Groupe: 4A IAD Groupe 1

Edouard NADAUD Guilhem NESPOULOUS Aymeric NOBLANC

1. Définir un modèle additif avec:

- une tendance quadratique (c'est-à-dire du type at^2 +bt+c)
- une composante saisonnière fonction périodique de période 4
- une erreur de type bruit blanc gaussien Pour simuler une série temporelle trimestrielle débutant au deuxième trimestre 1986 et finissant au premier trimestre 2001. Nous choisirons les différents paramètres du modèle : coefficients de la tendance (a, b et c), expression de la saisonnalité et écart-type du bruit blanc).

$(at^2 +bt+c)+\cos(2\pi(t/4 - 1986))+ \varepsilon\ t$

commence au 2eme semestre 1986 soit avril 1986 fin premier semestre 2001 soit janvier 2001 * Nombre de mois = 13*12+9=165, notre bruit gaussien fait aller de 1 a 165 de même pour nos données (167),* on choisit un écart type de 1

```
Modele additif Xt=Zt+St+εt

    Zt=(0.0001*i^2+0.04*i+2)
    St=cos(2*pi*((i/4 )- 1986))
    εt=erreur[i]
```

```
data=array()
pi=3.14
nbrmonth=167
erreur<-rnorm(nbrmonth,sd=1,mean=0)
```

On choisit a=0.0001 b=0.04 c=2

Creation de notre serie temporelle numérique

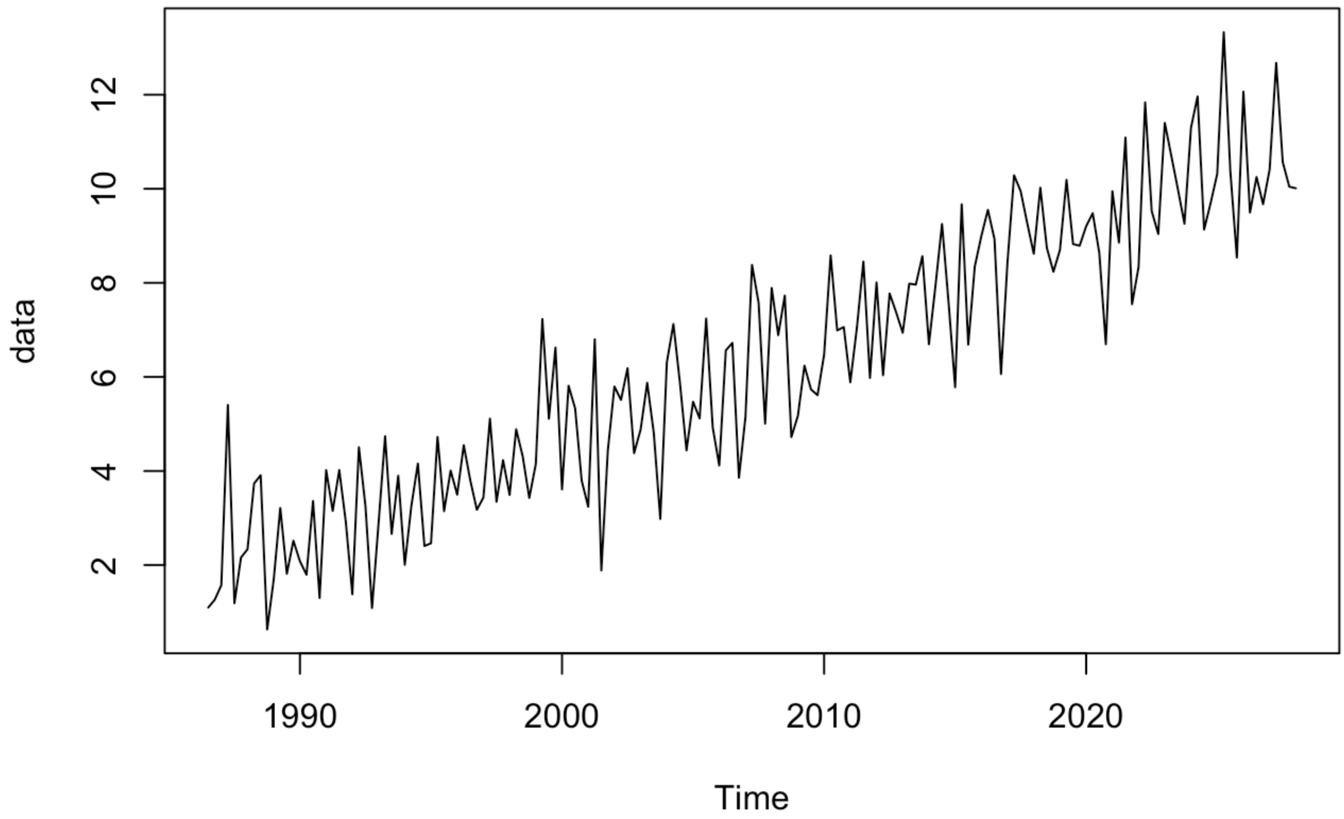
```
for (i in 1:nbrmonth){
  data[i]= (0.0001*i^2+0.04*i+2)+cos(2*pi*((i/4 )- 1986))+erreur[i]

}
```

On faitcommencer notre serie temporelle au mois de mars 1986

```
data<-as.ts(data)

data<-ts(data,start=c(1986,3),frequency=4)
```

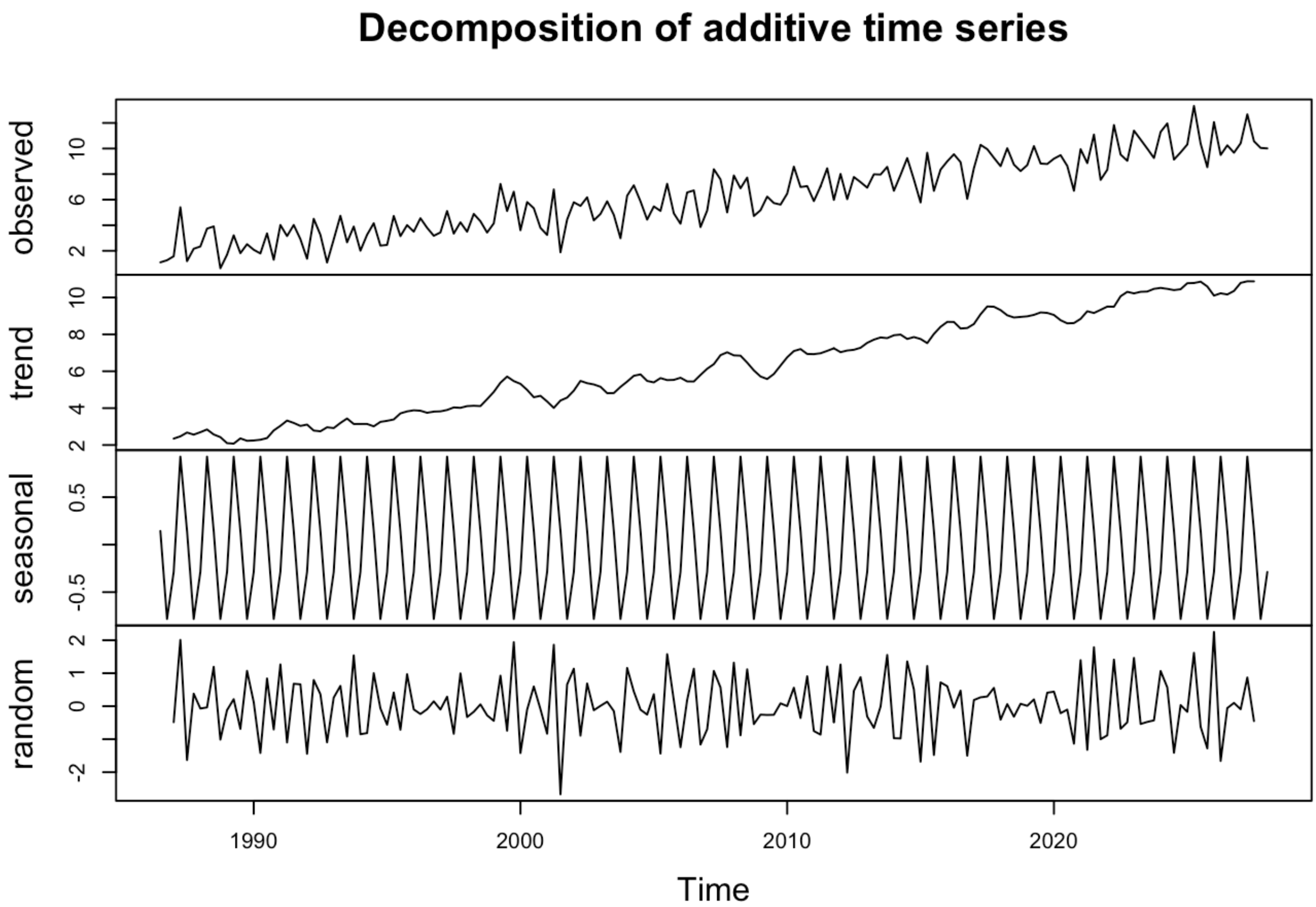
Affichage de notre serie temporelle

```
ts.plot(data)
```



2. Afficher les différentes composantes. Tracer la tendance, la saisonnalité, le bruit ainsi que la série chronologique obtenue.

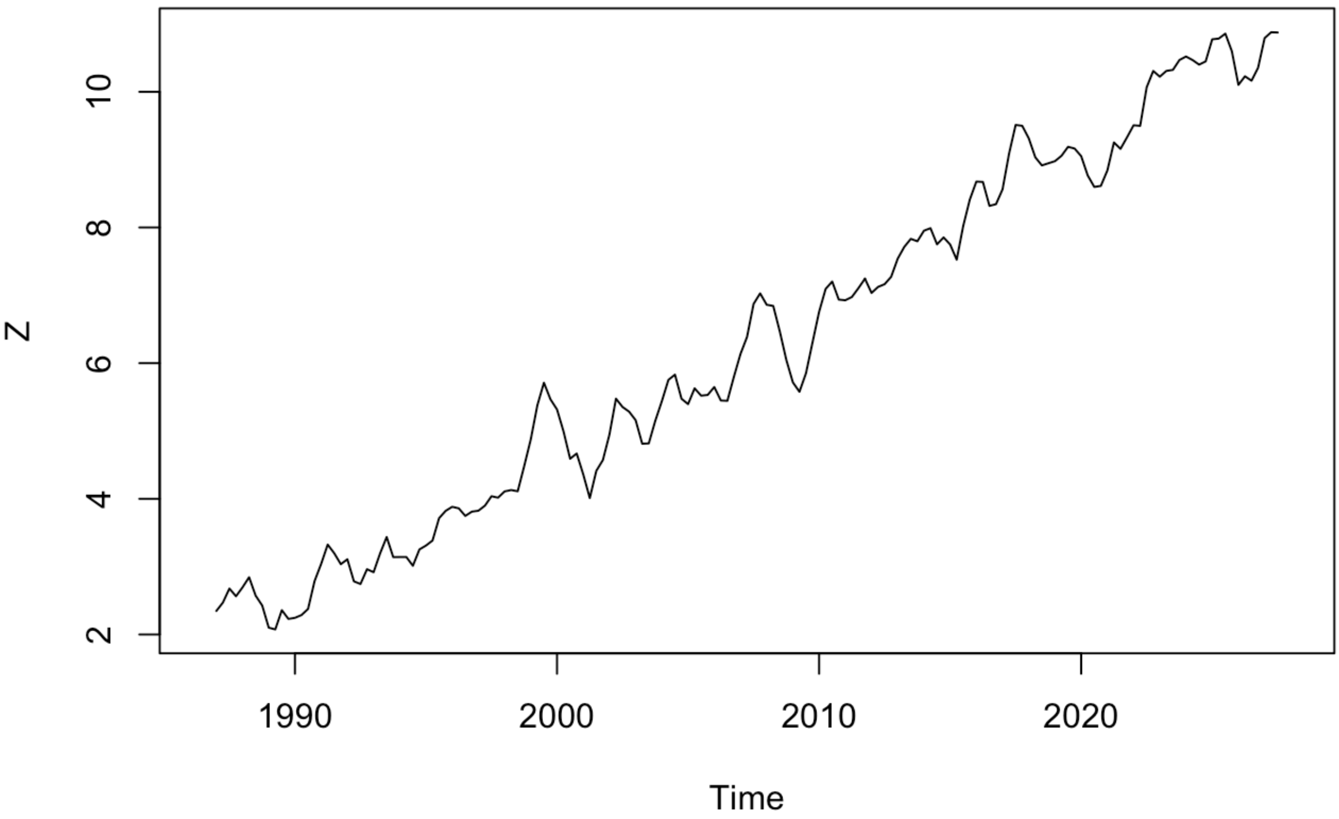
```
data.dcp= decompose(data,type="additive")
plot(data.dcp)
```




```
help(decompose)
```

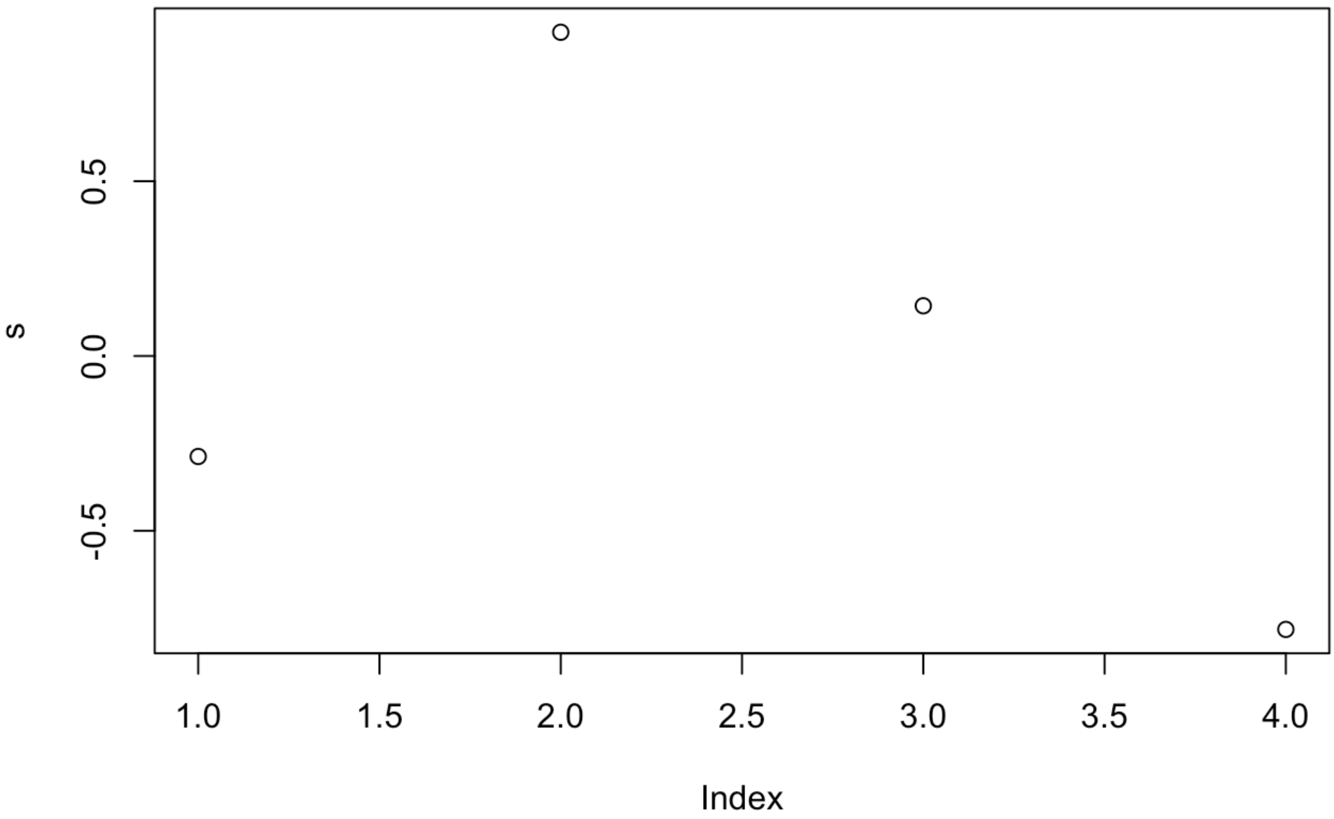
Pour estimer la tendance nous allons calculer nos moyennes mobiles qui seront un bon estimateur de la tendance (avec le bruit en plus)

```
filt<-rep(1/4,3)
filt<-(c(1/8,filt,1/8))
Z<-filter(data,filter=filt,sides=2)
Z<-ts(Z,start=c(1986,3),frequency=4)
ts.plot(Z)
```



Nous allons maintenant optenir la saisonalité pour cela nouss faisons nos data moins la tendance et le bruit gaussien (Z)

```
S<-data-Z
s<-tapply(S,cycle(S),mean,na.rm=T)
s<-s-mean(s)
plot(s)
```

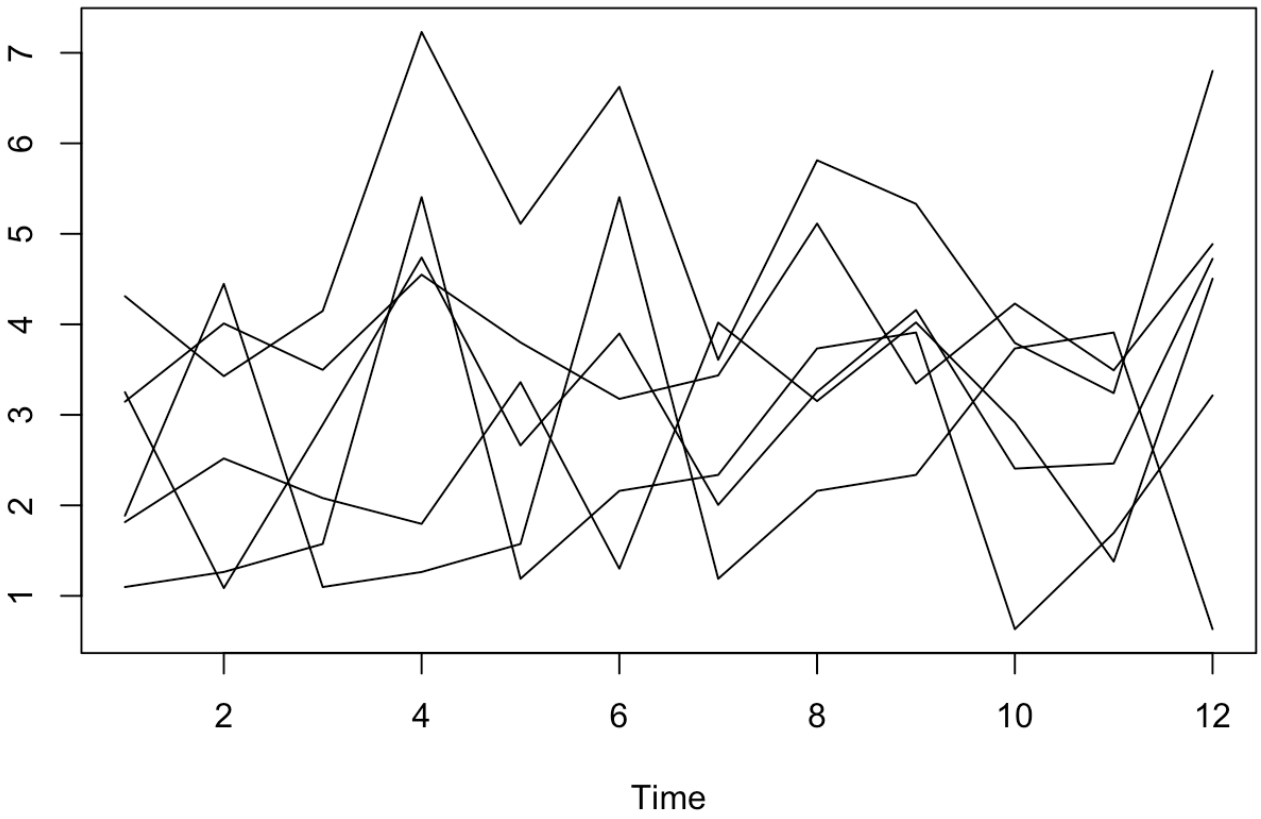


3. Superposer les séries annuelles des années 1987 à 2000.

```
ts.plot(matrix(window(data,1986,c(1999,12)),12))
```

```
## Warning in window.default(x, ...): 'start' value not changed
```

```
## Warning in matrix(window(data, 1986, c(1999, 12)), 12): la longueur des données
## [62] n'est pas un diviseur ni un multiple du nombre de lignes [12]
```



4. Extraire une sous série commençant au deuxième trimestre 1989 et finissant au troisième trimestre 1991.

```
Xdata=window(data,start=c(1989,4),end=c(1991,9))
plot(Xdata)
```

Groupe: 4A IAD Groupe 1

Edouard NADAUD Guilhem NESPOULOUS Aymeric NOBLANC

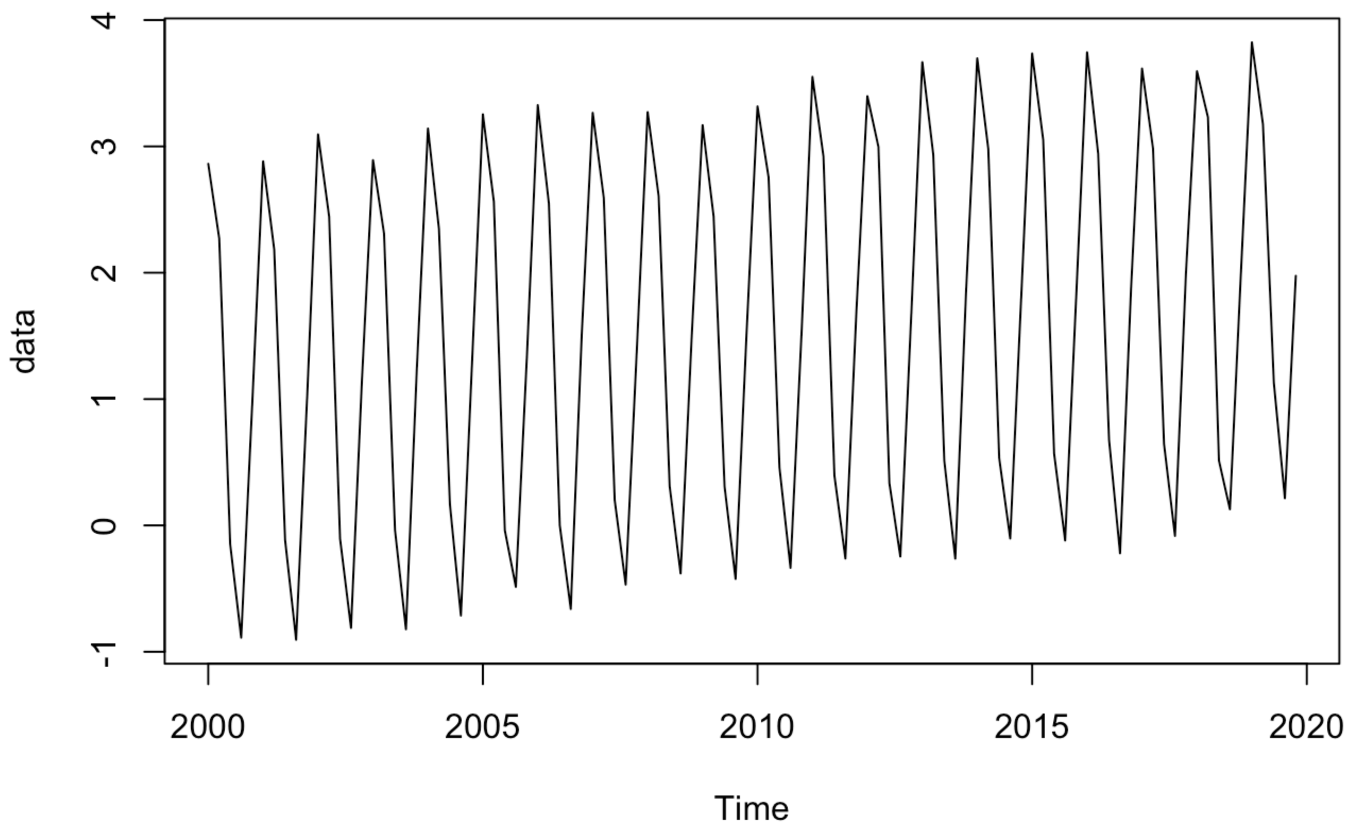
Exercice : à rendre pour le dimanche 7 mars 2021 à 23h30 Simuler une série chronologique(Yt)t=1,…,100 suivant le modèle $Yt = 0{,}01t + 1 + 2 \sin(2\pi t/5) + \varepsilon t$, où$(\varepsilon t)t$ est un bruit blanc gaussien de variance 1/100.

```
data=array()
erreur<-rnorm(100,sd=1/10,mean=0)
```

```
for (i in 1:100){
  data[i]= (0.01*i)+1+(2*sin((2*pi*i)/5) )+ erreur[i]
}
```

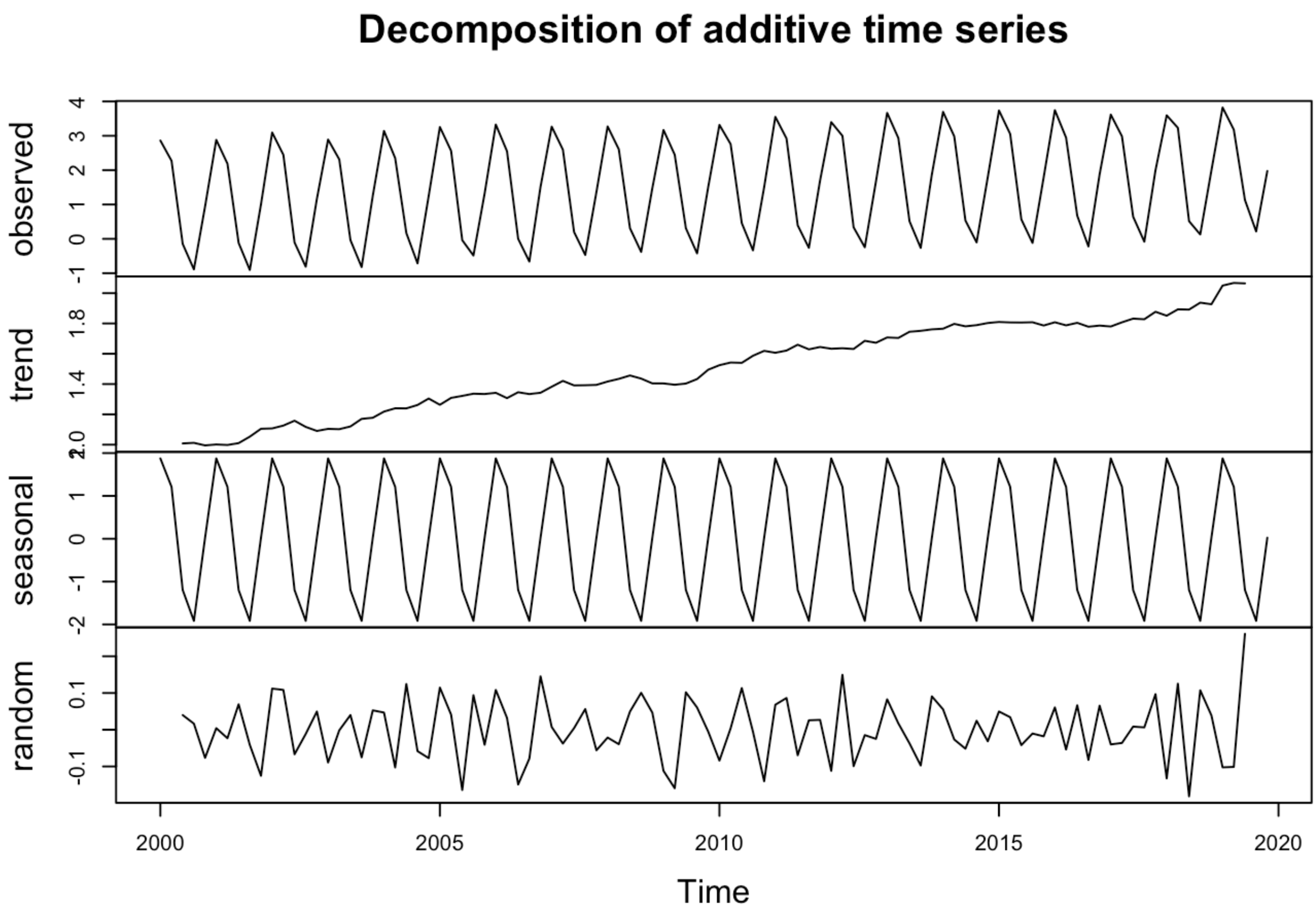
On affiche notre serie temporelle

```
data<-as.ts(data)
data<-ts(data,start=2000,frequency=5)
ts.plot(data)
```



Déterminer la tendance, la saisonnalité (période) de cette série chronologique. Les tracer.

```
data.dcp= decompose(data,type="add")
plot(data.dcp)
```

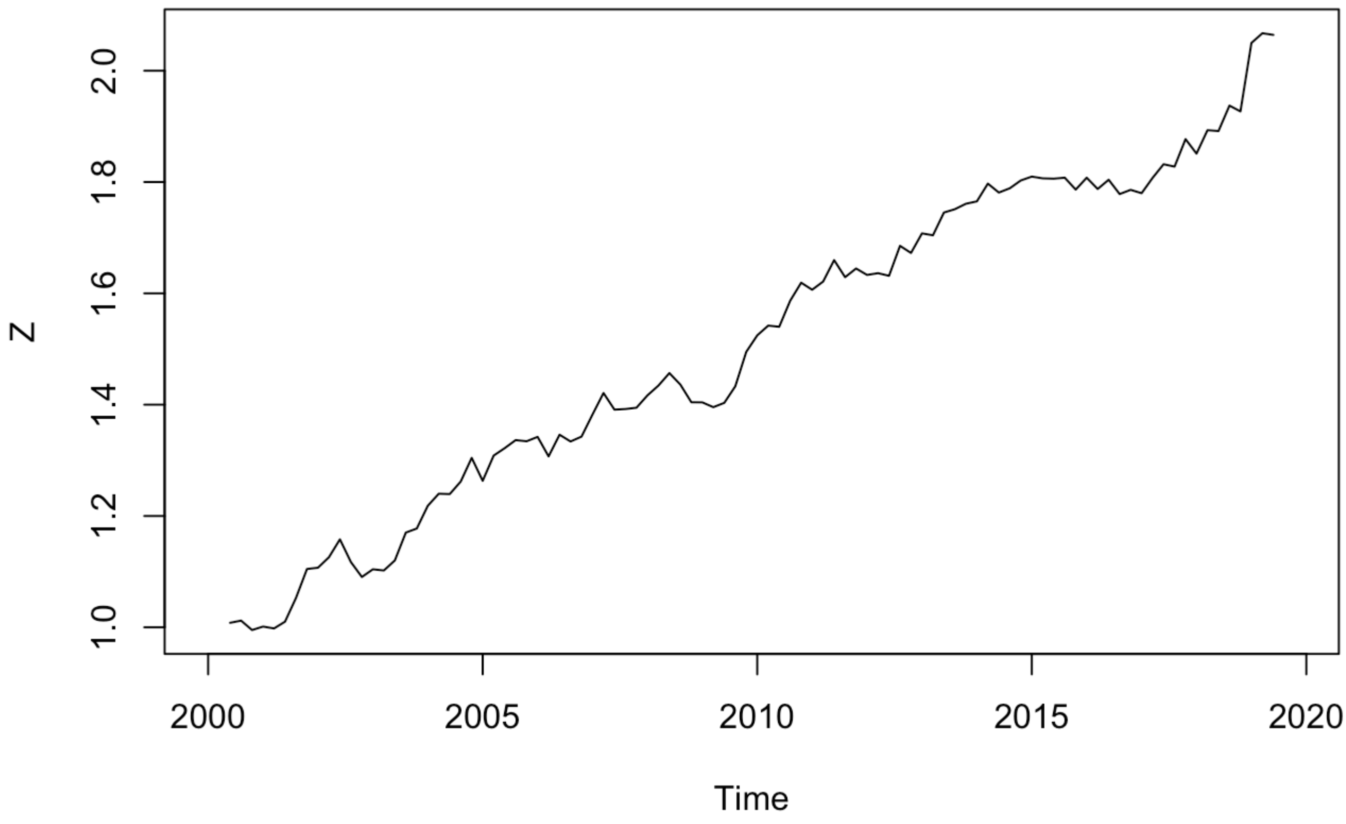


Notre serie temporelle est un modele aditif de forme

```
Modele additif Xt=Zt+St+Ɛt

    Zt=(0.01*i)+1
    St=(2*sin((2*pi*i)/5))
    Ɛt=erreur[i]



    Notre St est une fonction de periode 5 nous avons donc une saisonalité de 5. No
tre Zt est une fonction affiche de coéfficient directeur 0.01, notre tendance sera
donc croisante. Néanmoins le coéfficient directeur etant tres faible la fonction au
ra ue croissante extrement faible et donc une tendance plutot constante.
```

Utiliser la méthode des moyennes mobiles ci-dessus pour éliminer la saisonnalité puis estimer les coefficients du saisonnier.

```
filt<-rep(1/5,5)
Z<-filter(data,filter=filt,sides=2)
Z<-ts(Z,start=2000,frequency=5)
ts.plot(Z)
```

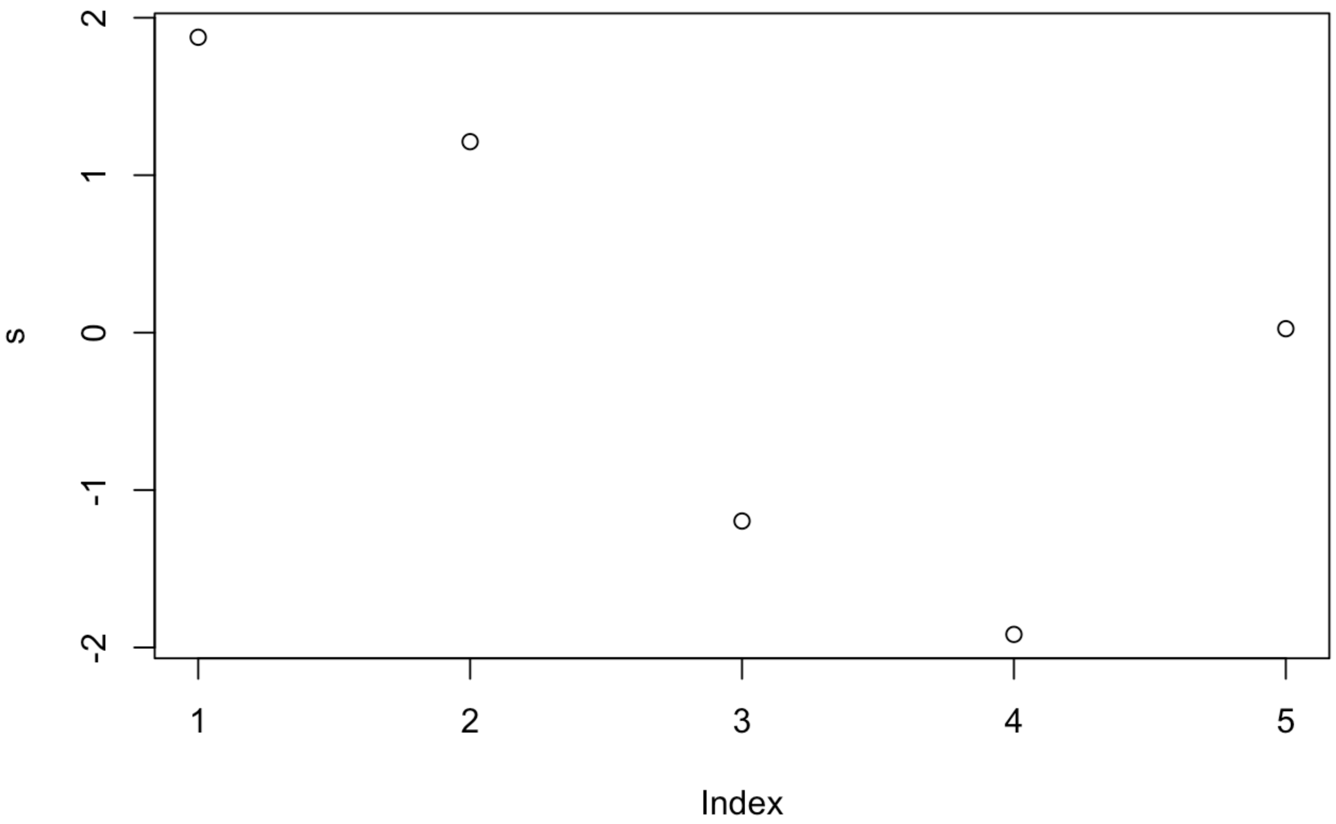


Estimation de la composante saisonnière S la saisonnalité

```
S<-data-Z
```

Estimation c des coefficients du saisonnier Série corrigée des variations saisonnières Voici la saisonalité sur une periode 5

```
s<-tapply(S,cycle(S),mean,na.rm=T)
s<-s-mean(s)
plot(s)
```




```
CVS<-matrix(1,20,5)
```

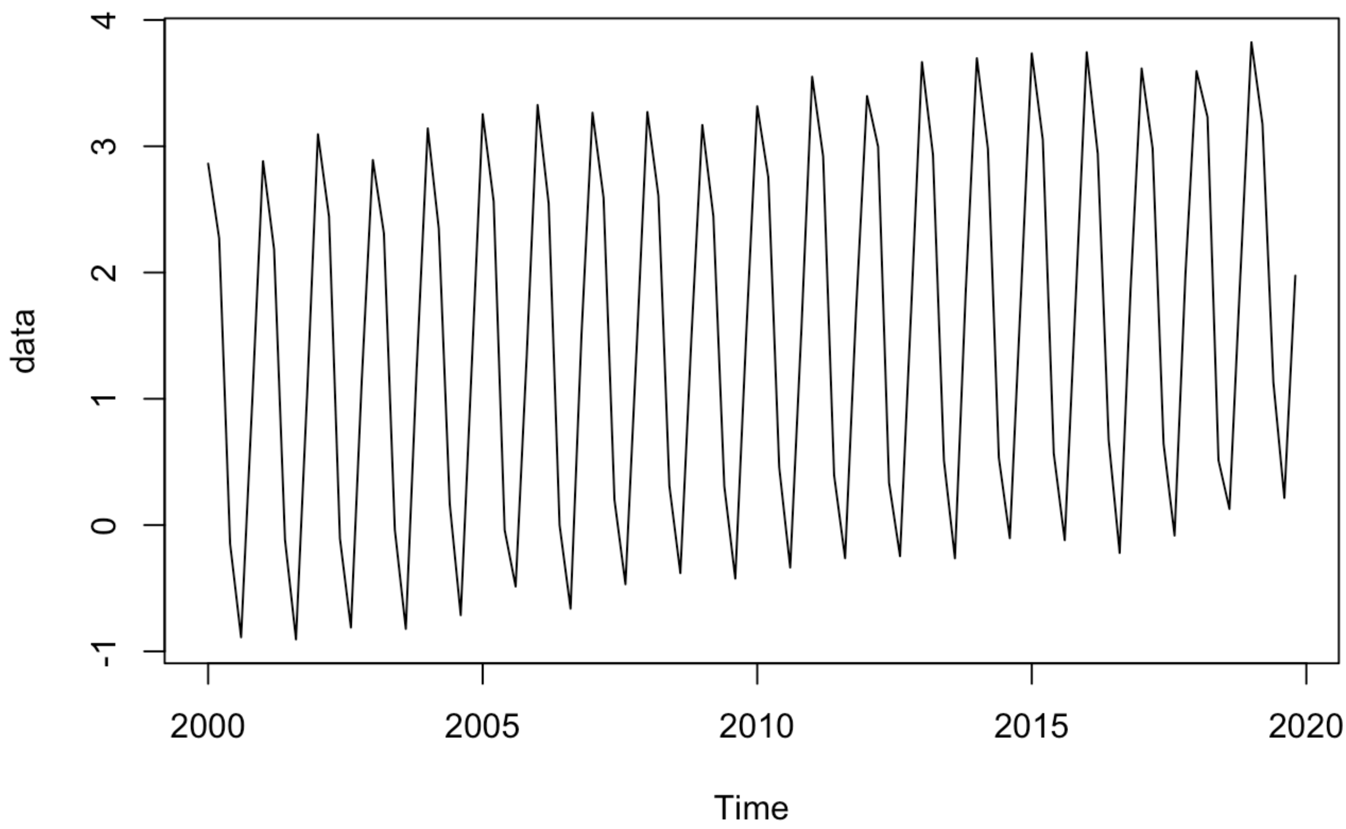
Calcul de XCV S = X – S

Nous avons 5 données par an su 20 ans

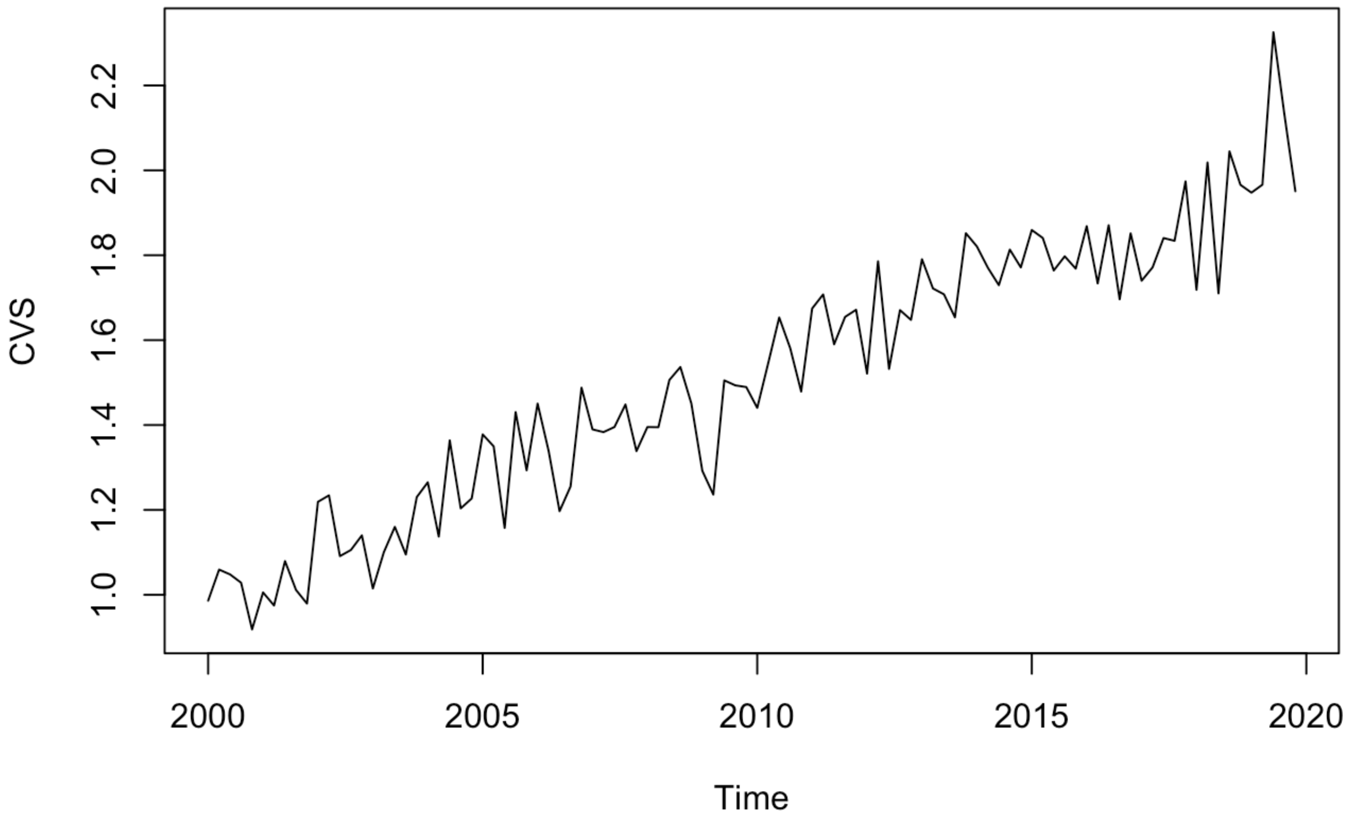
```
for (i in 1 :20) {
  for (j in 1 :5) {
    CVS[i,j]=t(matrix(data,5,20))[i,j]-s[j]
  }
}
```

```
CVS=as.vector(t(CVS))
CVS=as.ts(CVS)
CVS=ts(CVS,start=2000,frequency=5)
```

```
ts.plot(data)
```




```
ts.plot(CVS)
```



coefficient saisonnier

```
s
```

```
##           1           2           3           4           5
##  1.87612091  1.21281802 -1.19676170 -1.91701991  0.02484268
```

Utiliser une régression linéaire par moindres carrés pour estimer les coefficients de la tendance.

```
y=time(CVS)
CVS.lm=lm(CVS~y)
CVS.lm$coefficient
```

```
##   (Intercept)             y
## -102.97832740    0.05198707
```

Comparer les estimateurs avec les vrais coefficients.

yreel=0.01 ycalculé=0.05

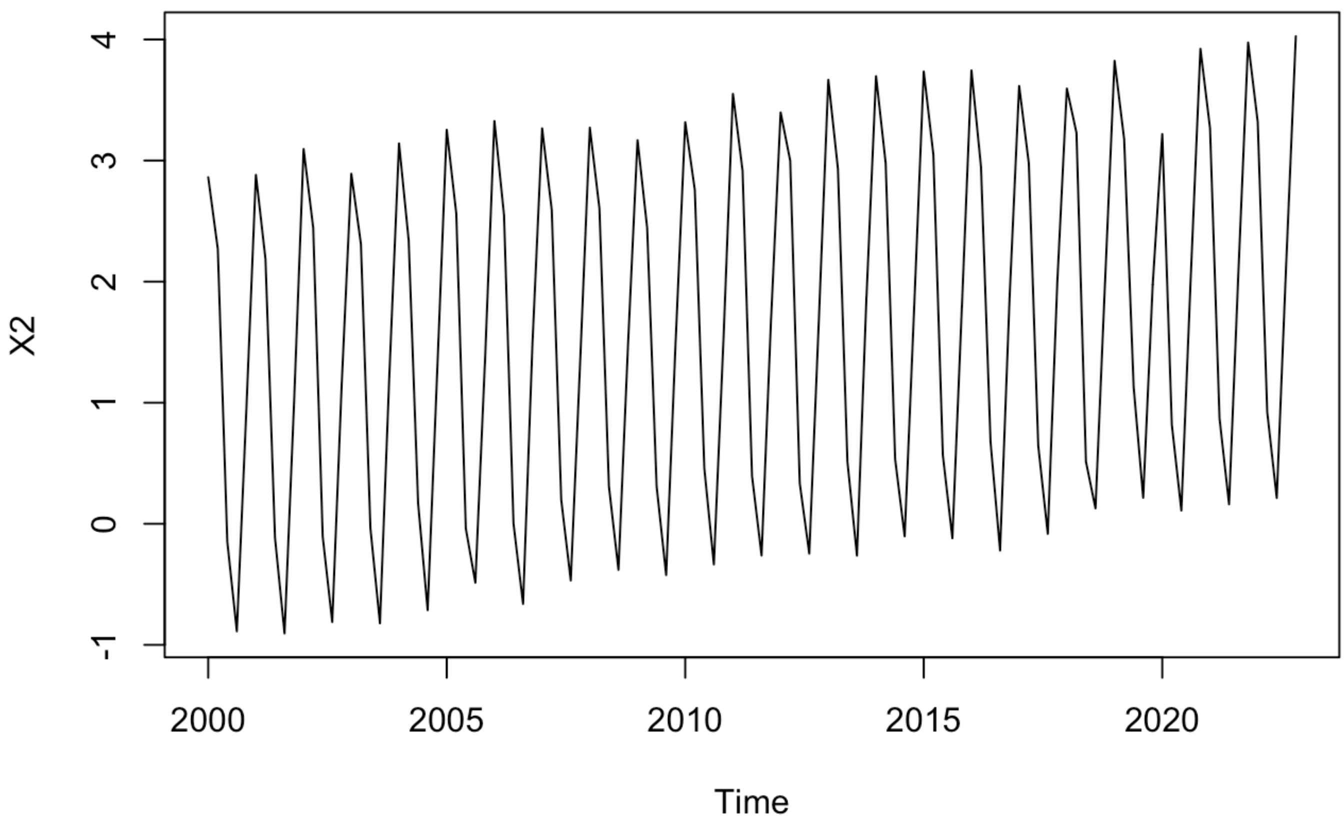
Coefficient reel : 1.903555650 1.156133321 -1.155990654 -1.895566966 -0.008131351 trouvé 1.902113 1.175571 -1.175571 -1.902113 -4.898587e-16

```
for (i in 1 :5) {
  print(2*sin((2*pi*i)/5))
}
```

```
## [1] 1.902113
## [1] 1.175571
## [1] -1.175571
## [1] -1.902113
## [1] -4.898587e-16
```

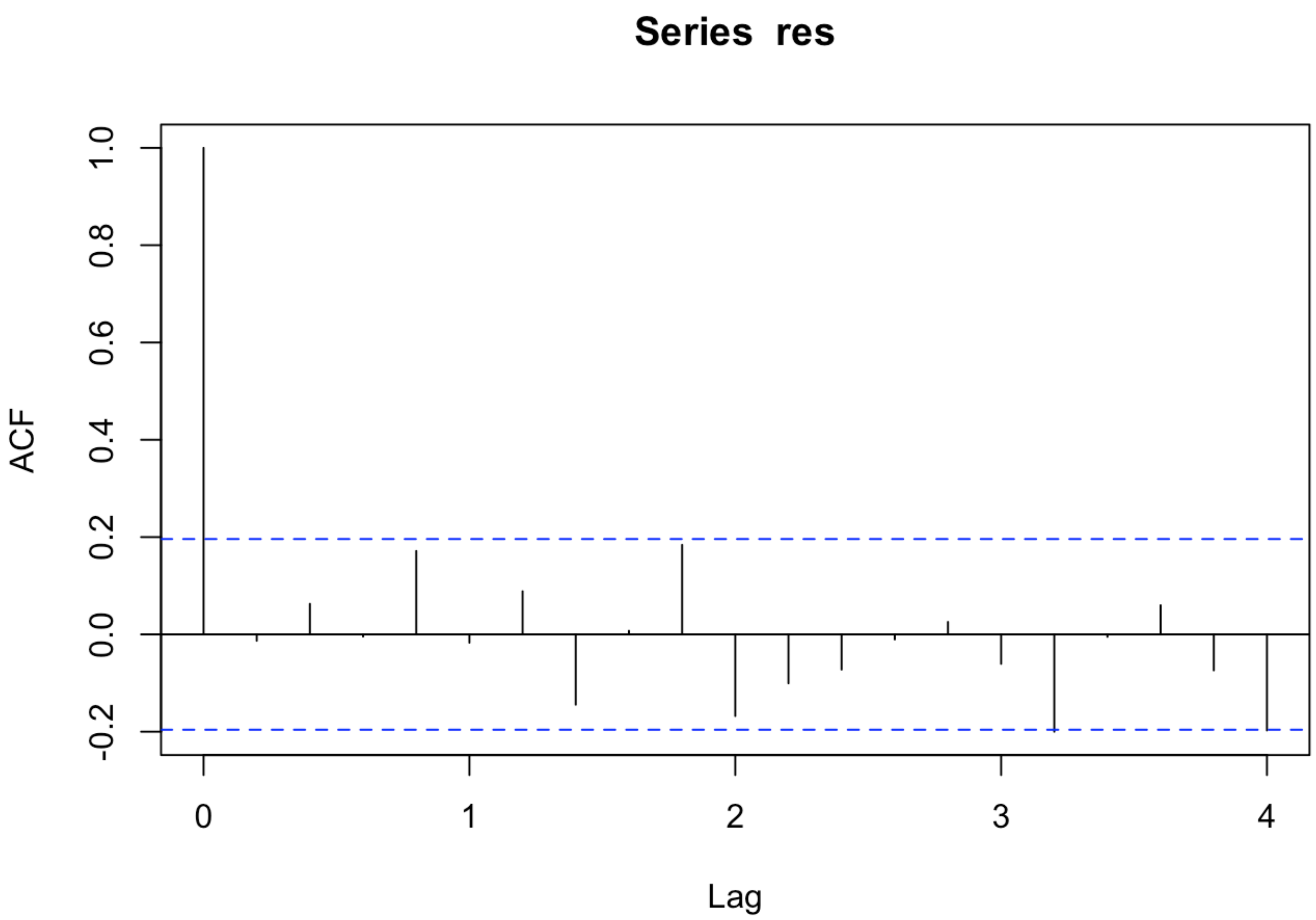
Proposer une prévision à l'horizon 3.

```
X1=rep(1,15)
for (i in 1:15)
  {X1[i]=-102.44482083+0.05170816*(2020+5%%1+(i-1)/5) +s[i%%5+1]}

X2=c(as.vector(data),X1)

X2=as.ts(X2)
X2=ts(X2,start=2000,frequency=5)
plot(X2)
```

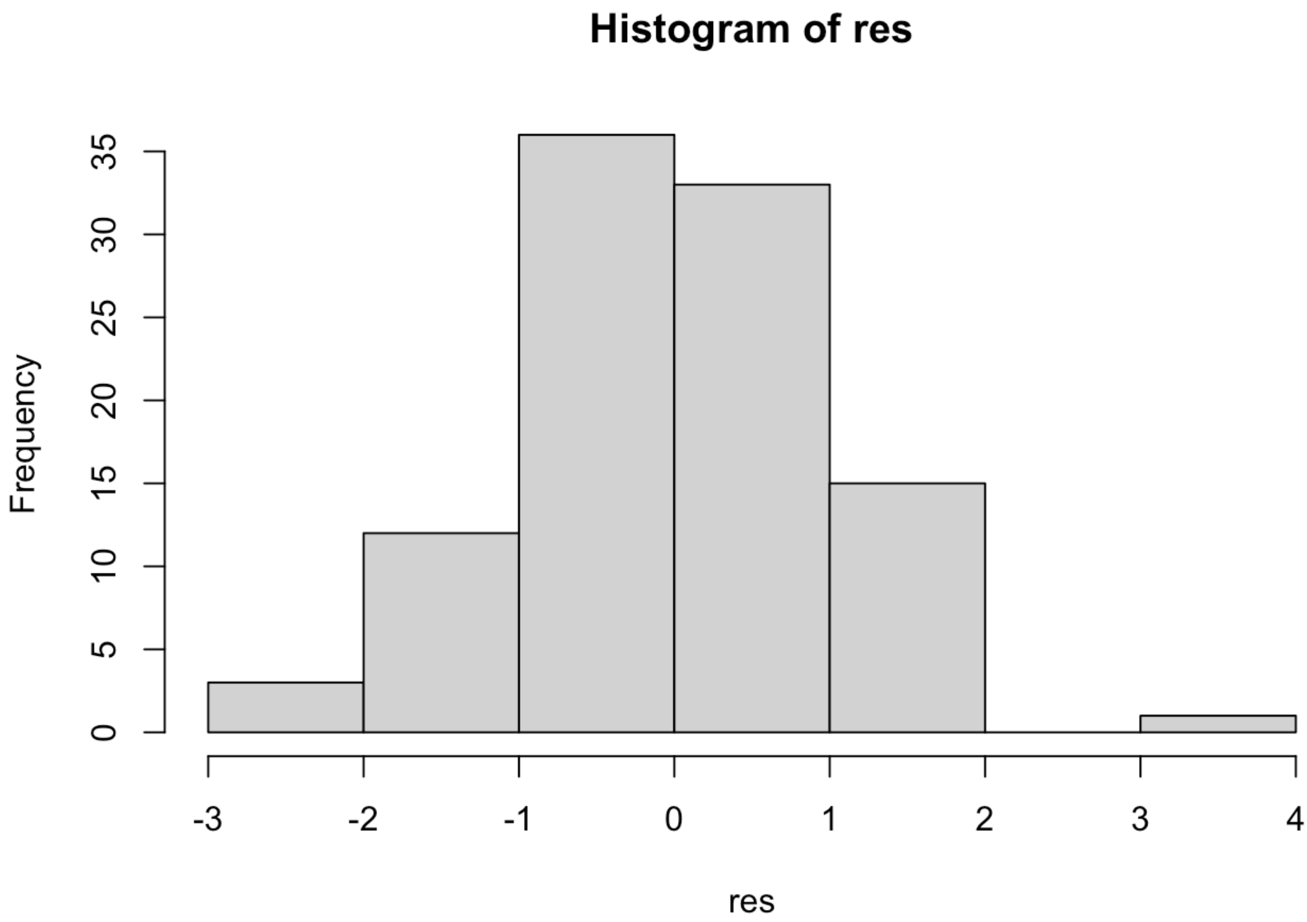


Analyser les résidus.Les représenter.

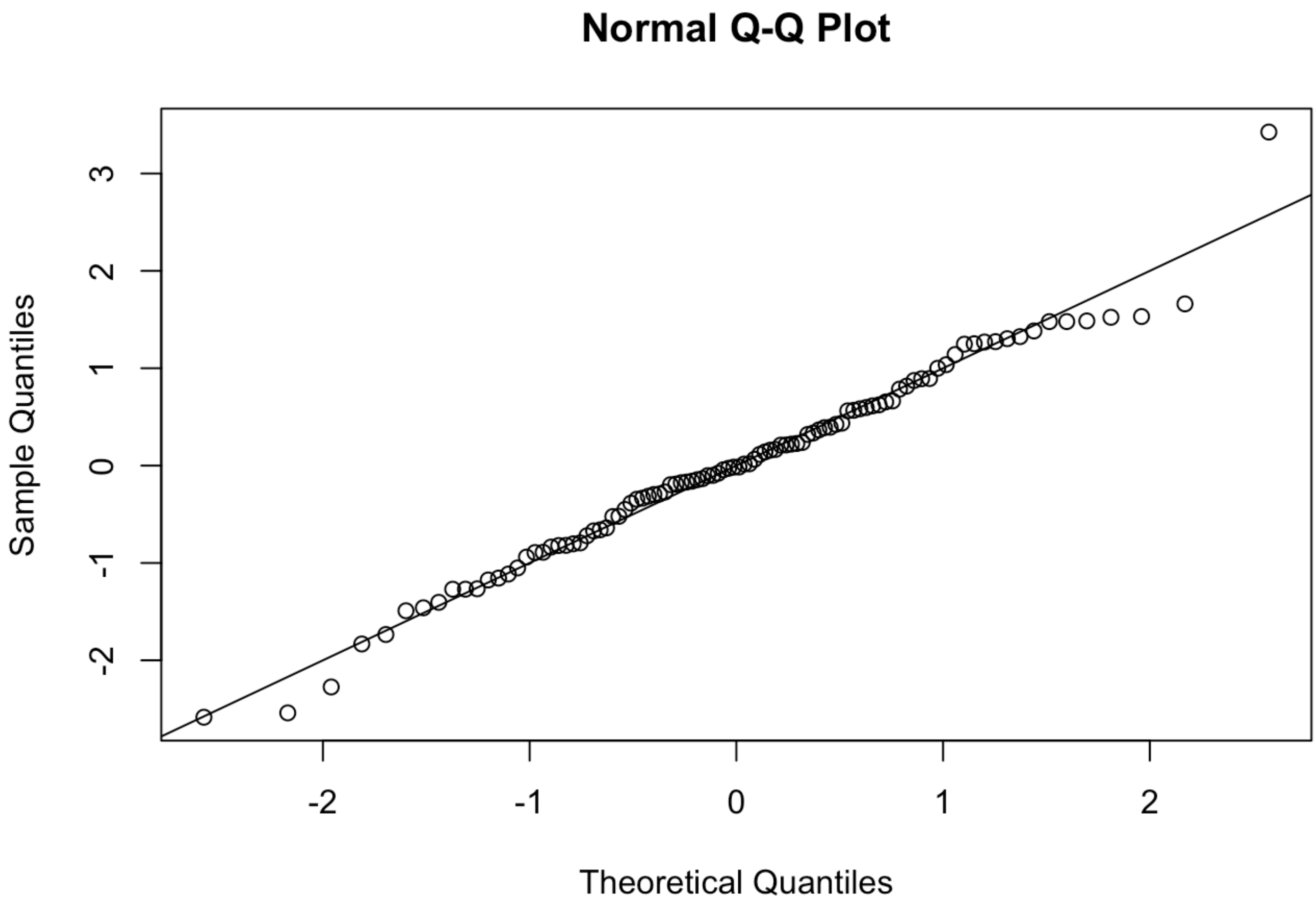
```
res=CVS-CVS.lm$fitted.values
res=res/sqrt(var(res))
acf(res)
```




```
hist(res)
```

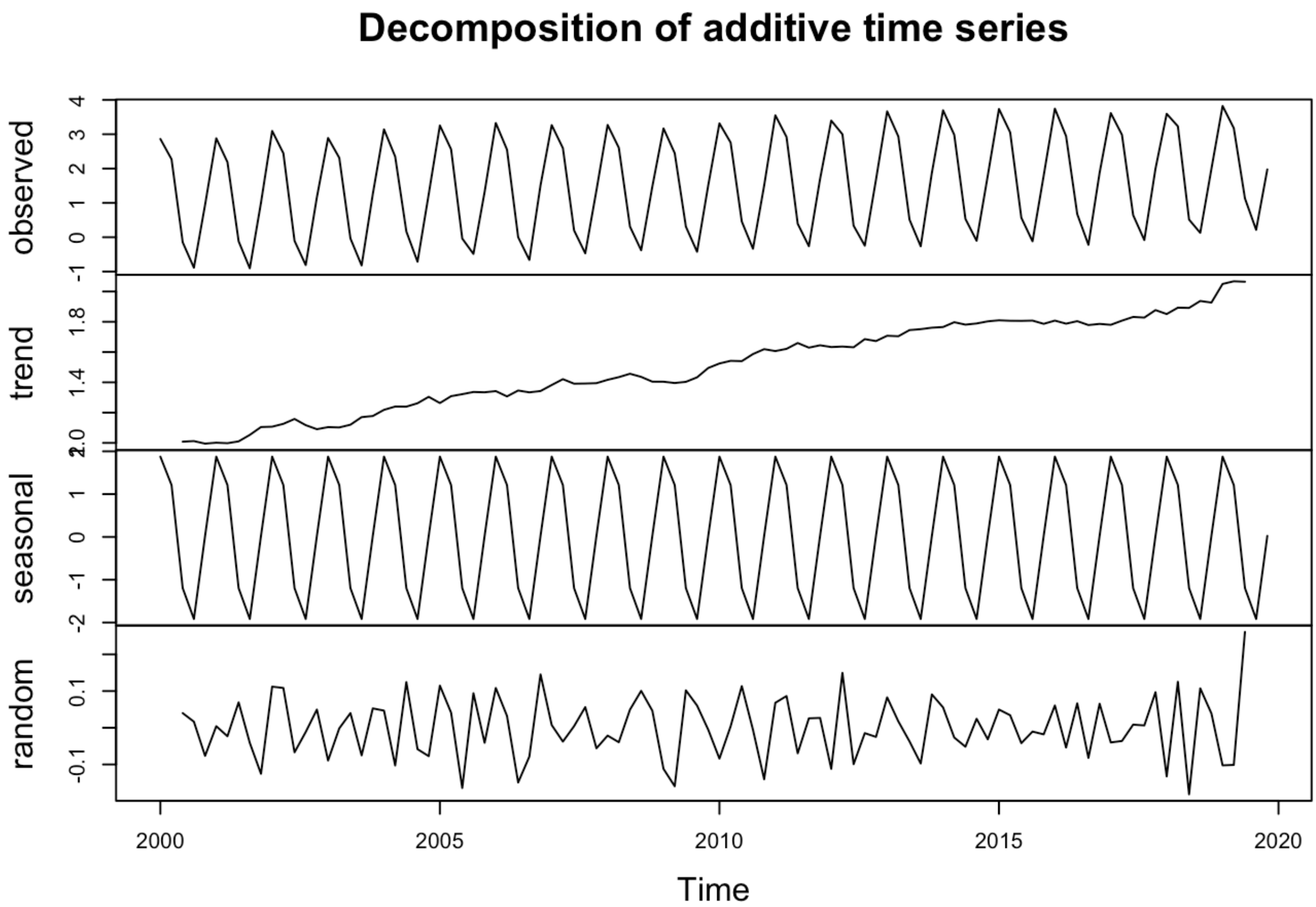



```
qqnorm(res)
abline(0,1)
```



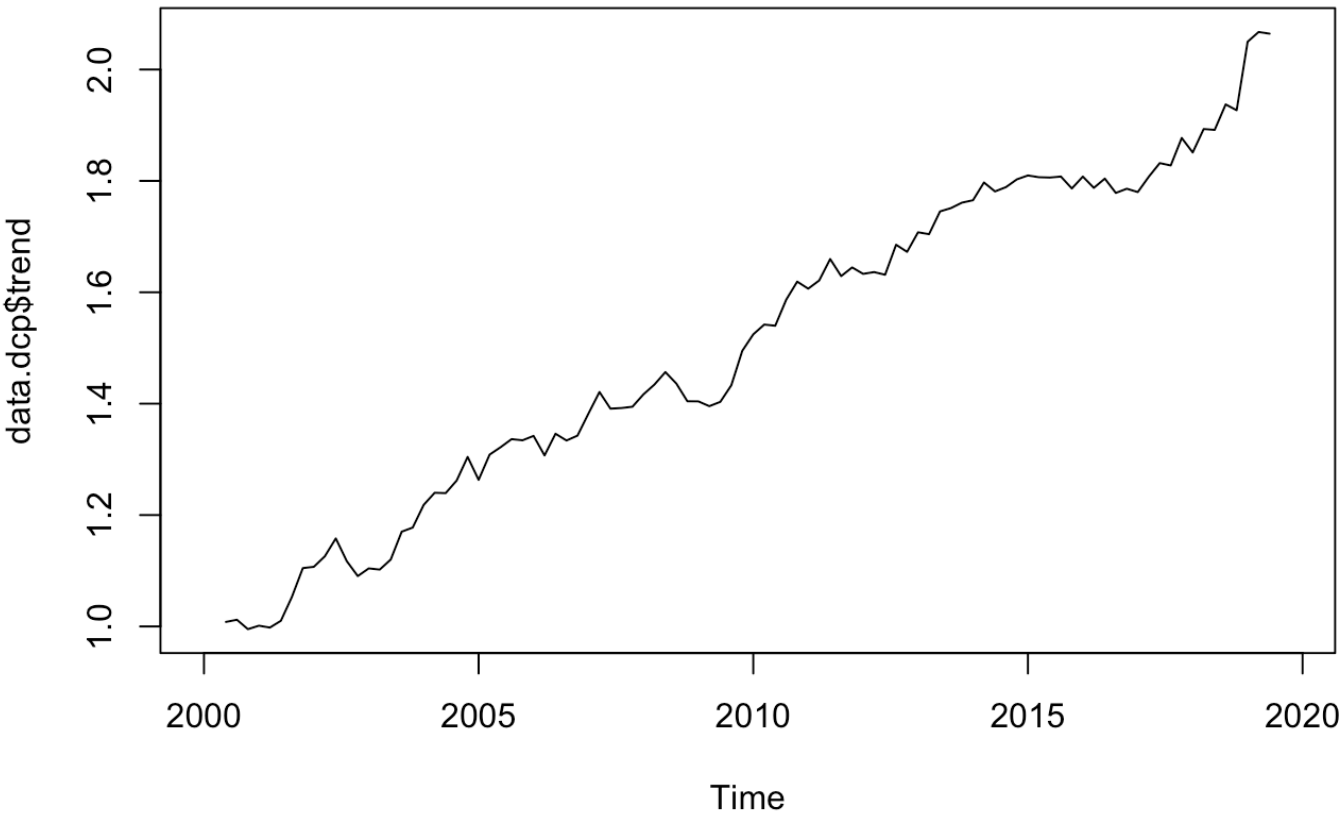
Appliquer la fonction decompose et comparer avec les vraies valeurs.

```
data.dcp= decompose(data,type="add")
plot(data.dcp)
```

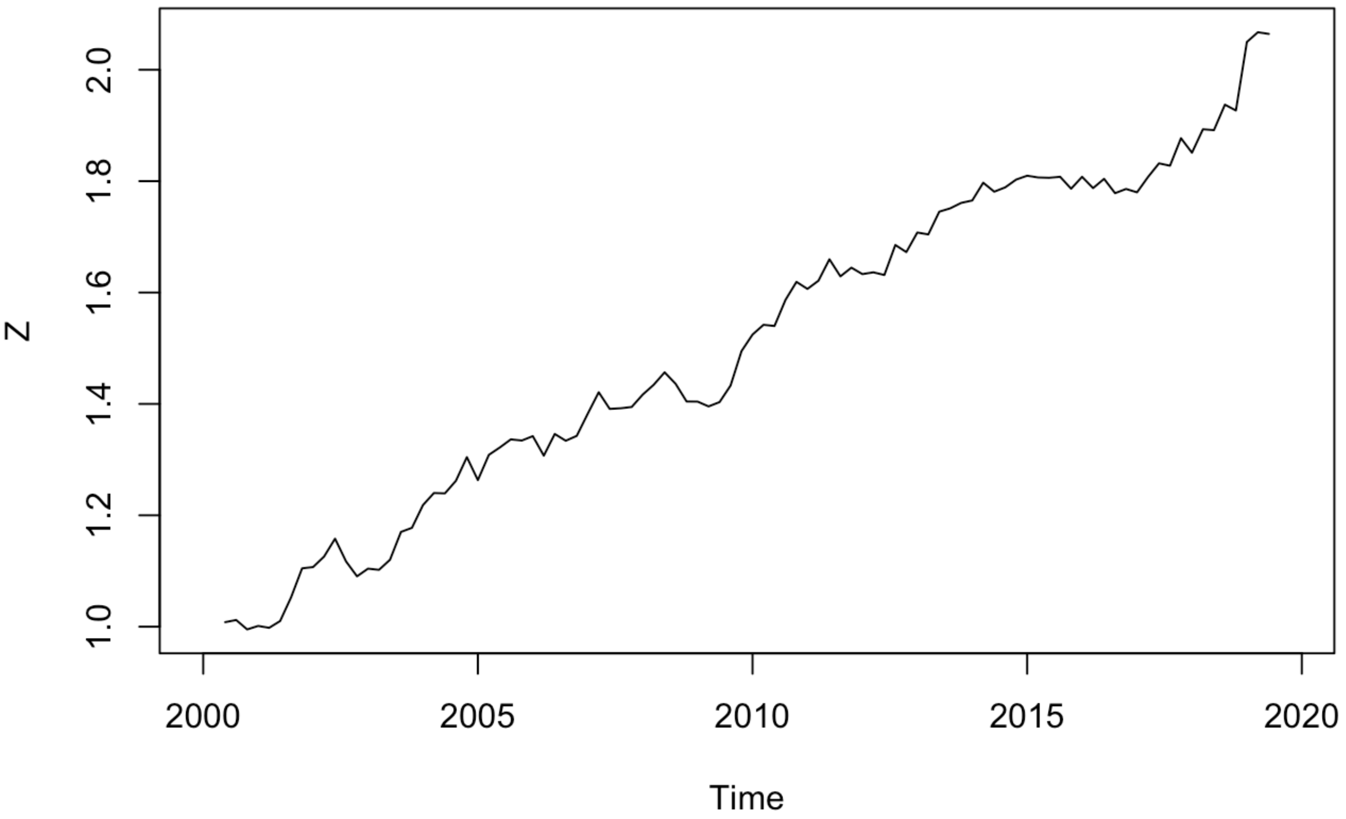


comparaison de la tendance decompose versus reel Nous pouvons remarqué que les valeurs réelle sont pratiquement identique aux valeurs estimée de decompose

```
plot(data.dcp$trend)
```




```
plot(Z)
```

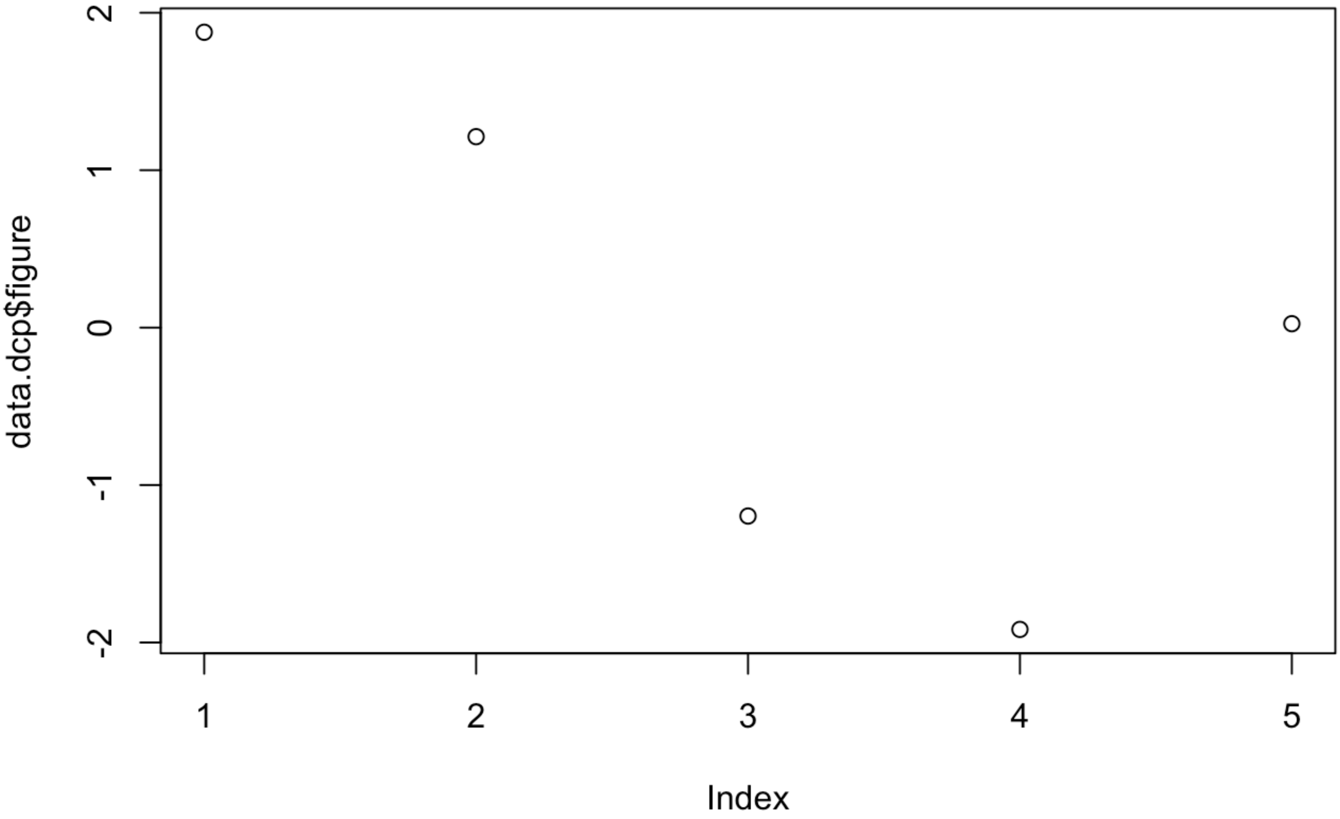


comparaison de la saisonalité decompose versus reel Nous pouvons remarqué que les valeurs réelle sont identique identique aux valeurs estimée de decompose

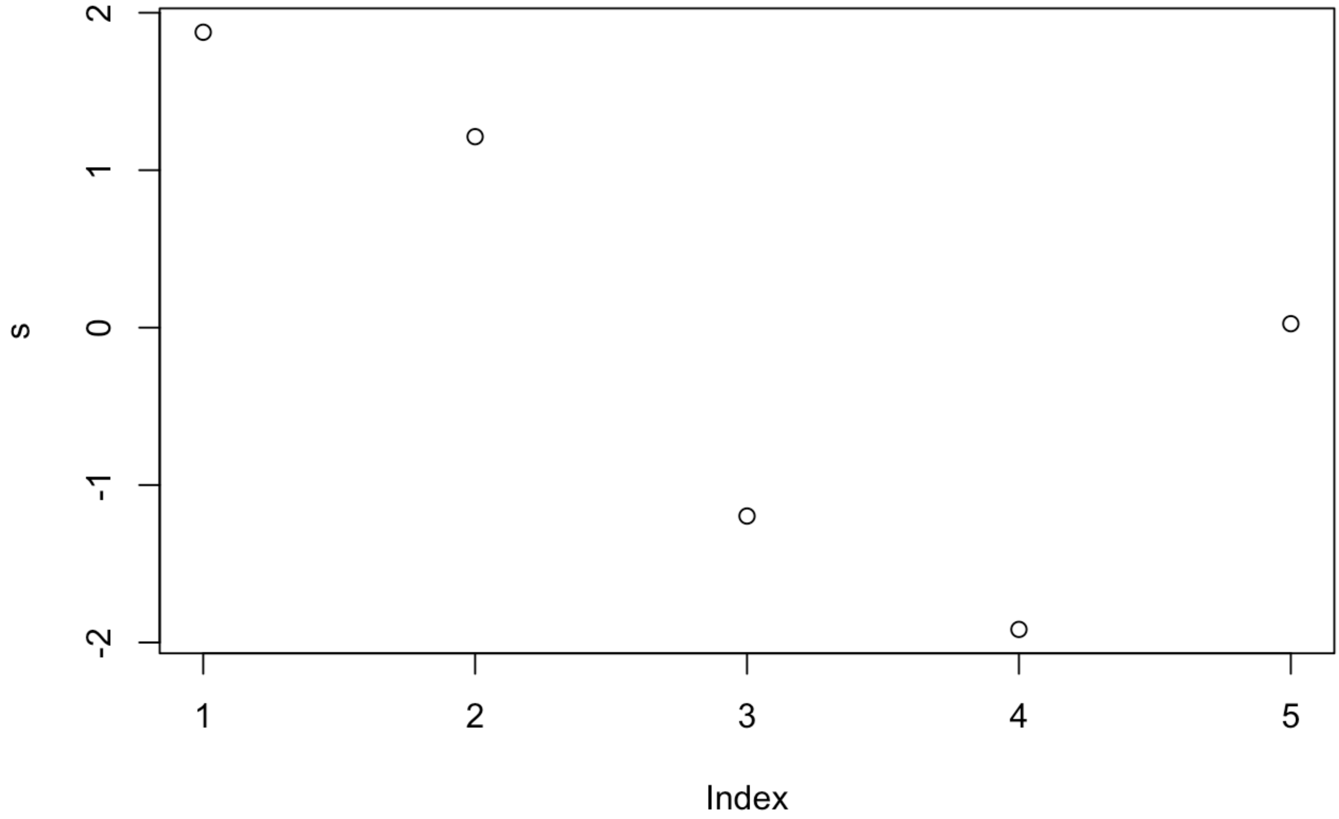
```
if(sum(data.dcp$figure==s)==length(s)){
  print("les valeurs sont identique")
}
```

```
## [1] "les valeurs sont identique"
```

```
plot(data.dcp$figure)
```




```
plot(s)
```


s
Index
2
1
0
-1
-2
1
2
3
4
5

Xdata
Time
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
1990.0
1990.5
1991.0
1991.5
1992.0
1992.5
1993.0