Interpolation et lissage

- (x_i, y_i), $i = 1, \ldots, n$
  Toute fonction g telle que $y_i = g(x_i)$, $i = 1, \ldots, n$,
  s'appelle une interpolation.

Interpolation et lissage

- (x_i, y_i), $i = 1, \ldots, n$
  Toute fonction g telle que $y_i = g(x_i)$, $i = 1, \ldots, n$,
  s'appelle une interpolation.
- L'interpolation reflète toutes les variations, y compris les fluctuations aléatoires et les erreurs (bruit).
  Généralement trop irrégulière pour satisfaire un statisticien.

Interpolation et lissage

- (x_i, y_i), $i = 1, \ldots, n$
  Toute fonction g telle que $y_i = g(x_i)$, $i = 1, \ldots, n$,
  s'appelle une interpolation.
- L'interpolation reflète toutes les variations, y compris les fluctuations aléatoires et les erreurs (bruit).
  Généralement trop irrégulière pour satisfaire un statisticien.
- En régression non paramétrique, on fait du lissage (angl. *smoothing*).

Interpolation et lissage

- (x_i, y_i), $i = 1, \ldots, n$
  Toute fonction g telle que $y_i = g(x_i)$, $i = 1, \ldots, n$,
  s'appelle une interpolation.
- L'interpolation reflète toutes les variations, y compris les fluctuations aléatoires et les erreurs (bruit).
  Généralement trop irrégulière pour satisfaire un statisticien.
- En régression non paramétrique, on fait du lissage (angl. *smoothing*).
- Le lissage vise à estimer une fonction de régression m qui lierait parfaitement y à x s'il n'y avait pas de bruit.
  En général, un lissage n'est pas une interpolation.

Interpolation de Lagrange

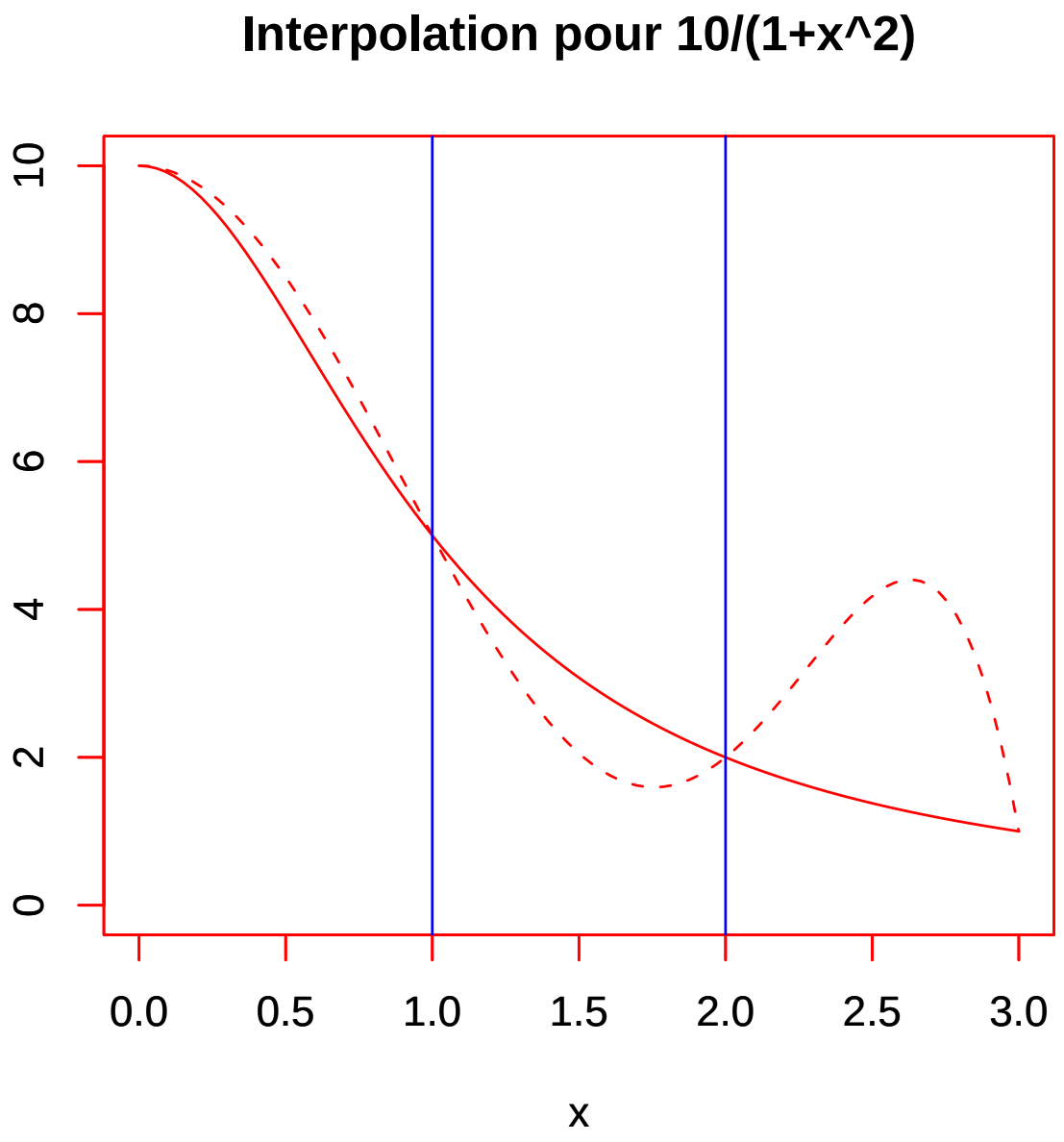
- Points sur la courbe $x \mapsto 10/(1+x^2)$, $0 \leq x \leq 3$

```
> x = seq(0,3,length=100)
> plot(x,10/(1+xˆ2),type=“l”„xlim=c(0,3),ylim=c(0,10),
ylab=”)
> par(new=T)# première superposition
```

Interpolation de Lagrange

- Points sur la courbe $x \mapsto 10/(1+x^2)$, $0 \leq x \leq 3$
  > x = seq(0,3,length=100)
  > plot(x,10/(1+xˆ2),type="l"„xlim=c(0,3),ylim=c(0,10), ylab=")
  > par(new=T)# première superposition
- Polynôme de Lagrange de degré 6 interpolant entre les points $(0,10),(1,5),(2,2),(3,1)$
  > plot(x,10-6.4*xˆ2+1.5*xˆ4-0.1*xˆ6,type="l",xlim=c(0,3),ylim=c(0,10),ylab=" ",lty=2, main="Interpolation pour 10/(1+xˆ2)")
  > abline(v=1)
  > abline(v=2)
  > par(new=T)# deuxième superposition

Interpolation de Lagrange (suite)



Spline cubique d'interpolation

- Spline cubique interpolant entre $(0,10),(1,5),(2,2),(3,1)$

```
> plot(x,(5-5.65385*(x-1)+3.69231*(x-1)^2
+4.34615*(x-1)^3)*(0<=x)*(x<1)
+ (2-1.38462*(x-2)+0.57692*(x-2)^2
-1.03846*(x-2)^3)*(1<=x)*(x<2)
+ (1-0.80769*(x-3)-0.19231*(x-3)^3)
*(2<=x)*(x<3),type="l",xlim=c(0,3),
ylim=c(0,10),ylab="  ",col="black")
```

Spline cubique d'interpolation

- Spline cubique interpolant entre $(0, 10), (1, 5), (2, 2), (3, 1)$

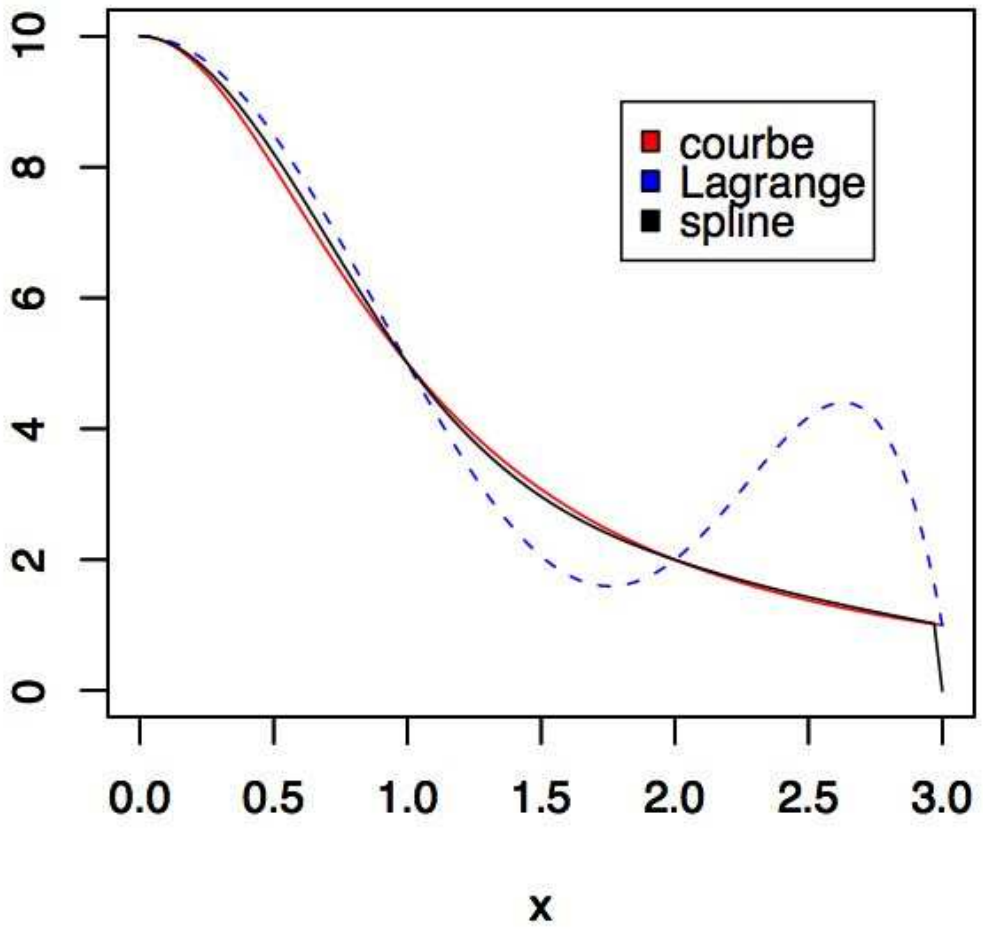
```
> plot(x,(5-5.65385*(x-1)+3.69231*(x-1)^2
+4.34615*(x-1)^3)*(0<=x)*(x<1)
+ (2-1.38462*(x-2)+0.57692*(x-2)^2
-1.03846*(x-2)^3)*(1<=x)*(x<2)
+ (1-0.80769*(x-3)-0.19231*(x-3)^3)
*(2<=x)*(x<3),type="l",xlim=c(0,3),
ylim=c(0,10),ylab="  ",col="black")
```

- Nous verrons plus loin comment obtenir ce spline avec la fonction `interpSpline{splines}`.

Spline cubique d'interpolation

- Spline cubique interpolant entre $(0,10),(1,5),(2,2),(3,1)$
  > plot(x,(5-5.65385*(x-1)+3.69231*(x-1)^2
  +4.34615*(x-1)^3)*(0<=x)*(x<1)
  + (2-1.38462*(x-2)+0.57692*(x-2)^2
  -1.03846*(x-2)^3)*(1<=x)*(x<2)
  + (1-0.80769*(x-3)-0.19231*(x-3)^3)
  (2<=x)(x<3),type="l",xlim=c(0,3),
  ylim=c(0,10),ylab=" ",col="black")
- Nous verrons plus loin comment obtenir ce spline avec la fonction `interpSpline{splines}`.
- > legend(1.8,9,c("courbe","Lagrange","spline"),
  col=c("red","blue","black")
  # légende

Deux interpolations comparées



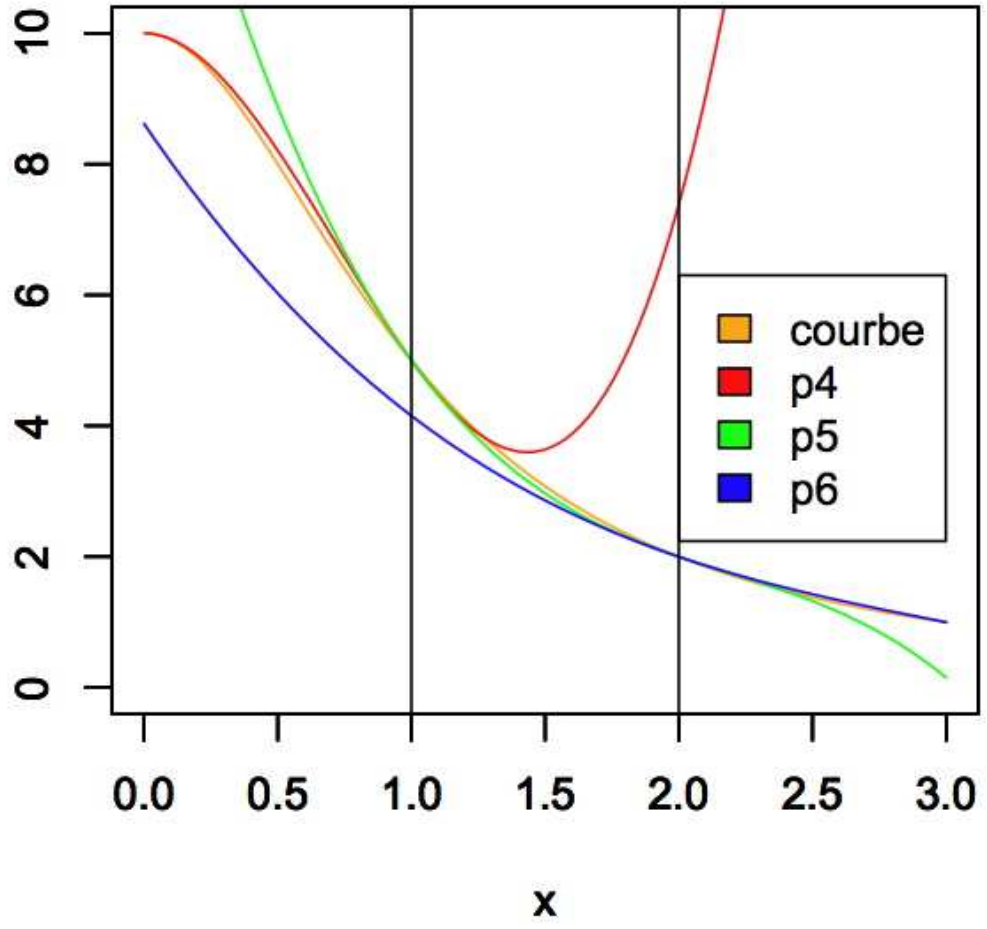
Trois polynômes formant spline

```
> plot(x,10/(1+x^2),type='l',xlim=c(0,3),
ylim=c(0,10),ylab='',col='orange')
```
$y = \frac{10}{1+x^2}, 0 \leq x \leq 3$

```
> par(new=T) #polynôme cubique p4
> plot(x,5-5.65385*(x-1)+3.69231*(x-1)^2+4.34615*(x-
1)^3,type='l',xlim=c(0,3),ylim=c(0,10),ylab='',col='red')
> par(new=T) #polynôme cubique p5
> plot(x,2-1.38462*(x-2)+0.57692*(x-2)^2-1.03846*(x-
2)^3,type='l',xlim=c(0,3),ylim=c(0,10),ylab='',col='green')
> par(new=T) #polynôme cubique p6
> plot(x,1-0.80769*(x-3)-0.19231*(x-
3)^3,type='l',xlim=c(0,3),ylim=c(0,10),ylab='',col='blue')

> legend(2,6.3,c('courbe','p4','p5','p6'),
c('orange','red','green','blue'))
```

Trois polynômes formant spline



Régression spline

- Rappel : on cherche à minimiser en g

$$L_\lambda(g) = \sum_1^n (y_i - g(x_i))^2 + \lambda \int_a^b g''(x)^2 dx,$$

où $\lambda \geq 0$ est le paramètre de lissage,
$a = x_0 < x_1 < x_2 < \cdots < x_n < x_{n+1} = b$ sont les nœuds.

Régression spline

- Rappel : on cherche à minimiser en g

$$L_\lambda(g) = \sum_1^n (y_i - g(x_i))^2 + \lambda \int_a^b g''(x)^2 dx,$$

où $\lambda \geq 0$ est le paramètre de lissage,
$a = x_0 < x_1 < x_2 < \cdots < x_n < x_{n+1} = b$ sont les nœuds.

- Sous certaines conditions sur g (différentiabilité, valeur des dérivées aux nœuds, il existe pour tout λ un minimum unique g_λ.

Régression spline

- Rappel : on cherche à minimiser en g

$$L_\lambda(g) = \sum_1^n (y_i - g(x_i))^2 + \lambda \int_a^b g''(x)^2 dx,$$

où $\lambda \geq 0$ est le paramètre de lissage,
$a = x_0 < x_1 < x_2 < \cdots < x_n < x_{n+1} = b$ sont les nœuds.

- Sous certaines conditions sur g (différentiabilité, valeur des dérivées aux nœuds, il existe pour tout λ un minimum unique g_λ.
- Ce minimum unique g_λ est un spline cubique S_λ.

Régression spline

- Rappel : on cherche à minimiser en g

$$L_\lambda(g) = \sum_1^n (y_i - g(x_i))^2 + \lambda \int_a^b g''(x)^2 dx,$$

où $\lambda \geq 0$ est le paramètre de lissage,
$a = x_0 < x_1 < x_2 < \cdots < x_n < x_{n+1} = b$ sont les nœuds.

- Sous certaines conditions sur g (différentiabilité, valeur des dérivées aux nœuds, il existe pour tout λ un minimum unique g_λ.
- Ce minimum unique g_λ est un spline cubique S_λ.
- On obtient le λ optimal en minimisant la fonction de validation croisée.

Régression spline

- Données `sulfate.dat` : Exercices #7
  pluies acides ($n = 3321$)
  V1: distance entre stations de mesure en km
  V2: corrélation entre les dépôts de sulfate

Régression spline

- Données `sulfate.dat` : Exercices #7
  pluies acides ($n = 3321$)
  V1: distance entre stations de mesure en km
  V2: corrélation entre les dépôts de sulfate
- > sulfate = read.table(”sulfate.dat”)
  > attach(sulfate)
  > plot(V1,V2,xlab="Distance",ylab="Corrélation",
  main="Dépôt d’acide")

Régression spline

- Données `sulfate.dat` : Exercices #7
  pluies acides ($n = 3321$)
  V1: distance entre stations de mesure en km
  V2: corrélation entre les dépôts de sulfate
- > sulfate = read.table(”sulfate.dat”)
  > attach(sulfate)
  > plot(V1,V2,xlab="Distance",ylab="Corrélation",
  main="Dépôt d’acide")
- Dans R, on trouve:
  1) la fonction `smooth.spline` dans le package `stats`;

  2) 22 fonctions pour la régression spline dans le package `splines`, incluant une fonction pour obtenir un spline d’interpolation.

Régression spline (suite)

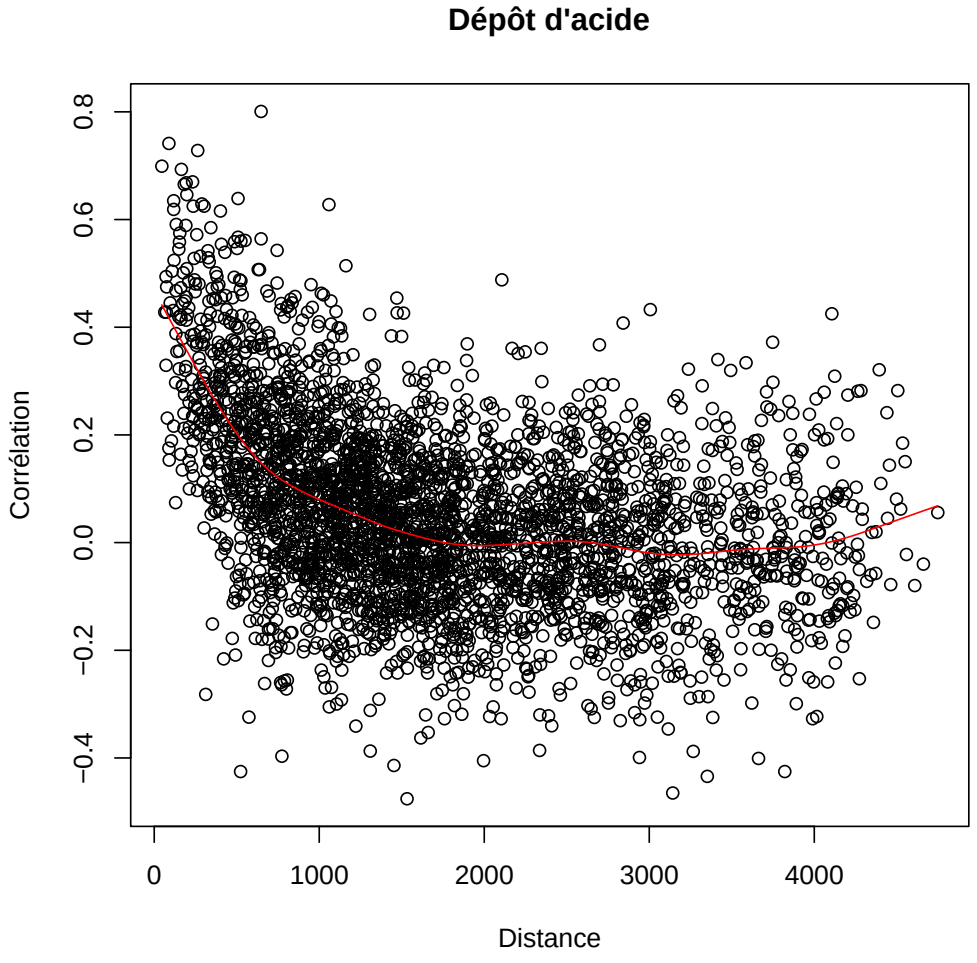
- smooth.spline(x, y = NULL, w = NULL, df, spar = NULL, cv = FALSE, all.knots = FALSE, nknots = NULL, keep.data = TRUE, df.offset = 0, penalty = 1, control.spar = list())
  #x, y: données
  cv = T (validation croisée), = F (validation croisée généralisée par défaut)

Régression spline (suite)

- smooth.spline(x, y = NULL, w = NULL, df, spar = NULL, cv = FALSE, all.knots = FALSE, nknots = NULL, keep.data = TRUE, df.offset = 0, penalty = 1, control.spar = list())
  #x, y: données
  cv = T (validation croisée), = F (validation croisée généralisée par défaut)
- > smooth.spline(V1,V2)
  Smoothing Parameter spar= 0.9942156 lambda= 0.005363714 (11 iterations)
  Equivalent Degrees of Freedom (Df): 10.23377
  Penalized Criterion: 72.60825
  GCV: 0.02207086 #minimum de $VCG(\lambda)$ (validation croisée généralisée par défaut)

Régression spline (suite)

> plot(V1,V2,xlab=“Distance”,ylab=“Correlation”,main=“Depot d’acide”)
> lines(smooth.spline(V1,V2),col=“red”)



Splines d’interpolation

- La fonction `interpSpline` du package `splines` crée un spline cubique d’interpolation.

Splines d’interpolation

- La fonction `interpSpline` du package `splines` crée un spline cubique d’interpolation.
- > x = c(-3, -2, -1, 0, 1, 2, 3)
  > y = 10/(1+x^2)

Splines d'interpolation

- La fonction `interpSpline` du package `splines` crée un spline cubique d'interpolation.
- > x = c(-3, -2, -1, 0, 1, 2, 3)
  > y = 10/(1+x^2)
- Interpolation aux points
  $(-3, 1), (-2, 2), (-1, 5), (0, 10), (1, 5), (2, 2), (3, 1)$

Splines d'interpolation

- La fonction `interpSpline` du package `splines` crée un spline cubique d'interpolation.
- > x = c(-3, -2, -1, 0, 1, 2, 3)
  > y = 10/(1+x^2)
- Interpolation aux points
  $(-3,1),(-2,2),(-1,5),(0,10),(1,5),(2,2),(3,1)$
- > interpSpline(x,y)
  #coefficients des polynômes vus aux notes de cours
  polynomial representation of spline for y~x
  constant linear quadratic cubic
  0 10 0.0000000 -9.3461538 4.3461538
  1 5 -5.6538462 3.6923077 -1.0384615
  2 2 -1.3846154 0.5769231 -0.1923077
  3 1 -0.8076923 0.0000000 0.0000000