("facethist", binwidth=0.1)))
```



c) Fent servir l'estètica (`aes()`), completeu les * de la següent comanda per tal de pintar el gràfic obtingut en l'apartat (b) segons l'espècie (*Species*). Pista: Com ha de ser la variable *Species* per poder usar l'estètica del color?**

```
> ggpairs(iris, aes(color=***),..)
```

```
> ggpairs (iris,aes(color=as.factor(Species)),lower=list(combo=wrap("facet
hist", binwidth=0.1)))
```



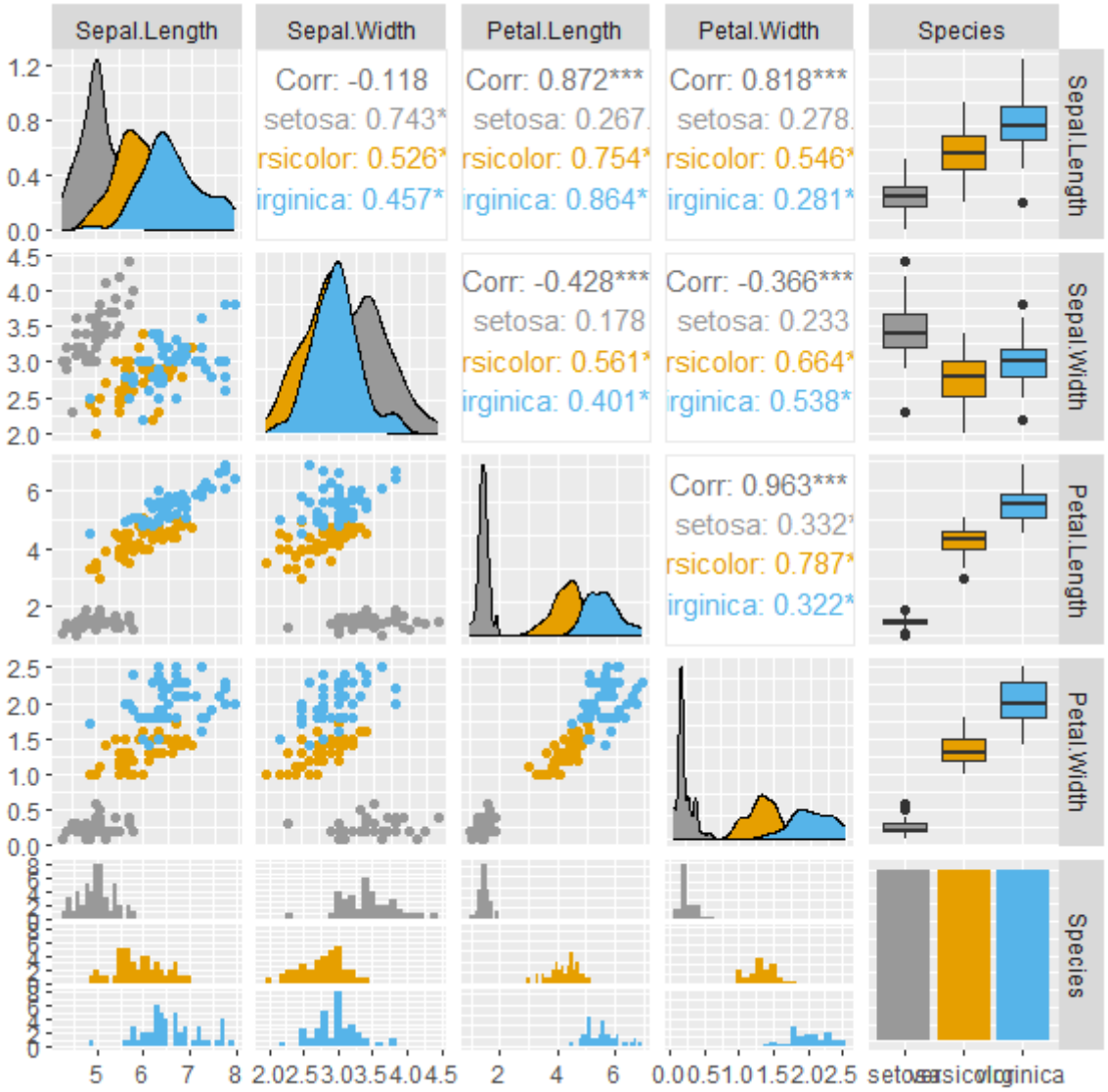
d) Poseu un altre color, per exemple manualment posant el color donat en: `c("#999999", "#E69F00", "#56B4E9")`

Per facilitar, assignem el gràfic a una variable

```
>gràfic<-ggpairs
(iris,aes(color=as.factor(Species)),lower=list(combo=wrap("facethist",
binwidth=0.1)))
```

Fixem-nos que alguns gràfics requereixen color i altres fill, per tant, haurem d'usar:

```
>grafic+scale_color_manual(values=c("#999999", "#E69F00", "#56B4E9"))+scale_fill_manual(values=c("#999999", "#E69F00", "#56B4E9"))
```



Podem millorar-ho usant el canal `theme.` Però com vam veure a classe de teoria, no usarem aquests gràfics amb moltes variables (4 com a màxim potser). Ara, ens poden ser de gran ajuda com a gràfiques exploratòries.

e) Ara anem a categoritzar la longitud del pètal en dos subgrups:

- **Direm que pertany al grup "llarg", si la longitud del pètal (`Petal.Length`) està per sobre de la longitud mitjana del pètal.**
- **Anàlogament, direm que és "curt", si la longitud del pètal està per sota de la longitud mitjana del pètal.**

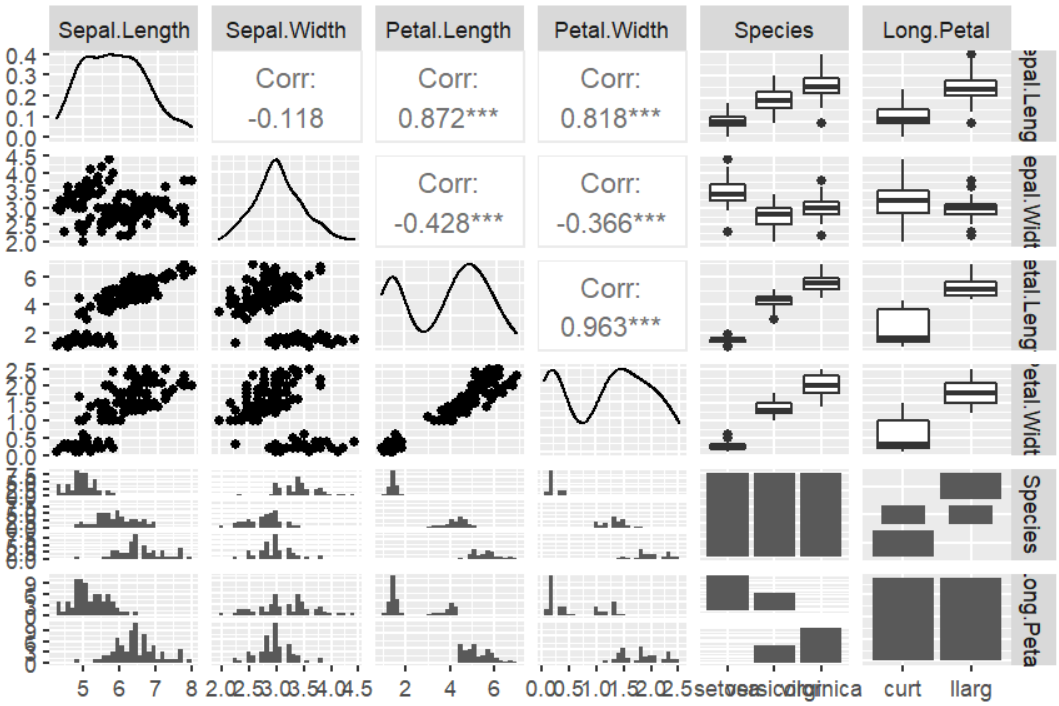
Tot i que us dono la comanda per fer aquest apartat (e), ja que conté part d'estadística que no hem vist en aquest curs, fixeu-vos com ho fem:

```
iris$Long.Petal<-
as.factor(ifelse(iris$Petal.Length>median(iris$Petal.Length), "llarg",
"curt"))
```

En R, les dades poden ser forçades per transformar-les a un altre tipus. Aquí hem fet una coerció explícita per categoritzar, usant la funció `as.factor`. Indirectament, ja vam veure això amb la llibreria `mtcars` i la variable cilindres en el mòdul 3.

(Podeu trobar més informació sobre coerció explícita en la secció 4.7.1 d'aquest document: https://bookdown.org/jboscomendoza/r-principiantes4/coercion.html)

f) Feu us de ggpairs per mostrar el dataframe *iris* modificat en (d)



En aquesta nova visualització que us sortirà podreu observar:

- **Uns gràfics de densitat i uns gràfics de barres a la diagonal que reflecteixen les distribucions marginals de les variables. Veureu, que per a les trames quantitatives-quantitatives, hi ha una forta associació positiva entre la longitud i l'amplada dels pètals, que també es recolza en una correlació de 0,872.**
- **També observareu un panell de mosaic entre les espècies d'iris i els grups de longitud de pètals mostrant ambdues distribucions condicionals; per exemple, en el panell de la fila 5, columna 6, us ha d'aparèixer la distribució dels grups de longitud de pètals per espècies.**

Podeu veure més exemples amb `ggpairs()` usant el dataframe `iris` en: https://r-charts.com/es/correlacion/ggpairs/.

Judit Chamorro Servent
Bellaterra, Abril 2025