

Fiche TD avec le logiciel  : tdr14a

Graphiques (données économiques)

D. Chessel, A.B. Dufour & J.R. Lobry

Illustrations des fonctions graphiques à partir d'une chronique multivariée.

Table des matières

1 Chroniques	2
1.1 Graphes d'une chronique	2
1.2 Chronique multivariée	5
2 Données et modèle additif	6
3 Données et modèle multiplicatif	8
4 Quelques paramètres graphiques	10

1 Chroniques

Récupérer le fichier :

`http://pbil.univ-lyon1.fr/R/donnees/bourse.txt`

Il contient les moyennes mensuelles de 1991 à 1999 de sept indices boursiers ce qui donne 108 lignes et 7 colonnes. On a ajouté un code mois (12 modalités) et l'année dans la variable an.

On peut importer directement dans le dossier de travail par :

```
download.file("http://pbil.univ-lyon1.fr/R/donnees/bourse.txt",
             "bourse.txt")
```

Assurez-vous du bon chargement du fichier en affichant son contenu dans une fenêtre de R :

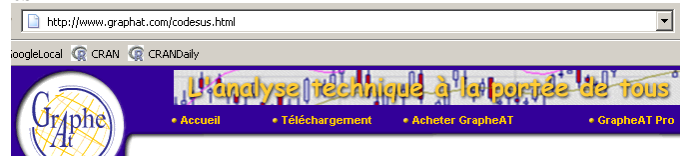
```
file.show("bourse.txt")
```

R Information										
1894.5	1533.1	2841.8	3308.4	435.7	675.31	16980	Ja	1993		
1766.3	1532.4	2821.9	3317.9	438.78	697.58	17038	Fe	1993		
2005.2	1695.8	2861.9	3374.9	443.38	670.88	16956	Ma	1993		
2023.2	1677.6	2871.3	3440.8	451.67	690.82	18596	Av	1993		
1925.3	1620.3	2820.3	3428.6	440.19	662.75	20939	Ma	1993		
1870.8	1624.9	2836.1	3543.2	450.23	700.96	20532	Ju	1993		
1982.7	1705.5	2901.5	3505	450.54	702.01	19639	Jl	1993		
2131.2	1795.4	2920	3550.9	448.13	706.38	20393	Ao	1993		
2205.1	1938.4	3099.5	3644.5	463.55	743.65	20929	Se	1993		
2114.4	1912.9	3039	3567.7	458.93	761.62	20097	Oc	1993		
2174.3	2064.7	3170.7	3680.3	467.83	780.37	19657	No	1993		
2118.6	2072.7	3197	3699.9	461.93	759.9	16450	De	1993		
2263	2241.7	3427.2	3753.8	466.51	774.11	17422	Ja	1994		
2339.1	2183.5	3511.4	3965.4	481.6	798.42	20256	Fe	1994		
2241.7	2090.5	3322.9	3810.9	467.19	792.1	20053	Ma	1994		
2067.8	2134.1	3060.9	3587.9	445.66	723.06	19162	Av	1994		
2178.1	2261	3131.9	3673.4	450.91	733.97	19649	Ma	1994		

Importer le contenu de ce fichier dans un data.frame :

```
bourse <- read.table("bourse.txt", h = T)
```

Editer quelques lignes du tableau et vérifier que tout est correct. Identifier les variables.



1.1 Graphes d'une chronique

Étudier diverses représentations de `bourse$CAC40` (figure 1) :

```
plot(bourse$CAC40)
plot(bourse$CAC40, type = "h")
plot(bourse$CAC40, type = "l")
plot(bourse$CAC40, type = "b")
plot(bourse$CAC40, type = "n")
plot(bourse$CAC40, type = "s")
```

Les `barplot`, `boxplot` et les `dotchart` sont des graphes fondamentaux (figure 2).

```
barplot(bourse$CAC40, ylim = c(1000, 5000))
dotchart(bourse$CAC40, gr = as.factor(bourse$an))
plot(bourse$CAC40 ~ bourse$an)
boxplot(split(bourse$CAC40, bourse$an))
```

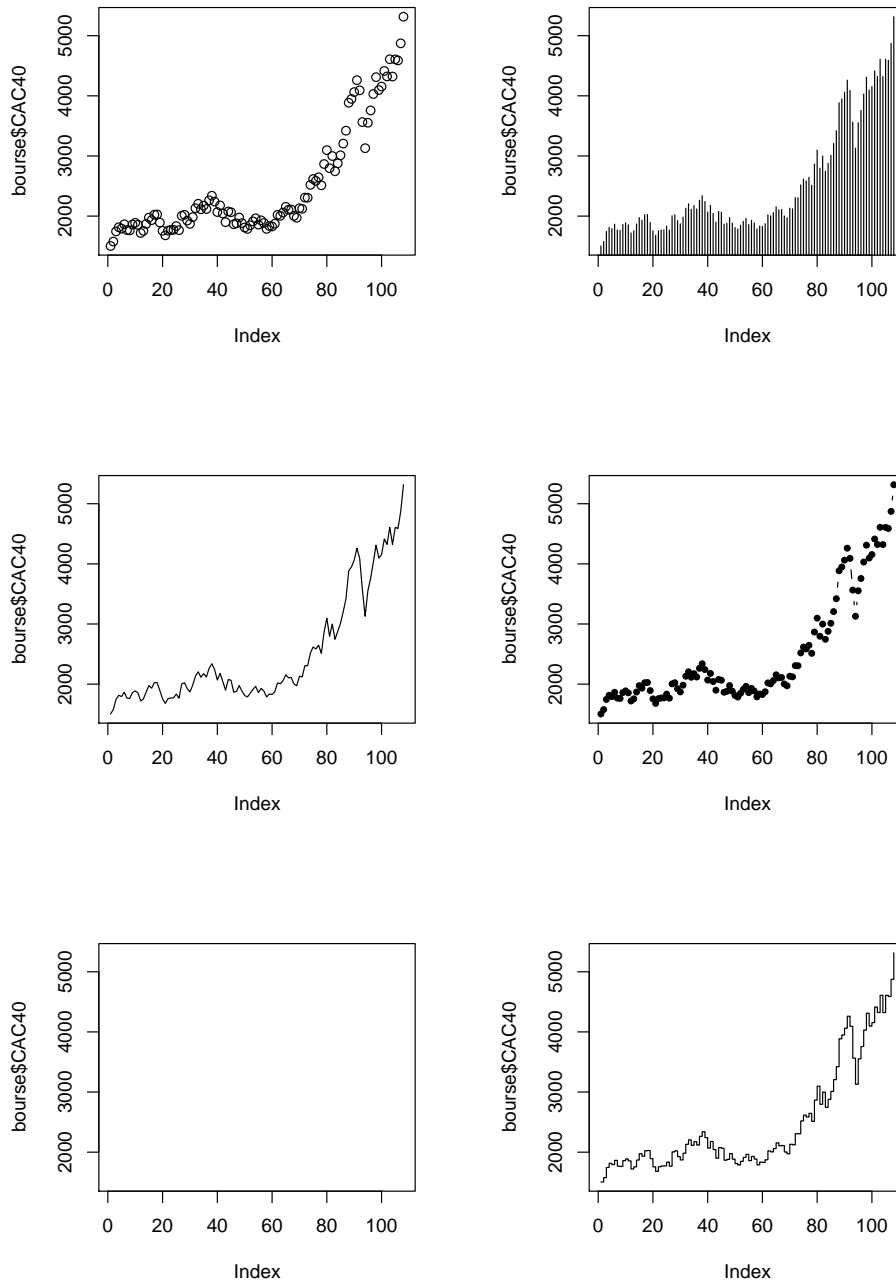


FIG. 1 – Types de points dans un plot.

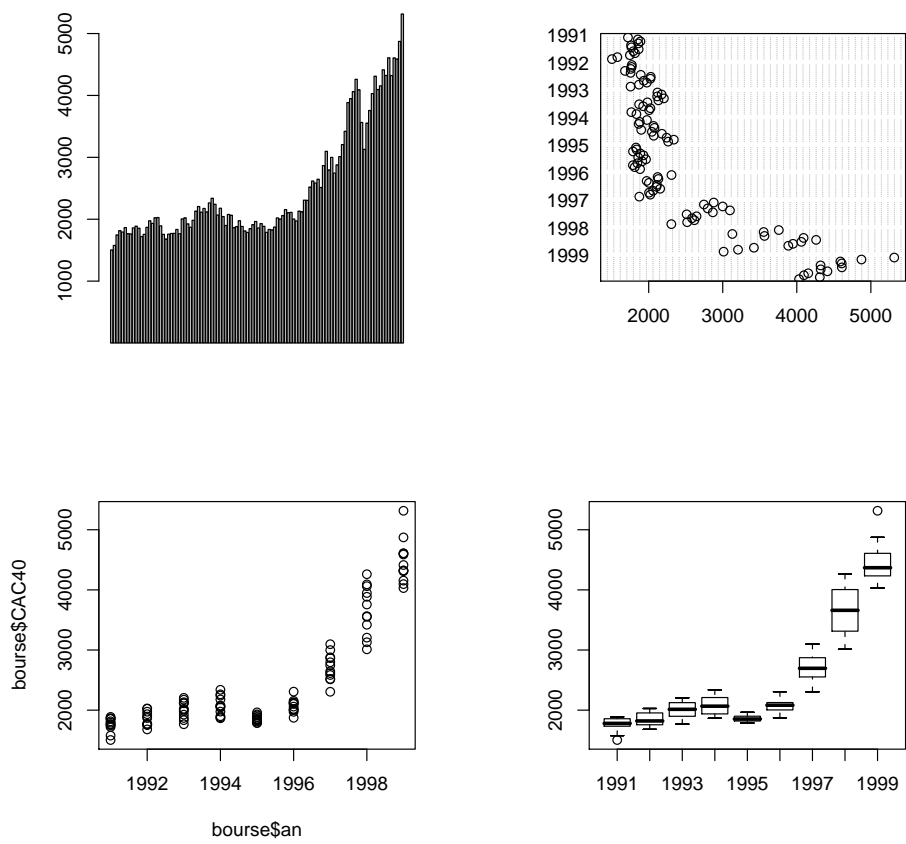
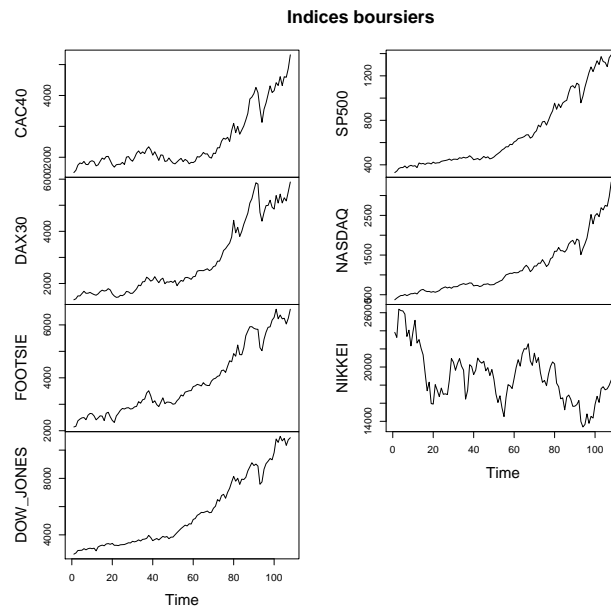


FIG. 2 – Types de plot.

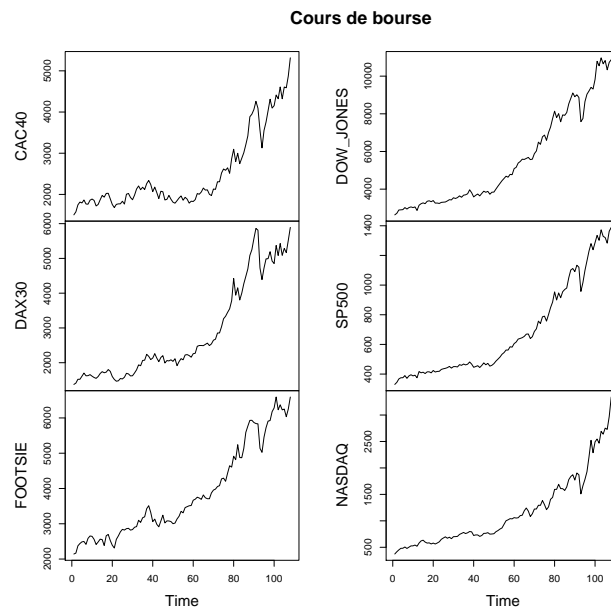
1.2 Chronique multivariée

```
plot(as.ts(as.matrix(bourse[, 1:7])), main = "Indices boursiers")
```



Il y a un *outlier* !

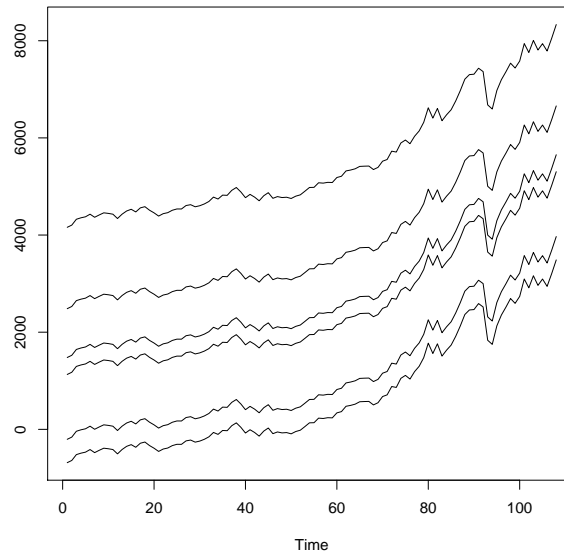
```
plot(as.ts(as.matrix(bourse[, 1:6])), main = "Cours de bourse")
```



On devrait pouvoir mettre en place un modèle commun. Que représente la figure obtenue par l'ordre suivant ?

```
plot(bourse[, 1:6])
```

2 Données et modèle additif

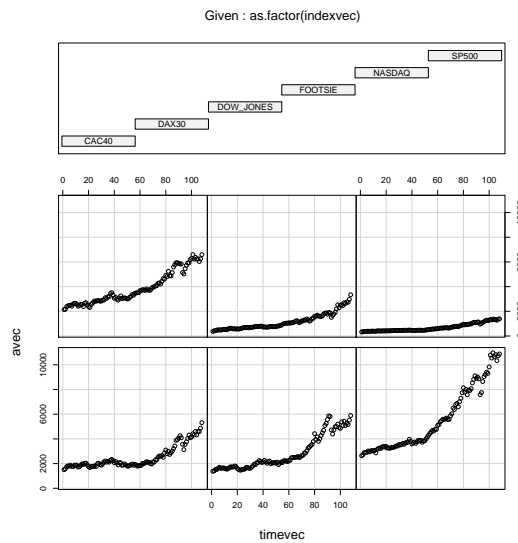


```
a <- as.matrix(bourse[1:6])
a <- as.matrix(bourse[1:6])
m1 <- apply(a, 1, mean)
mc <- apply(a, 2, mean)
mg <- mean(a)
amod <- matrix(m1[row(a)] + mc[col(a)] - mg, ncol = 6)
ts.plot(as.ts(amod))
```

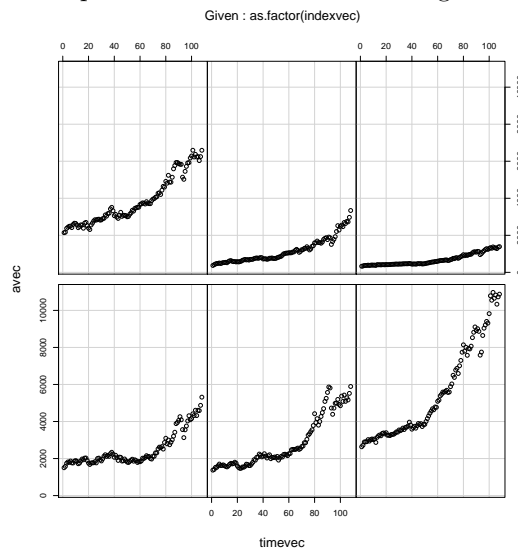
Identifier et justifier le calcul effectué.

On veut représenter les données et les modèles. La fonction `coplot` est fort utile.

```
avec <- as.vector(a)
amodvec <- as.vector(amod)
timevec <- rep(1:108, 6)
indexvec <- as.vector(names(bourse[, 1:6])[col(a)])
coplot(avec ~ timevec | as.factor(indexvec))
```

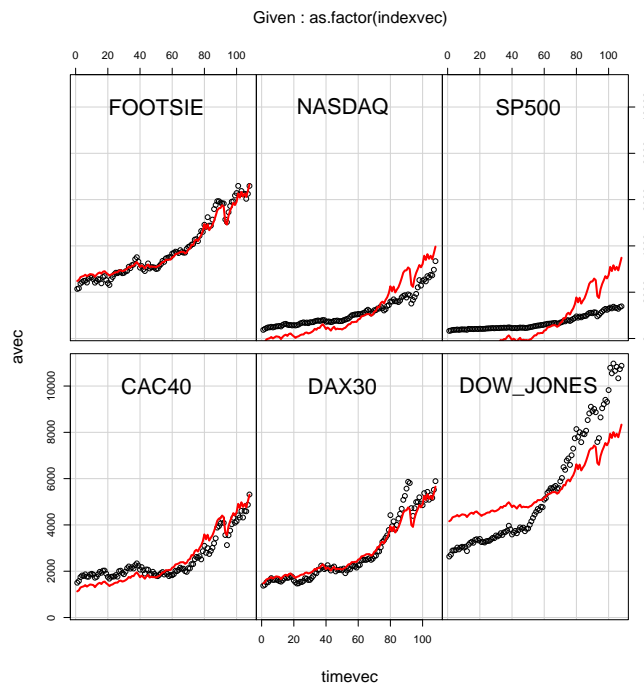


Quel est le paramètre qui contrôle les indications marginales ?



Des concepts fondamentaux : la fonction `panel` et l'indice `subscripts` :

```
pan1 <- function(x, y, subscripts, col, pch) {
  points(x, y)
  lines(x, amodvec[subscripts], lwd = 2, col = "red")
  text(50, 10000, unique(indexvec[subscripts]), cex = 2)
}
coplot(avec ~ timevec | as.factor(indexvec), pan = pan1, show = F,
       sub = T)
```



Un très mauvais modèle. Cherchons un modèle multiplicatif ¹.

3 Données et modèle multiplicatif

Pour faire une décomposition en valeurs singulières :

```
svd0 <- svd(a)
```

Faire le **summary** de l'objet `svd0`. Identifier ses composantes en se reportant à la documentation. Éditer les valeurs singulières de la matrice `a`.

Par définition les valeurs singulières de la matrice `X` sont les racines carrées des valeurs propres de la matrice $X^T X$. Vérifier d'abord que les matrices du type $X^T X$ sont carrées, symétriques, diagonalisables, à valeurs propres positives.

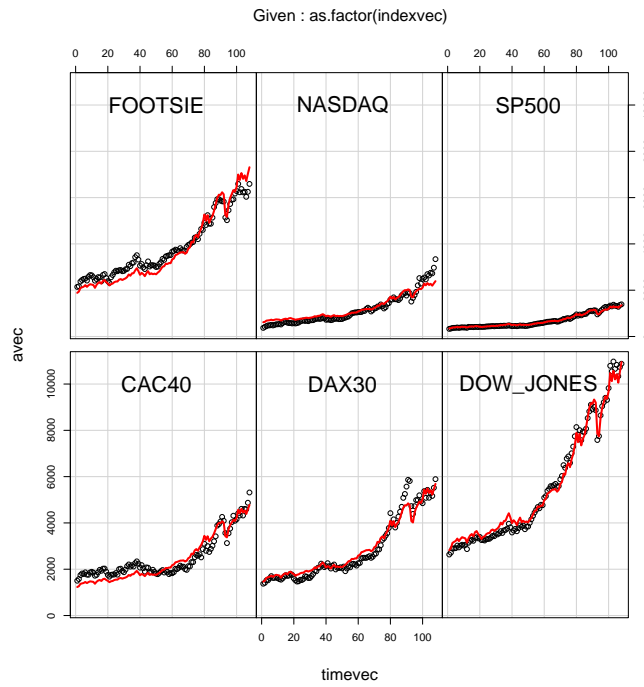
Calculer le produit de la matrice transposée de `a` par la matrice `a`, vérifier ses dimensions, éditer ses vecteurs propres et ses valeurs propres. Vérifier que ces dernières sont les carrés des valeurs singulières. On a besoin de savoir que :

- * la fonction qui diagonalise les matrices est `eigen`;
- * la fonction qui transpose une matrice est simplement `t`;
- * le produit de matrices se fait par `%*%`.

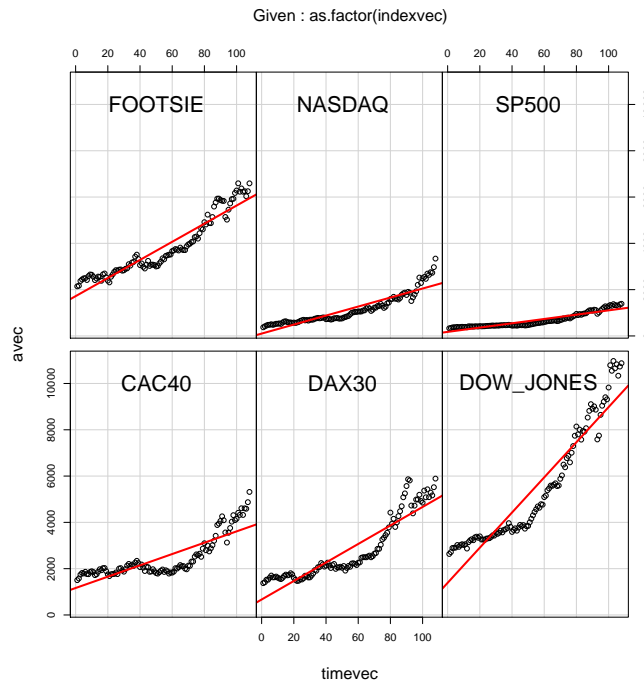
Calculer alors la matrice de rang 1 la plus proche de la matrice des données et représenter le modèle obtenu.

```
amod <- svd0$d[1] * svd0$u[, 1] %*% t(svd0$v[, 1])
dim(amod)
amodvec <- as.vector(amod)
coplot(avec ~ timevec | as.factor(indexvec), pan = pan1, show = F,
       sub = T)
```

¹Le justificatif théorique de cet exercice est plus difficile. Sauter en première lecture.



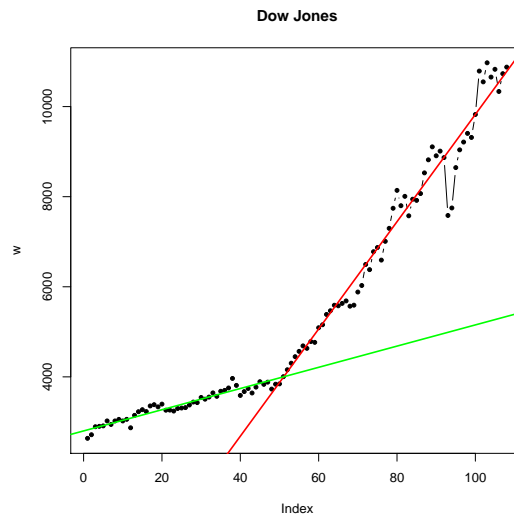
Donner une légende.



Retrouver le graphique ci-dessus qui contient une droite de régression par chronique. On utilisera la fonction `lm`. Ici c'est très mauvais, mais l'exercice est utile.

Question : plot, coplot et dotchart sont-elles des fonctions génériques ?

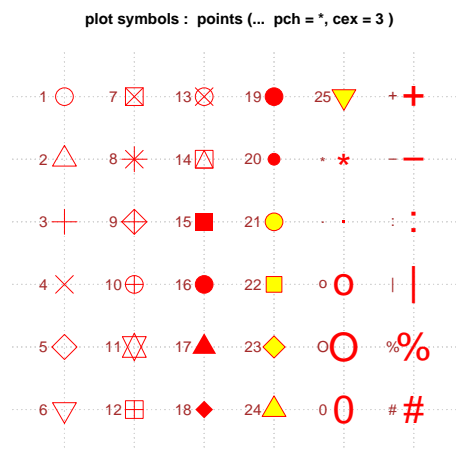
Question : refaire et commenter cette figure.



4 Quelques paramètres graphiques

La richesse des possibilités d'expression graphique est considérable dans . Une multitude de paramètres sont accessibles à l'utilisateur mais la plupart du temps, il suffit au débutant d'utiliser les valeurs par défaut et d'ignorer même l'existence de ces paramètres. On cherchera ensuite à enrichir sa palette avec `par(graphics)`.

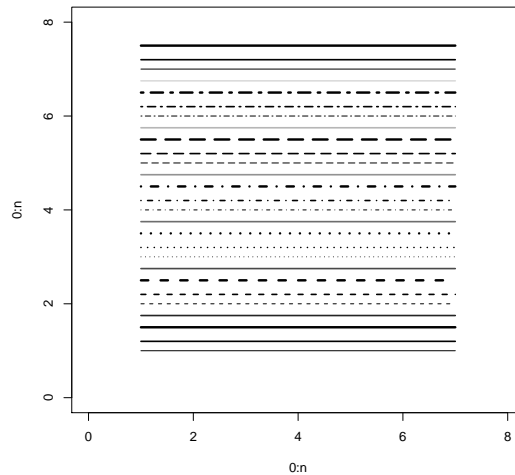
Pour commencer, utiliser la fiche de documentation de la fonction `points` pour obtenir quelques types de points :



Pour avoir quelques types de lignes :

```
n <- 7
plot(0:n, 0:n, type = "n", xlim = c(0, n + 1), ylim = c(0, n + 1))
for (i in 1:n) lines(1:n, rep(i, n), lwd = 1, lty = i)
for (i in 1:n) lines(1:n, rep(i + 0.2, n), lwd = 2, lty = i)
```

```
for (i in 1:n) lines(1:n, rep(i + 0.5, n), lwd = 3, lty = i)
for (i in 1:n) lines(1:n, rep(i + 0.75, n), lwd = 2, col = grey(i/n))
```



Pour un exemple de paramétrage plus évolué, prenons une question posée sur le forum *Adelist* (<http://pbil.univ-lyon1.fr/ADE-4/adelist.html>) par Stéphane Lamasse :

Comment est-il possible avec `plot.ts` ou `plot` d'afficher sur de longues séries chronologique, toutes les années ou seulement une ou deux ou celles qui nous semblent importantes. Si je dois, par exemple, réaliser une courbe de l'évolution de la mortalité entre 1940 et 2005 en France, comment avoir chaque année ?

Les échelles indiquées sur les axes sont entièrement accessibles à l'utilisateur dans R. Le défaut est évidemment le plus utilisé. Pour suivre l'exemple proposé, prenons le nombre de naissances en France entre 1940 et 2004 (<http://www.insee.fr/fr/ffc/figure/NATTEF02218.XLS>). On a donc 65 valeurs. En standard (figure 3-A), les axes sont légendés par des valeurs calculées par la fonction `pretty`.

On peut changer ceci à sa guise en faisant un plot sans axes, mais seul le tracé manque, tout est calculé quand même (figure 3-B). Les axes sont à tracer soi même par la fonction `axis`. On peut le faire pour l'axe des y avec `slide=2` (figure 3-C).

Pour l'axe des x `slide=1`, la variation est de 1 à 65, mais doit se lire 1940 à 2004. On peut mettre les axes au dessus (`slide=3`) et à droite (`slide=4`). Pour contrôler le nombre de lignes entre l'axe et les étiquettes, utiliser le paramètre `mgp`. Pour contrôler la position et la longueur des bâtons, utiliser le paramètre `tck`. Pour contrôler la taille des caractères, utiliser le paramètre `cex.axis` (figure 3-D). Pour contrôler l'orientation des étiquettes, utiliser le paramètre `las`.

Le problème posé concernait les échelles historiques. On peut marquer chaque année et laisser faire pour les labels (figure 3-E), où au contraire vouloir un maximum de dates et placer les étiquettes perpendiculaires à l'axe (figure 3-F). A noter :

★ pour avoir la liste des noms de paramètres (il y en a 73) `names(par())` ;

- ★ pour avoir leur définition `help(par)` ;
 - ★ pour avoir leur valeur par défaut `par("mfrow")`
 - ★ la fonction `axis` admet des valeurs pour ces paramètres mais n'en change pas la valeur par défaut après exécution.
- Voilà de quoi satisfaire les plus exigeants!

1. figure 3-A

```
w <- c(561.3, 522.3, 575.3, 615.8, 629.9, 645.9, 843.9, 870.5, 870.8,
      872.7, 862.3)
w <- c(w, c(826.7, 822.2, 804.7, 810.8, 805.9, 806.9, 816.5, 812.2,
      829.2, 819.8, 838.6))
w <- c(w, c(832.4, 868.9, 877.8, 865.7, 863.5, 840.6, 835.8, 842.2,
      850.4, 881.3, 877.5))
w <- c(w, c(857.2, 801.2, 745.1, 720.4, 744.7, 737.1, 757.4, 800.4,
      805.5, 797.2, 748.5))
w <- c(w, c(759.9, 768.4, 778.5, 767.8, 771.3, 765.5, 762.4, 759.1,
      743.7, 711.6, 711))
w <- c(w, c(729.6, 734.3, 726.8, 738.1, 744.8, 774.8, 770.9, 761.6,
      761.5, 764.7))
plot(w, type = "b", main = "A")
pretty(1:65)
pretty(w)
```

2. figure 3-B

```
plot(w, axes = F, main = "B")
box()
```

3. figure 3-C

```
plot(w, axes = F, main = "C", ylim = c(500, 1000), type = "n")
axis(2, seq(500, 1000, le = 11))
abline(h = seq(500, 1000, le = 11), col = grey(0.7))
points(w, pch = 19)
```

4. figure 3-D

```
plot(w, ax = F, main = "D")
box()
axis(1, mgp = c(3, 2, 0), tck = 0.5, cex.axis = 1.5, col = "red")
```

5. figure 3-E

```
plot(w, ax = F, pch = 19, main = "E")
box()
axis(1, seq(1, 65, by = 5), seq(1940, 2004, by = 5), lwd = 2)
abline(v = 1:65)
```

6. figure 3-F

```
plot(w, ax = F, type = "n", ylim = c(500, 1000), main = "F")
axis(3, seq(1, 65, by = 3), seq(1940, 2004, by = 3), lwd = 2, las = 2)
abline(v = 1:65, lty = 3)
abline(v = seq(1, 65, by = 3))
points(w, pch = 19, type = "b")
axis(2, seq(500, 1000, le = 11), tck = 1)
```

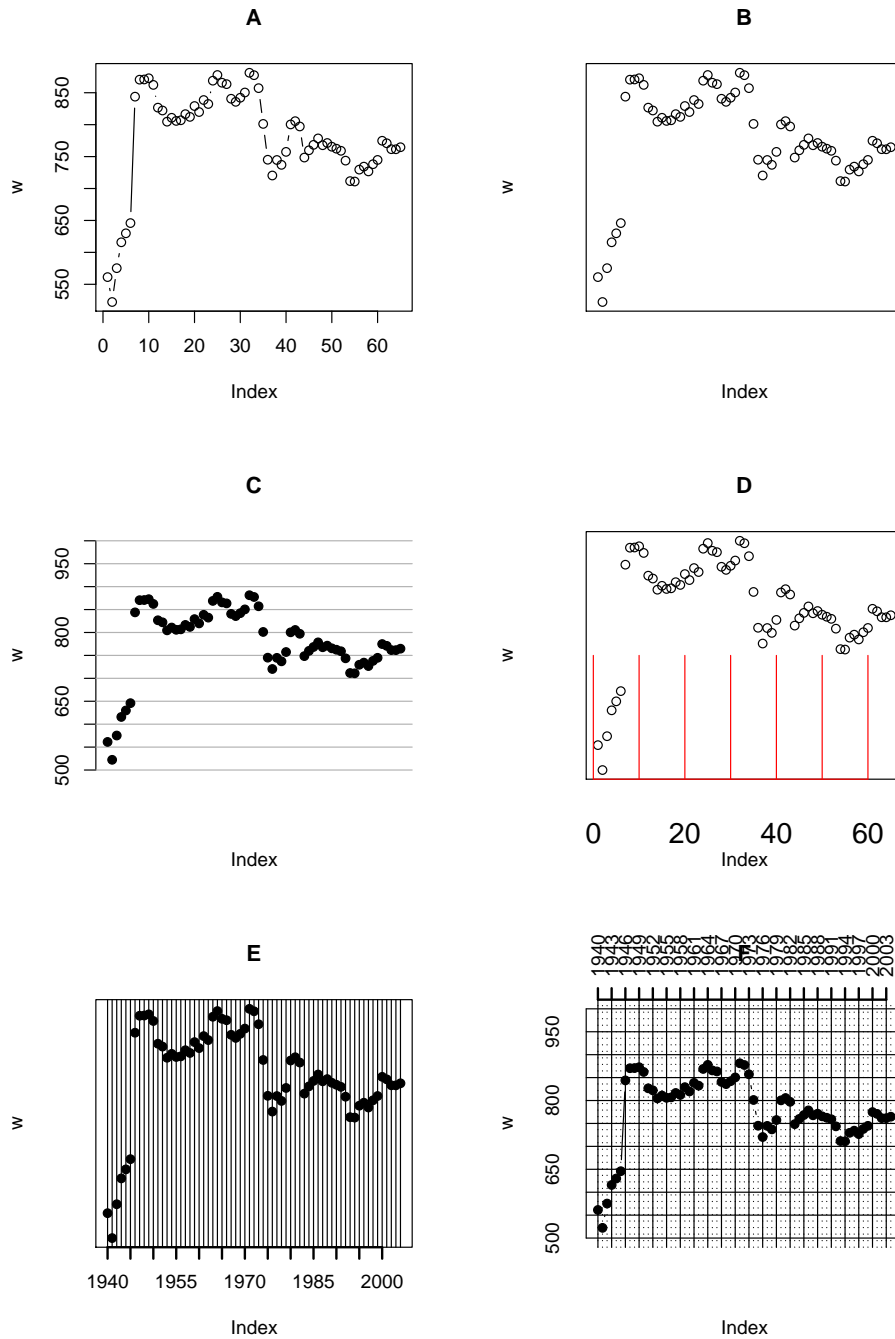


FIG. 3 – Mise en place des échelles sur les axes d'un dessin. Remarque : la chronique ne manque pas de sens !