Système d'équation

a) Etablir le système d'équation en définissant la matrice du modèle.Pour faire ce travail, il faut analyser la structure du réseau et déterminer les différents états possibles.

```
X=[
 1 0 0 0 0 0
 1 0 0 1 0 0
 1 0 0 0 1 0
 1 0 0 0 0 1
 1 1 0 0 0 0
 1 1 0 1 0 0
 1 1 0 0 1 0
 1 1 0 0 0 1
 1 0 1 0 0 0
 1 0 1 1 0 0
 1 0 1 0 1 0
 1 0 1 0 0 1
]
```

```
X =
     1     0     0     0     0     0
     1     0     0     1     0     0
     1     0     0     0     1     0
     1     0     0     0     0     1
     1     1     0     0     0     0
     1     1     0     1     0     0
     1     1     0     0     1     0
     1     1     0     0     0     1
     1     0     1     0     0     0
     1     0     1     1     0     0
     1     0     1     0     1     0
     1     0     1     0     0     1
```

b) Etablir le système linéaire

$Y = X\alpha + \epsilon$

```
R=sym('R',[6,1]);
e=sym('e',[12,1]);
Y=sym('Y',[12,1]);
Sys=Y==X*R+e
```

```
Sys =
```

$$
\begin{pmatrix}
Y_1 = R_1 + e_1 \\
Y_2 = R_1 + R_4 + e_2 \\
Y_3 = R_1 + R_5 + e_3 \\
Y_4 = R_1 + R_6 + e_4 \\
Y_5 = R_1 + R_2 + e_5 \\
Y_6 = R_1 + R_2 + R_4 + e_6 \\
Y_7 = R_1 + R_2 + R_5 + e_7 \\
Y_8 = R_1 + R_2 + R_6 + e_8 \\
Y_9 = R_1 + R_3 + e_9 \\
Y_{10} = R_1 + R_3 + R_4 + e_{10} \\
Y_{11} = R_1 + R_3 + R_5 + e_{11} \\
Y_{12} = R_1 + R_3 + R_6 + e_{12}
\end{pmatrix}
$$

Cette partie qui utilise le calcul symbolique est un peu alibi. Dans la réalité, la mise en équation a été faite avant d'avoir la matrice du modèle.

c) Calculer le rang, la matrice de dispersion, les VIF

```
Rk1=rank(X);
disp(['Rang : ', num2str(Rk1)])
```

```
Rang : 6
```

```
format rat
D=inv(X.'*X);
disp("Matrice de dispersion")
```

```
Matrice de dispersion
```

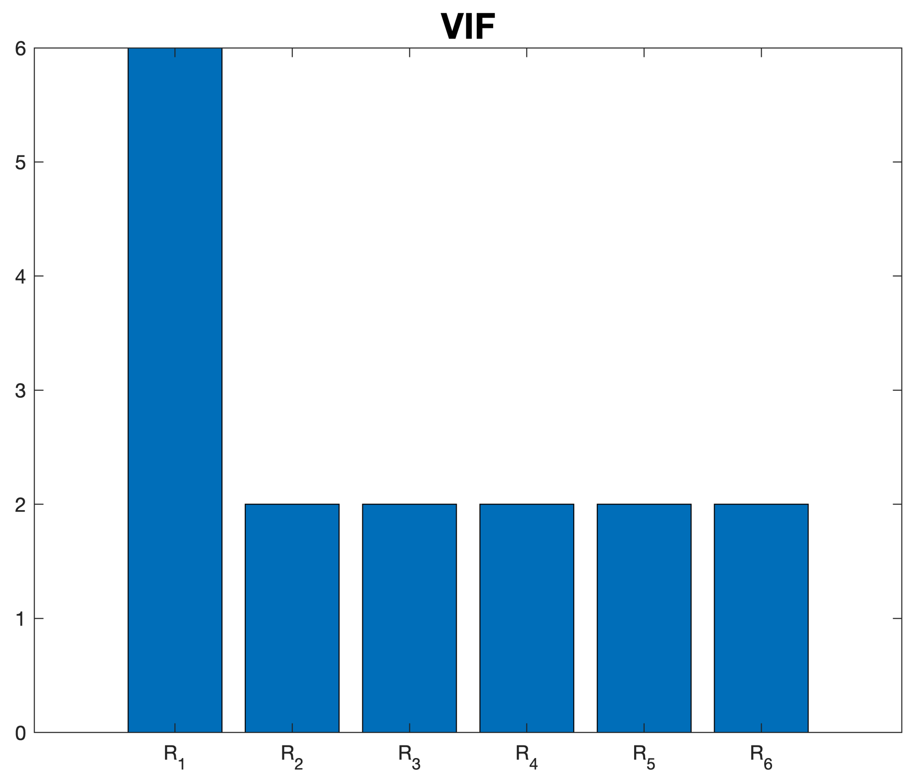
```
disp(num2str(D))
```

```
     0.5       -0.25       -0.25    -0.33333    -0.33333    -0.33333
   -0.25         0.5        0.25 -5.5511e-17 -5.5511e-17 -5.5511e-17
   -0.25        0.25         0.5 -5.5511e-17 -5.5511e-17 -5.5511e-17
-0.33333           0           0     0.66667     0.33333     0.33333
-0.33333           0           0     0.33333     0.66667     0.33333
-0.33333           0           0     0.33333     0.33333     0.66667
```

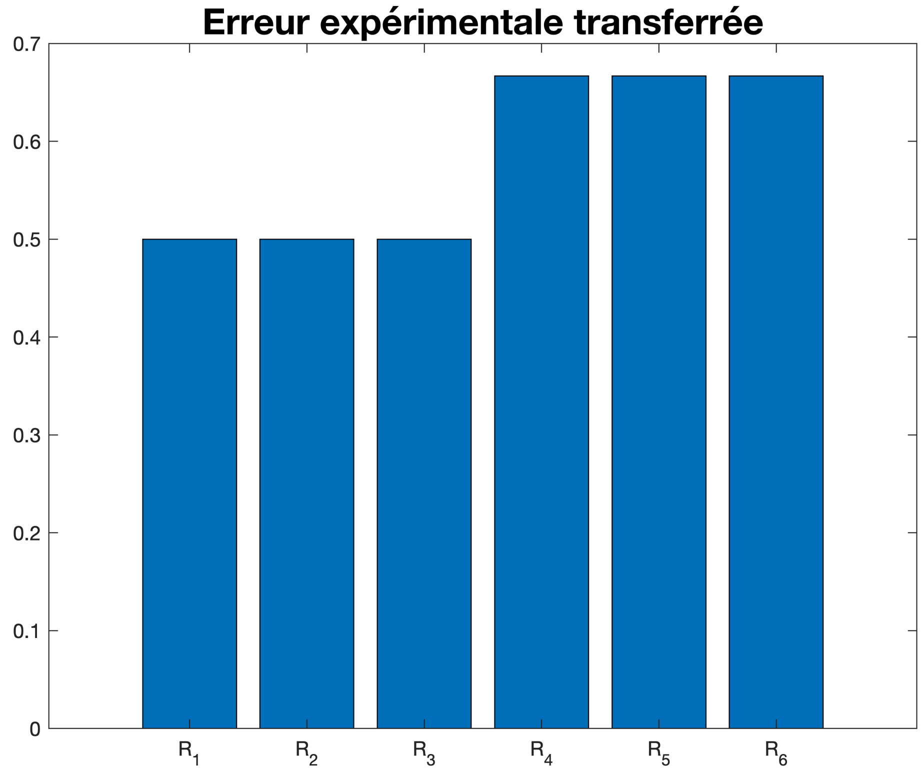
```
C=corrcov(D);
VIF=diag(inv(C))
```

```
VIF =
       6
       2
       2
       2
       2
       2
```

```
bar(VIF)
title('VIF',"FontSize",18)
xticklabels({'R_1' 'R_2' 'R_3' 'R_4' 'R_5' 'R_6'})
```




```
bar(diag(D))
title('Erreur expérimentale transferrée',"FontSize",18)
xticklabels({'R_1' 'R_2' 'R_3' 'R_4' 'R_5' 'R_6'})
```



- Le rang permet de vérifier que le système est résoluble. dans le cas présent, la valeur de 6 confirme que notre système pourra être résolu.
- Les VIF donnent une information sur l'impact d'éventuelles colinéarités entre les facteurs. Ils doivent être inférieurs à 6, et en tout cas à 10 sinon le risque est très important que les résultats soient entachés de trop d'incertitude pour être utilisable. Dans le cas présent , mis à part a_o, la collinéarité a une faible influence sur l'incertitude des résultats
- Les éléments diagonaux de la matrice de dispertion sont aussi des éléments qui permettent de prédire avant les expériences le niveau d'incertitude des résultats en rapport avec l'incertitude de la mesure, qui elle n'est pas connue avec certitude avant de réaliser les essais. Dans le cas présent les valeurs de 1/2 et 2/3 indique un plan qui n'est pas optimal, mais qui est réalisable.

d) `Diminuer le système d'équation au minimum (pour qu'il reste résoluble) et recalculer la matrice de dispersion ainsi que les facteurs d'inflation de la variance. Comparer avec le cas précédent.`

```
nchoosek(12,6)
```

```
ans =
   924
```

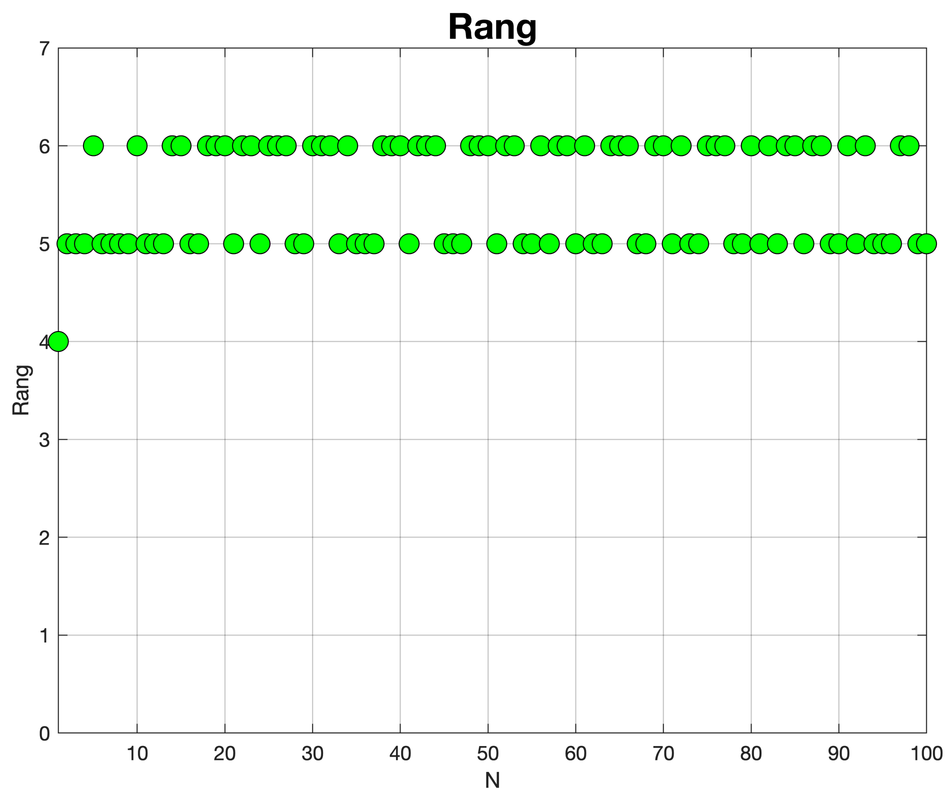
Le nombre d'équations initiales, 12, permet d'imaginer qu'il soit possible de rencontrer un set de 6 béquations qui permettraient de résoudre notre problème en moins d'expériences. Il y a $N = C_6^{12} = \frac{12!}{6!\ 6!} = 924$ possibilités.

Pour ne pas les tester toutes, on va en tester 100 au hasard.

```
NN=100; % nombre de test

% initialisation des matrices
% (Recommendé pour minimiser le temps de calcul)
P=zeros(NN,6);
Rk1=ones(NN,1)*nan;
T1=ones(NN,1)*nan;


for i=1:NN
    P(i,1:6)=randperm(12,6);

    X1=X(P(i,1:6),:);
    Rk1(i)=rank(X1);

    if Rk1(i)==6
        D1=inv(X1'*X1); % matrice de dospèersion
        T1(i)=trace(D1); % trace
    end
end
```

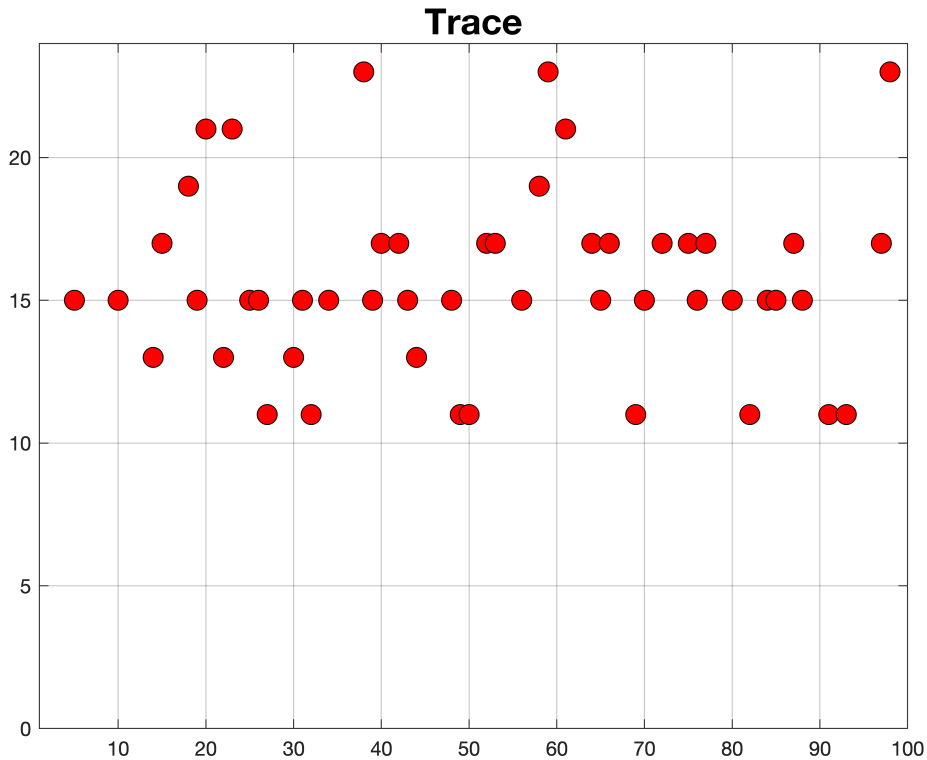
Graphe du rang

```
plot(Rk1,'ok',"MarkerSize",10,"MarkerFaceColor","green")
axis([1,NN,0,7])
grid on
title("Rang","FontSize",18)
xlabel("N")
ylabel("Rang")
```



Graphe de la trace

```
plot(T1,'ok',"MarkerSize",10,"MarkerFaceColor","red")
axis([1,NN,0,max(T1)+1])
grid on
title("Trace","FontSize",18)
```



Pour les cas correspondants à un rank suffisant, on a calculé la matrice de dispersion, $D = (X'X)^{-1}$ et sa trace. Les cas intéressants sont ceux avec la trace minimale puisqu'ils correspondent aux cas qui donneront les intervalles de confiance minimaux.

Cas avec la trace minimale

```
[T_min,indmin]=min(T1);

disp(['La tace minimale est de ' num2str(T_min)])
```

```
La tace minimale est de 11
```

```
disp('Les expériences correspondantes sont:')
```

```
Les expériences correspondantes sont:
```

```
disp(P(indmin,:))
```

```
     9     2    11     5     4     1
```

```
Xmin=X(P(indmin,:),:);

% teste si le rang est bien de 6
```

```
if(rank(Xmin)~=6)
    error('Trace minimale: problème avec le rang')
end

disp("Un des plans donnants une trace minimale")
```

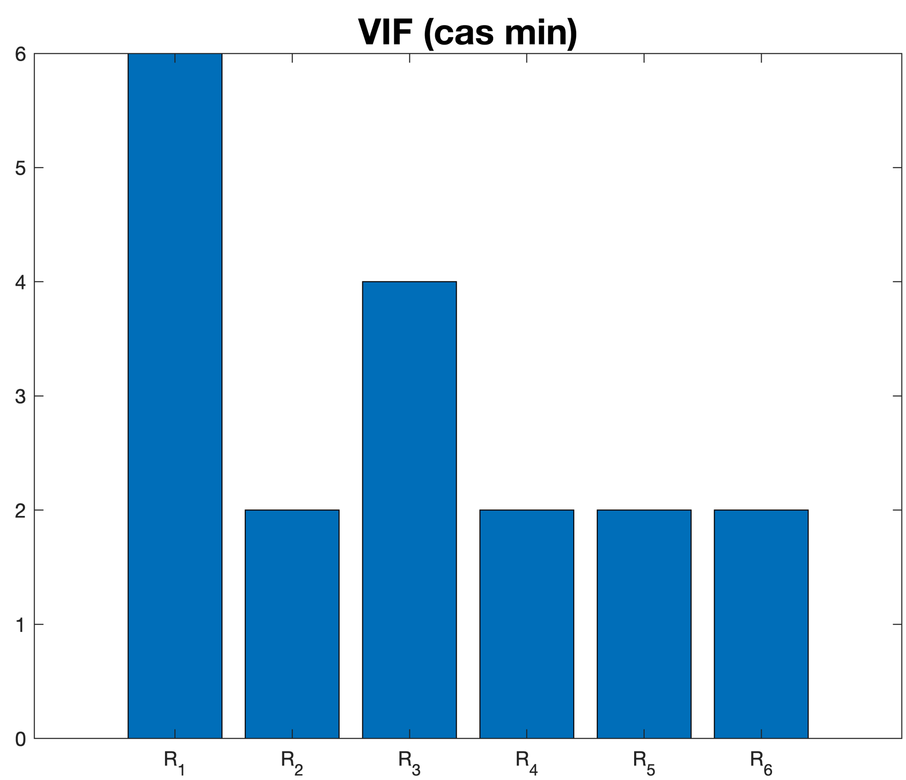
```
Un des plans donnants une trace minimale
```

```
disp(Xmin)
```

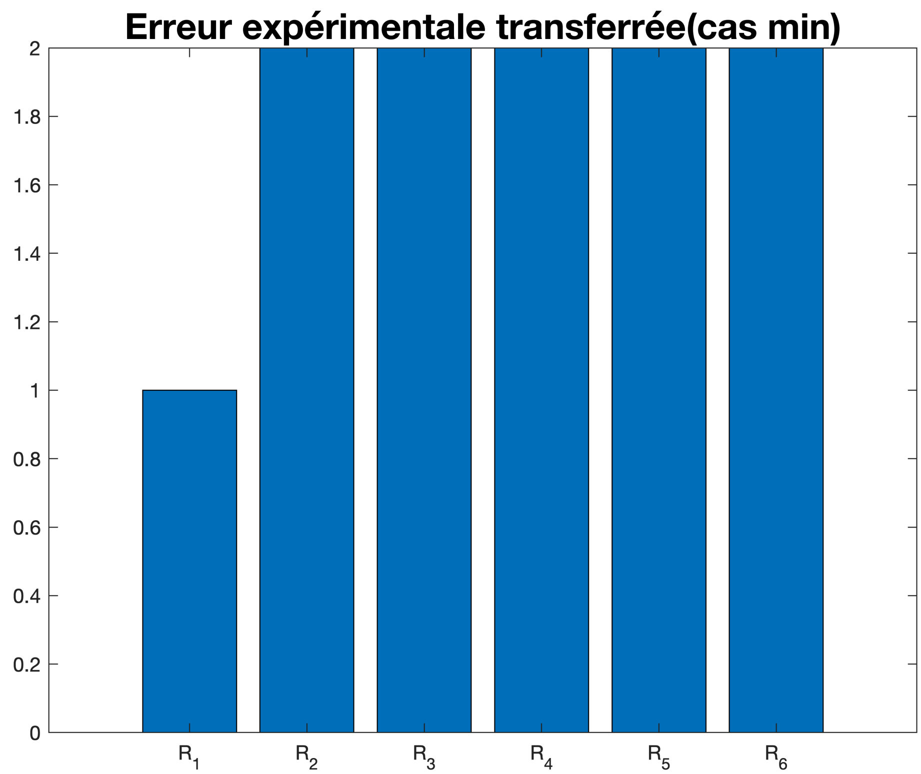
```
     1     0     1     0     0     0
     1     0     0     1     0     0
     1     0     1     0     1     0
     1     1     0     0     0     0
     1     0     0     0     0     1
     1     0     0     0     0     0
```

```
disp("  ")
```

```
Dmin=inv(Xmin'*Xmin);
VIF_min=diag(inv(corrcov(Dmin)));

bar(VIF_min)
title('VIF (cas min)',"FontSize",18)
xticklabels({'R_1' 'R_2' 'R_3' 'R_4' 'R_5' 'R_6'})
```



```
bar(diag(Dmin))
title('Erreur expérimentale transferrée(cas min)',"FontSize",18)
xticklabels({'R_1' 'R_2' 'R_3' 'R_4' 'R_5' 'R_6'})
```



On peut observer dans ce plan optimal à 6 expériences que les VIF et les diagonaux de la matrice de dispersion se sont péjorés en comparaison de la situations avec 12 expériences.

Cas avec la trace maximale

```
[T_max,indmax]=max(T1)
```

```
T_max =
      23
indmax =
      38
```

```
disp(P(indmax,:))
```

```
      12           9           8           2           7           3
```

```
Xmax=X(P(indmax,:),:);
rank(Xmax)
```

```
ans =
       6
```

```
disp("Matrice donnant la trace maximale")
```

Matrice donnant la trace maximale

```
disp(Xmax)
```

```
     1     0     1     0     0     1
     1     0     1     0     0     0
     1     1     0     0     0     1
     1     0     0     1     0     0
     1     1     0     0     1     0
     1     0     0     0     1     0
```

```
Dmax=inv(Xmax'*Xmax)
```

```
Dmax =
     5    -2    -4    -5    -4    -2
    -2     2     2     2     1     *
    -4     2     4     4     3     1
    -5     2     4     6     4     2
    -4     1     3     4     4     2
    -2     *     1     2     2     2
```

```
VIF_max=diag(inv(corrcov(Dmax)))
```

```
VIF_max =
    30
     4
     8
     6
     8
     4
```

On peut observer dans ce plan non-optimal à 6 expériences que les VIF et les diagonaux de la matrice de dispersion se sont péjorés en comparaison de la situations avec 12 expériences, encore plus que dans le cas optimal bien évidemment.

e) Faire l'hypothèse que $R_1 = 0$ et re-calculer la matrice de dispersion ainsi que les facteurs d'inflation de la variance. Comparer avec les cas précédents.

```
X2=X(:,2:6)
```

```
X2 =
     0     0     0     0     0
     0     0     1     0     0
     0     0     0     1     0
     0     0     0     0     1
     1     0     0     0     0
     1     0     1     0     0
     1     0     0     1     0
     1     0     0     0     1
     0     1     0     0     0
     0     1     1     0     0
     0     1     0     1     0
     0     1     0     0     1
```

```
Rk1=rank(X2);
disp(['Rang : ', num2str(Rk1)])
```

```
Rang : 5
```

```
format rat
D=inv(X2.'*X2);
disp("Matrice de dispersion")
```

```
Matrice de dispersion
```

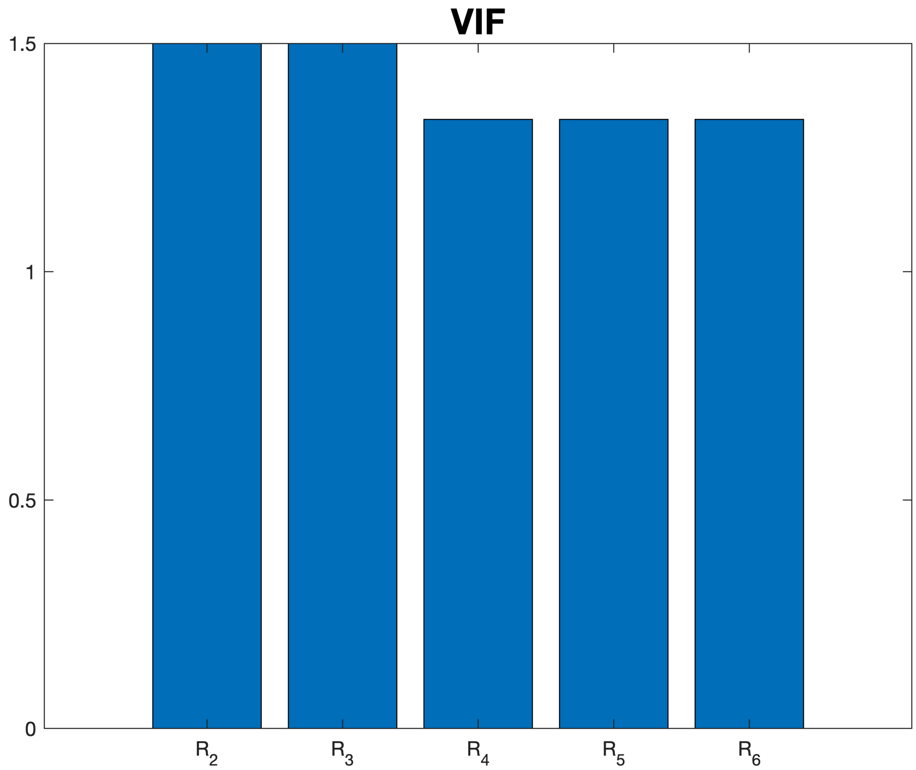
```
disp(num2str(D))
```

```
   0.375       0.125    -0.16667    -0.16667    -0.16667
   0.125       0.375    -0.16667    -0.16667    -0.16667
-0.16667    -0.16667     0.44444     0.11111     0.11111
-0.16667    -0.16667     0.11111     0.44444     0.11111
-0.16667    -0.16667     0.11111     0.11111     0.44444
```

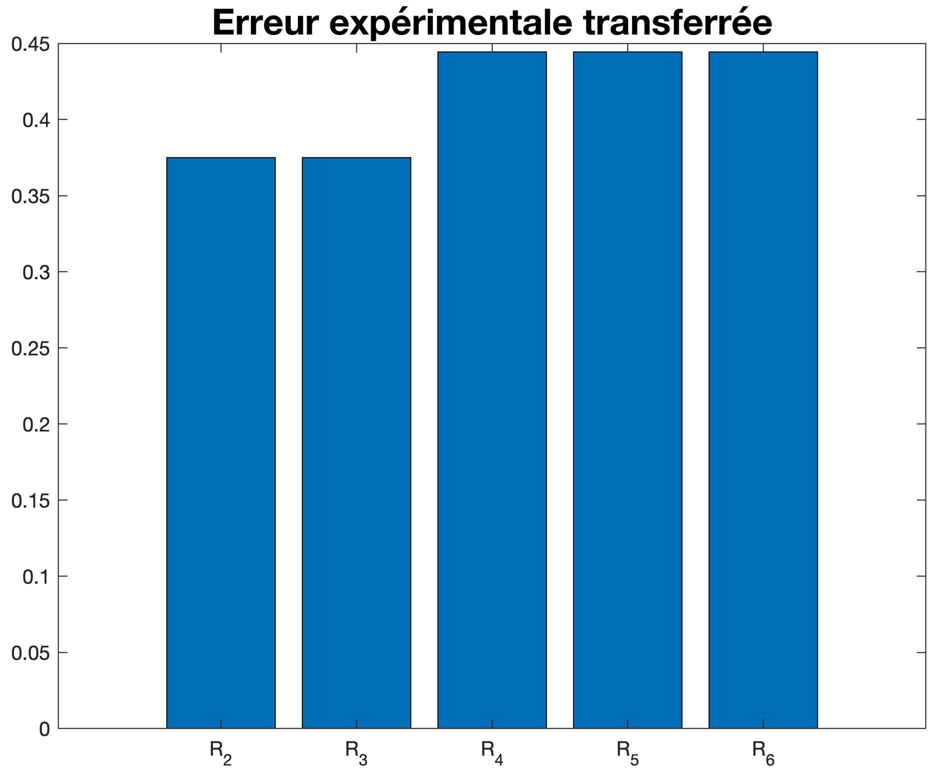
```
C=corrcov(D);
VIF=diag(inv(C))
```

```
VIF =
       3/2
       3/2
       4/3
       4/3
       4/3
```

```
bar(VIF)
title('VIF',"FontSize",18)
xticklabels({'R_2' 'R_3' 'R_4' 'R_5' 'R_6'})
```




```
bar(diag(D))
title('Erreur expérimentale transferrée',"FontSize",18)
xticklabels({ 'R_2' 'R_3' 'R_4' 'R_5' 'R_6'})
```



f) Résoudre le système

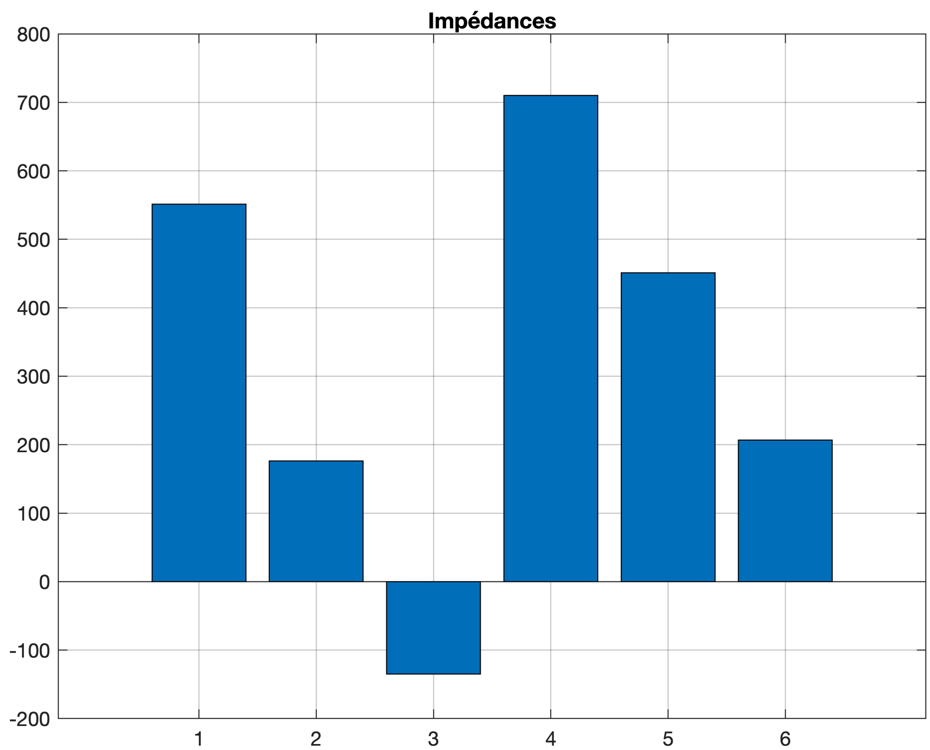
```
Ymes=[990
    1118.3
    850.9
    613.7
    511.0
    1508.3
    1253.9
    1004.1
    193.8
    1198.1
    942.7
    697.7]
```

```
Ymes =
     990
   11183/10
    8509/10
    6137/10
     511
   15083/10
   12539/10
   10041/10
     969/5
   11981/10
    9427/10
    6977/10
```

```
Inductances=(X'*X)\X'*Ymes
```

```
Inductances =
   33077/60
    1761/10
   -2703/20
   21299/30
    4509/10
    2069/10
```

```
bar(Inductances)
grid on
title("Impédances")
```




```
mdl=fitlm(X,Ymes,eye(6))
```

```
mdl =
Linear regression model:
    y ~ x1 + x2 + x3 + x4 + x5 + x6

Estimated Coefficients:
```

	Estimate	SE	tStat	pValue
x1	551.28	179.13	3.0776	0.021729
x2	176.1	179.13	0.98311	0.36352
x3	-135.15	179.13	-0.7545	0.47911
x4	709.97	206.84	3.4325	0.013928
x5	450.9	206.84	2.18	0.072063
x6	206.9	206.84	1.0003	0.35578

```
Number of observations: 12, Error degrees of freedom: 6
Root Mean Squared Error: 253
```