

## Le fleuve Vam Co Tay : espace-temps sans répétition

M.P.L. Nguyen & D. Chessel

---

La fiche traite un exemple de données hydrobiologiques avec une mesure par date et par station. On utilise le modèle linéaire pour classer les variables par type de modèles spatio-temporels

### Table des matières

1	Les données	1
2	Les effets espace et temps	2
3	Modèles multiplicatifs	7
4	Synthèse	10

### 1 Les données

6 stations d'échantillonnage sont réparties sur le fleuve Vam Co Tay sur environ 120 km. Elles sont notées A, B, C, D, E et F *d'amont en aval*. Elles sont visitées à 6 reprises les 03/08/99 (t1), 20/09/99(t2), 07/11/99(t3), 12/03/00(t4), 28/04/00(t5) et 29/05/00(t6). Le plan est presque complet puisque seul manque le couple E/t1. Lire le fichier du plan d'échantillonnage et éditer l'objet `plan` :

```
plan <- read.table("http://pbil.univ-lyon1.fr/R/donnees/VamCoTayplan.txt",
  h = T)
options(show.signif.stars = FALSE)
```

A chaque visite une analyse physico-chimique de l'eau est effectuée et donne 12 variables :

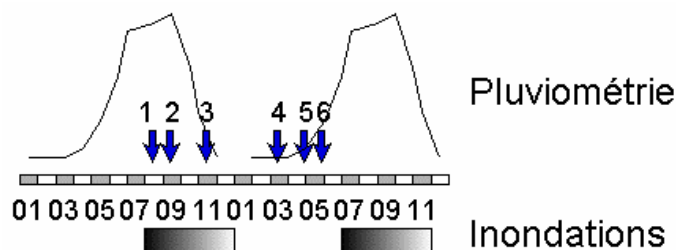
1. `ph` pH
2. `con` Conductivité ( $\mu\text{S}/\text{cm}$ )
3. `alc` Alcalinité (mg de  $\text{CaCO}_3/\text{l}$ )
4. `dbo` Demande biologique en oxygène à 5 jours (mg/l)
5. `dco` Demande chimique en oxygène (mg/l)

6. no3 Nitrates (mg/l)
7. no2 Nitrites (mg/l)
8. nh4 Azote ammoniacal (mg/l)
9. po4 Phosphates (mg/l)
10. mes Matières en suspension (mg/l)
11. tem Température de l'eau (°C)
12. tra Transparence (disque de Secchi en cm)

Obtenir et éditer le tableau à 35 lignes et 12 colonnes :

```
chimi <- read.table("http://pbil.univ-lyon1.fr/R/donnees/VamCoTaychimi.txt",
  h = T)
```

Le système étudié est caractérisé par une saison des pluies (mai à octobre) et une saison sèche (novembre à avril) et une inondation annuelle qui commence en juillet et peut durer suivant les années de 1 à 5 mois. Schéma de principe :



## 2 Les effets espace et temps

Éditer `plan$tem` et `plan$sta` puis créer et éditer les deux variables numériques :

```
tempo <- as.numeric(plan$tem)
numsta <- as.numeric(plan$sta)
```

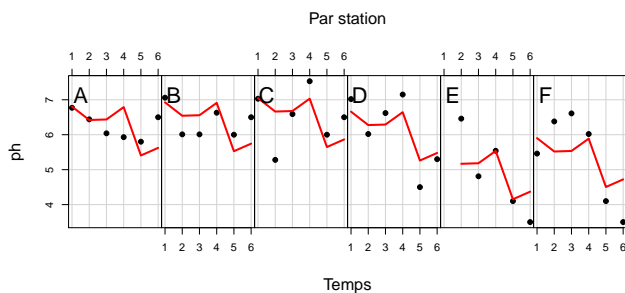
Étudier la fonction qui suit en prévoyant ce qu'elle va faire (utiliser l'aide en ligne) :

```
additif <- function(numvar = 1) {
  f3 <- function(x, y, subscripts, col, pch, z) {
    points(x, y, pch = 20, cex = 1.5)
    lines(x, z[subscripts], lwd = 2, col = "red")
    w1 <- 0.05 * par("usr")[3] + 0.95 * par("usr")[4]
    text(1.5, w1, as.character(unique(plan$sta[subscripts])),
      cex = 2, pos = 1)
  }
  namevar <- names(chimi[numvar])
  z <- chimi[, numvar]
  lm1 <- lm(z ~ plan$tem + plan$sta)
  print(namevar)
  print(anova(lm1))
  coplot(chimi[, numvar] ~ tempo | plan$sta, sh = F, colu = 6,
    sub = T, panel = f3, xlab = c("Temps", "Par station"), ylab = namevar,
    z = predict(lm1))
}
```

Étudier chacune des variables avec une analyse de variance.

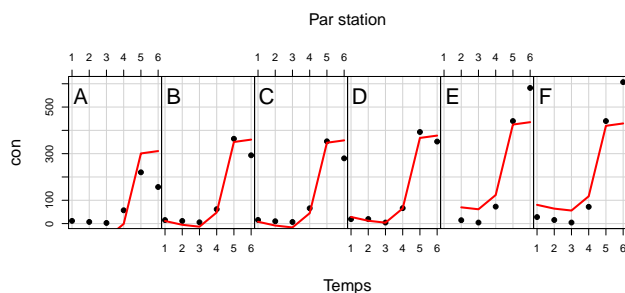
## 01 pH

```
[1] "ph"
Analysis of Variance Table
Response: z
      Df Sum Sq Mean Sq F value Pr(>F)
plan$tem  5 11.5047  2.3009  4.0915 0.007911
plan$sta  5 10.0762  2.0152  3.5835 0.014611
Residuals 24 13.4968  0.5624
```



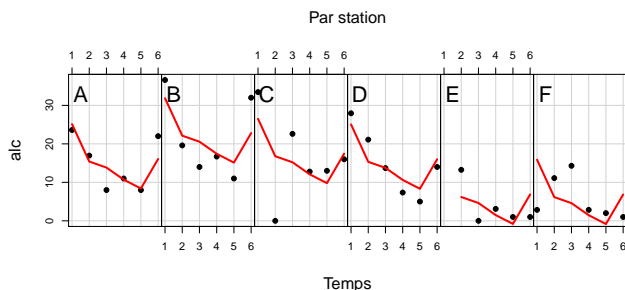
## 02 Conductivité

```
[1] "con"
Analysis of Variance Table
Response: z
      Df Sum Sq Mean Sq F value Pr(>F)
plan$tem  5 967126 193425 35.463 2.509e-10
plan$sta  5  63078  12616  2.313  0.07533
Residuals 24 130902  5454
```



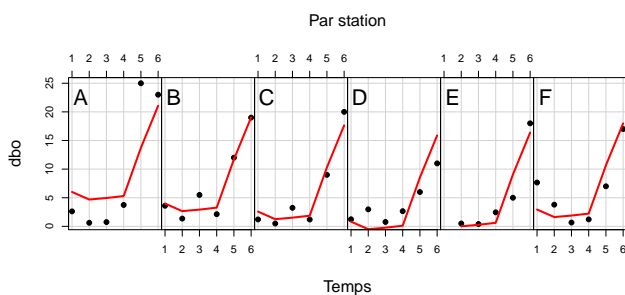
## 03 Alcalinité

```
[1] "alc"
Analysis of Variance Table
Response: z
      Df Sum Sq Mean Sq F value Pr(>F)
plan$tem  5 1062.90  212.58  4.4088 0.005462
plan$sta  5 1128.97  225.79  4.6828 0.003996
Residuals 24 1157.22  48.22
```



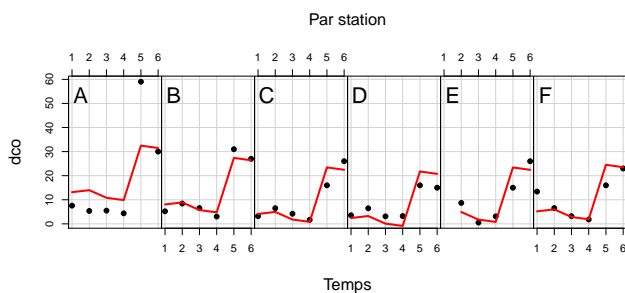
#### 04 Demande biologique en oxygène

```
[1] "dbo"
Analysis of Variance Table
Response: z
      Df Sum Sq Mean Sq F value Pr(>F)
plan$tem  5 1328.98  265.80 20.0473 7.297e-08
plan$sta  5  103.04   20.61  1.5543  0.2109
Residuals 24   318.20   13.26
```



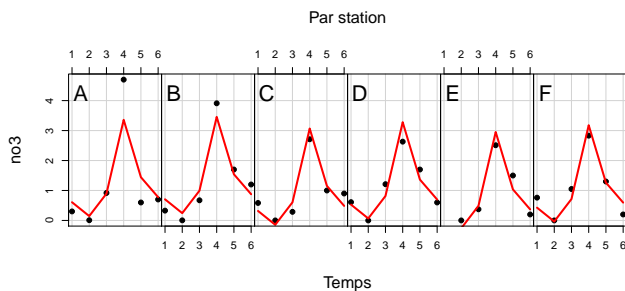
#### 05 Demande chimique en oxygène

```
[1] "dco"
Analysis of Variance Table
Response: z
      Df Sum Sq Mean Sq F value Pr(>F)
plan$tem  5 3226.5  645.3 11.7137 8.258e-06
plan$sta  5  450.4   90.1  1.6353  0.1889
Residuals 24 1322.1   55.1
```



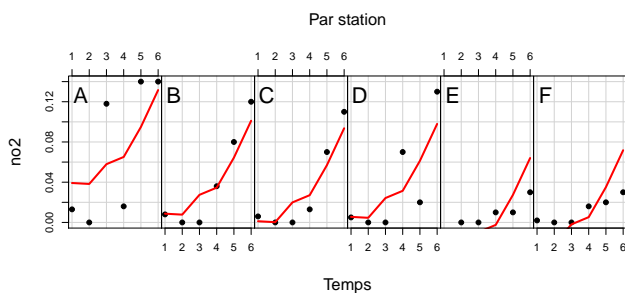
#### 06 Nitrates

```
[1] "no3"
Analysis of Variance Table
Response: z
      Df Sum Sq Mean Sq F value Pr(>F)
plan$tem  5  38.033  7.607 33.0658 5.18e-10
plan$sta  5  0.970  0.194  0.8437  0.5323
Residuals 24  5.521  0.230
```



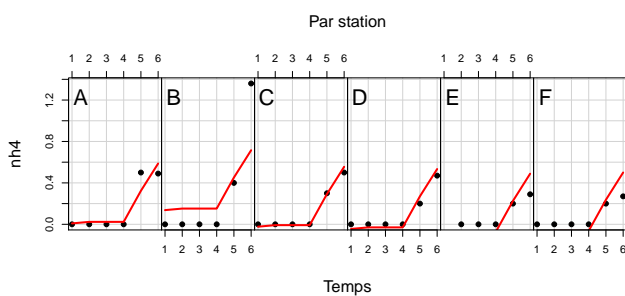
## 07 Nitrites

```
[1] "no2"
Analysis of Variance Table
Response: z
      Df Sum Sq Mean Sq F value Pr(>F)
plan$tem  5 0.036366  0.007273  7.5211 0.0002273
plan$sta  5 0.016184  0.003237  3.3472 0.0196027
Residuals 24 0.023209  0.000967
```



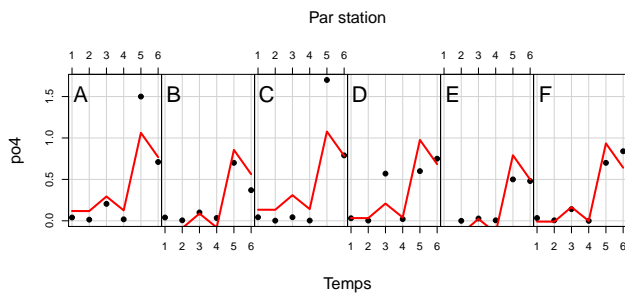
## 08 Azote ammoniacal

```
[1] "nh4"
Analysis of Variance Table
Response: z
      Df Sum Sq Mean Sq F value Pr(>F)
plan$tem  5 1.67743  0.33549 11.5790 9.07e-06
plan$sta  5 0.19816  0.03963  1.3679  0.2712
Residuals 24 0.69537  0.02897
```



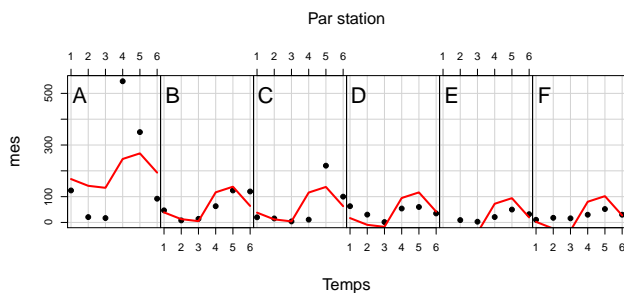
## 09 Phosphates

```
[1] "po4"
Analysis of Variance Table
Response: z
      Df Sum Sq Mean Sq F value Pr(>F)
plan$tem  5 4.7325  0.9465 16.9330 3.471e-07
plan$sta  5 0.3532  0.0706  1.2638  0.3116
Residuals 24 1.3415  0.0559
```



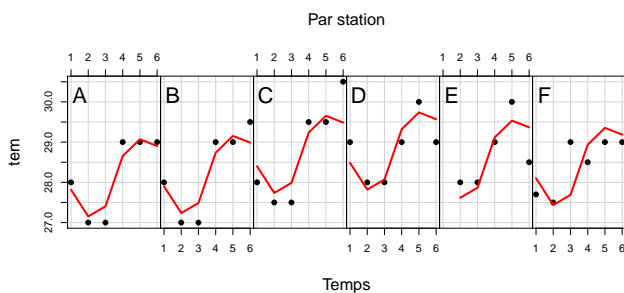
## 10 Matières en suspension

```
[1] "mes"
Analysis of Variance Table
Response: z
      Df Sum Sq Mean Sq F value Pr(>F)
plan$tem  5  87732   17546  2.2149 0.08597
plan$sta  5 119346   23869  3.0130 0.02998
Residuals 24  190129     7922
```



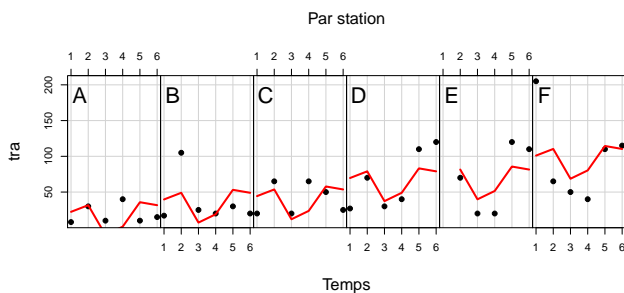
## 11 Température de l'eau

```
[1] "tem"
Analysis of Variance Table
Response: z
      Df Sum Sq Mean Sq F value Pr(>F)
plan$tem  5 19.9257   3.9851 14.1206 1.728e-06
plan$sta  5  2.1770   0.4354  1.5428  0.2142
Residuals 24  6.7733   0.2822
```



## 12 Transparence

```
[1] "tra"
Analysis of Variance Table
Response: z
      Df Sum Sq Mean Sq F value Pr(>F)
plan$tem  5  10462   2092  1.4524 0.24205
plan$sta  5  23654   4731  3.2839 0.02123
Residuals 24  34574   1441
```



Bilan : toutes les variables évoluent dans le temps et/ou dans l'espace. La moitié des variables ont une évolution purement temporelle. Reconstituer la figure 1.

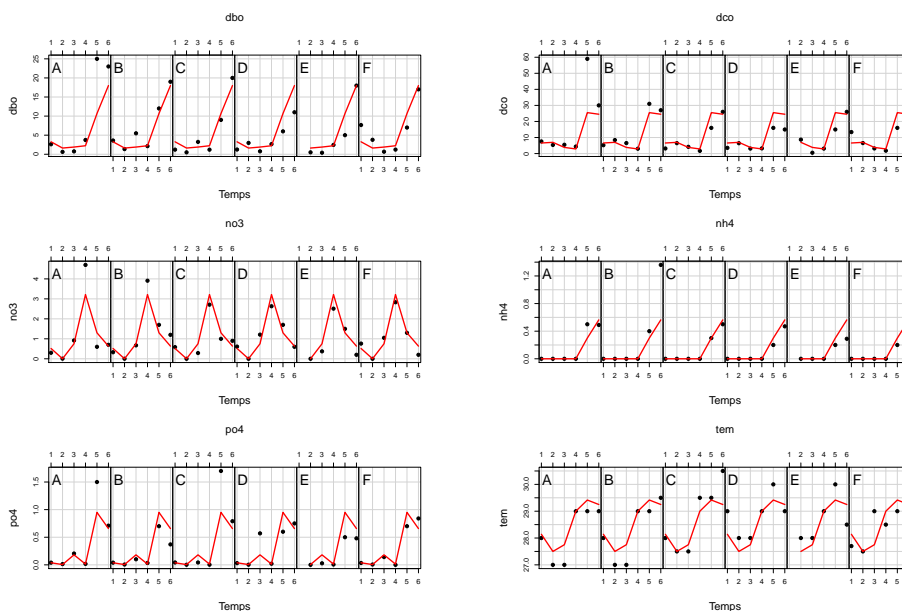


FIG. 1 – Modèles à variation temporelle simple. En chaque station, le fleuve engendre les mêmes variations du paramètre.

### 3 Modèles multiplicatifs

Les autres variables ont une variation double qui posent un problème beaucoup plus difficile.

```

moddouble <- function(numvar = 1) {
  f3 <- function(x, y, subscripts, col, pch, z, znew) {
    points(x, y, pch = 20, cex = 1.5)
    lines(x, znew[subscripts], lty = 1, lwd = 2, col = "red")
    lines(x, z[subscripts], lty = 2, lwd = 1, col = "blue")
    w2 <- 0.05 * par("usr")[3] + 0.95 * par("usr")[4]
    text(1.5, w2, as.character(unique(statot[subscripts])),
         cex = 2, pos = 1)
  }
  namevar <- names(chimi[numvar])
  z <- chimi[, numvar]
  x <- plan$tem
  y <- plan$sta
  lm1 <- lm(z ~ x + y)
  w1 <- predict(lm1, newdata = list(x = as.factor("t1"), y = as.factor("E")))
  ztot <- c(z[1:4], w1, z[5:35])
  temtot <- as.factor(paste(rep("t", 36), rep(1:6, rep(6, 6)),
                           sep = ""))
  statot <- as.factor(rep(LETTERS[1:6], 6))
  A <- matrix(ztot, 6, 6)
  w <- svd(A)
  zest <- as.vector((w$u[, 1] * w$d[1]) %*% t(w$v[, 1]))
  lm2 <- lm(ztot ~ temtot + statot)
  cplot(ztot ~ as.numeric(temtot) | statot, sh = F, colu = 6,
        sub = T, panel = f3, z = predict(lm2), znew = zest, xlab = c("Temps",

```

```

    namevar), ylab = namevar)
}

```

Deux variables sont à effet additif simple. Reconstituer la figure 2.

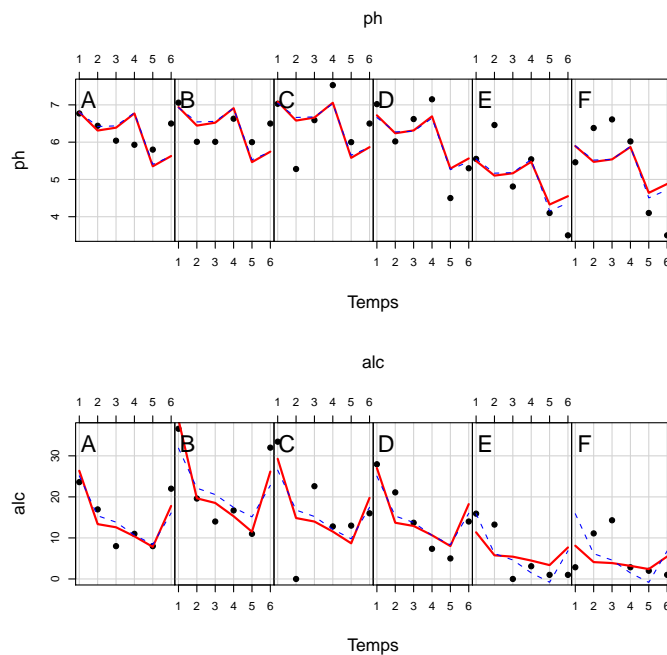


FIG. 2 – Variables de milieu à évolution spatio-temporelle additive. En tirets, le modèle additif, en plein le modèle multiplicatif. Les performances sont très voisines et le modèle additif est retenu. En chaque station, le fleuve engendre des variations globales du paramètre auxquelles s'ajoutent une constante locale.

Quatre variables sont à effet multiplicatif. Reconstituer la figure 3.



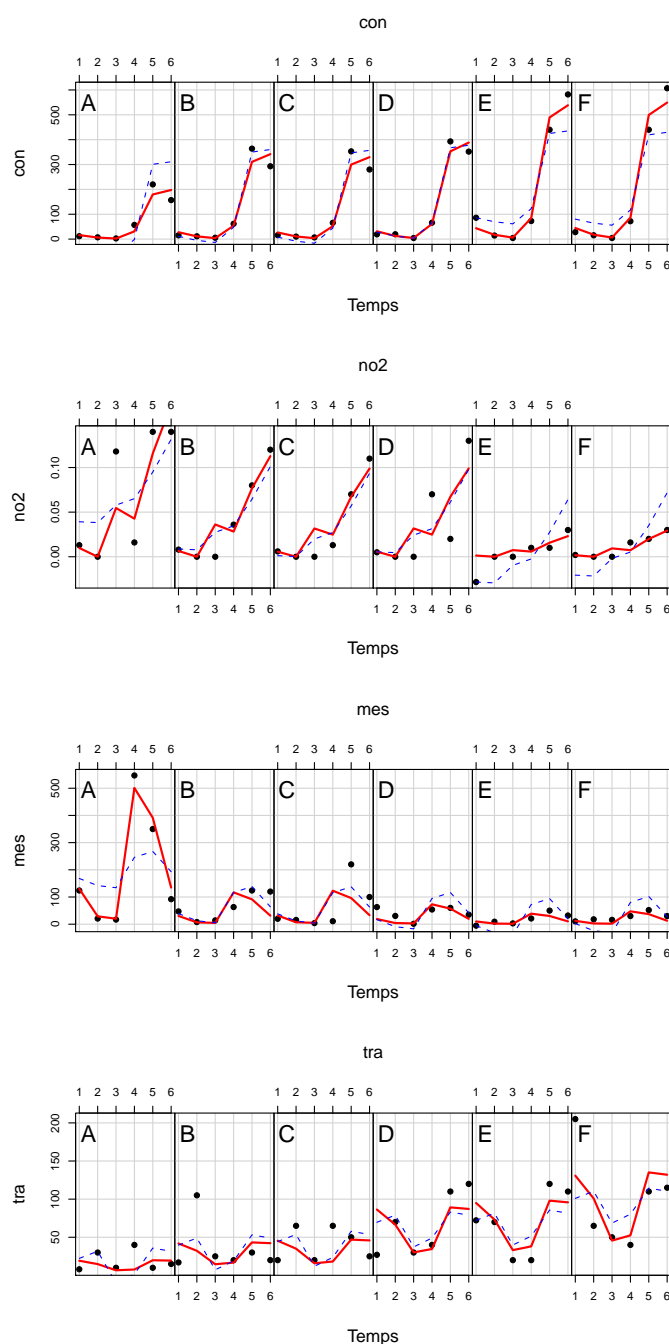
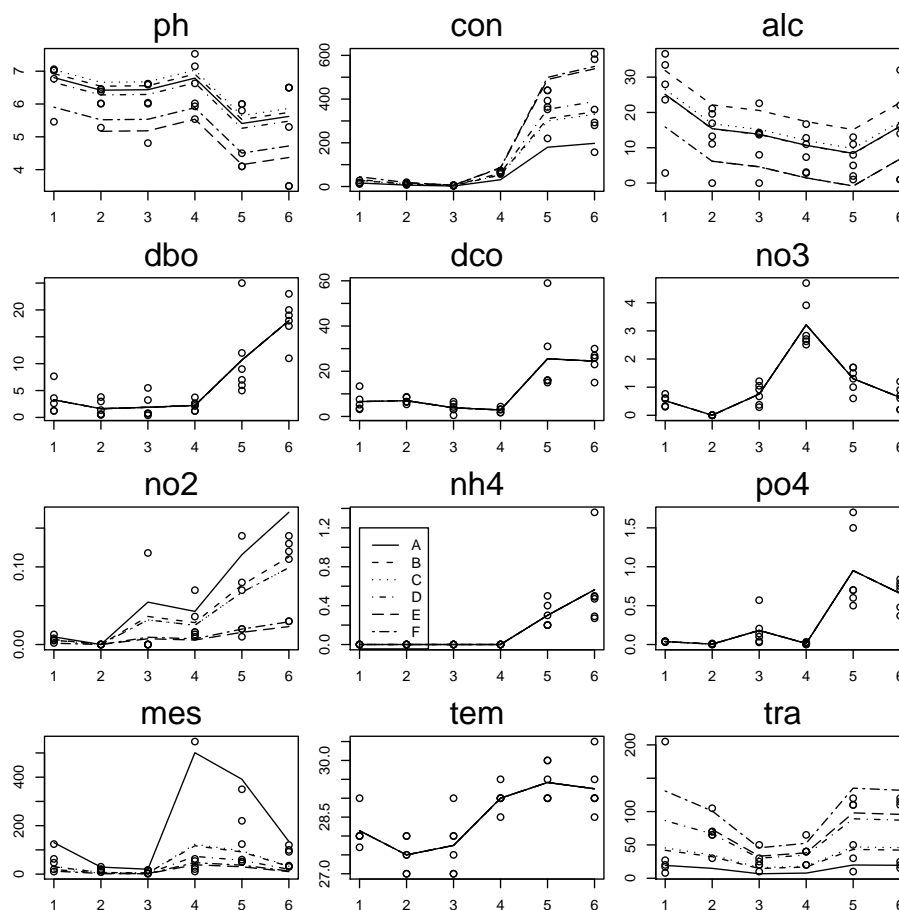


FIG. 3 – Variables de milieu à évolution spatio-temporelle multiplicative. En tirets, le modèle additif, en plein le modèle multiplicatif. Les performances sont très différentes et le modèle additif est invalide. On a de remarquables exemples expérimentaux de la notion d'interaction. En chaque station, le fleuve engendre des variations globales du paramètre auxquelles modifiée en chacune des stations.

## 4 Synthèse

Faire une figure pour résumer ce qui précède.



```
plotdef <- function() {
  type <- c(2, 3, 2, 1, 1, 1, 3, 1, 1, 3, 1, 3)
  par(mfrow = c(4, 3))
  par(mar = c(3, 2, 2, 1))
  x <- plan$tem
  y <- plan$sta
  for (i in 1:12) {
    z <- chimi[, i]
    if (type[i] == 1)
      zest <- predict(lm(z ~ x))
    if (type[i] == 2)
      zest <- predict(lm(z ~ x + y))
    if (type[i] == 3) {
      lm1 <- lm(z ~ x + y)
      w1 <- predict(lm1, newdata = list(x = as.factor("t1"),
        y = as.factor("E")))
      ztot <- c(z[1:4], w1, z[5:35])
      temtot <- as.factor(paste(rep("t", 36), rep(1:6, rep(6,
        6)), sep = ""))
      statot <- as.factor(rep(LETTERS[1:6], 6))
      A <- matrix(ztot, 6, 6)
      w <- svd(A)
      zest <- as.vector((w$u[, 1] * w$d[1]) %*% t(w$v[, 1]))
      zest <- zest[-5]
    }
  }
}
```

```
    }  
    ry <- range(c(zest, z))  
    plot(as.numeric(x), z, type = "p", ylim = ry, xlab = "Temps",  
         ylab = names(chimi)[i])  
    for (j in 1:6) {  
      lines(x[y == levels(y)[j]], zest[y == levels(y)[j]],  
           lty = j)  
    }  
    mtext(names(chimi)[i], side = 3, line = 0.5, cex = 1.5)  
    if (i == 8)  
      legend(1, 1.2, LETTERS[1:6], lty = 1:6)  
  }  
}
```

Comparer les performances graphiques des juxtapositions et des superpositions.