



MONS

UCL Université catholique de Louvain - **Mons**

**Analyse des performances
des installations de chargements camions
TOTAL BELGIUM SA – Dépôt de Feluy**

ANNEXES

Promoteur :
Professeur J.-S. TANCREZ

Mémoire présenté par :
Xavier DUSSART
en vue de l'obtention du diplôme
de Master en sciences de gestion

Année académique 2011-2012

Préambule

Ce document contient des informations confidentielles et stratégiques. Aucune partie de ce document ne peut être publiée, copiée, utilisée ou distribuée d'aucune manière sans l'accord explicite et écrit de TOTAL BELGIUM SA.

Table des annexes :

Annexe 1 : Organigramme du Groupe TOTAL

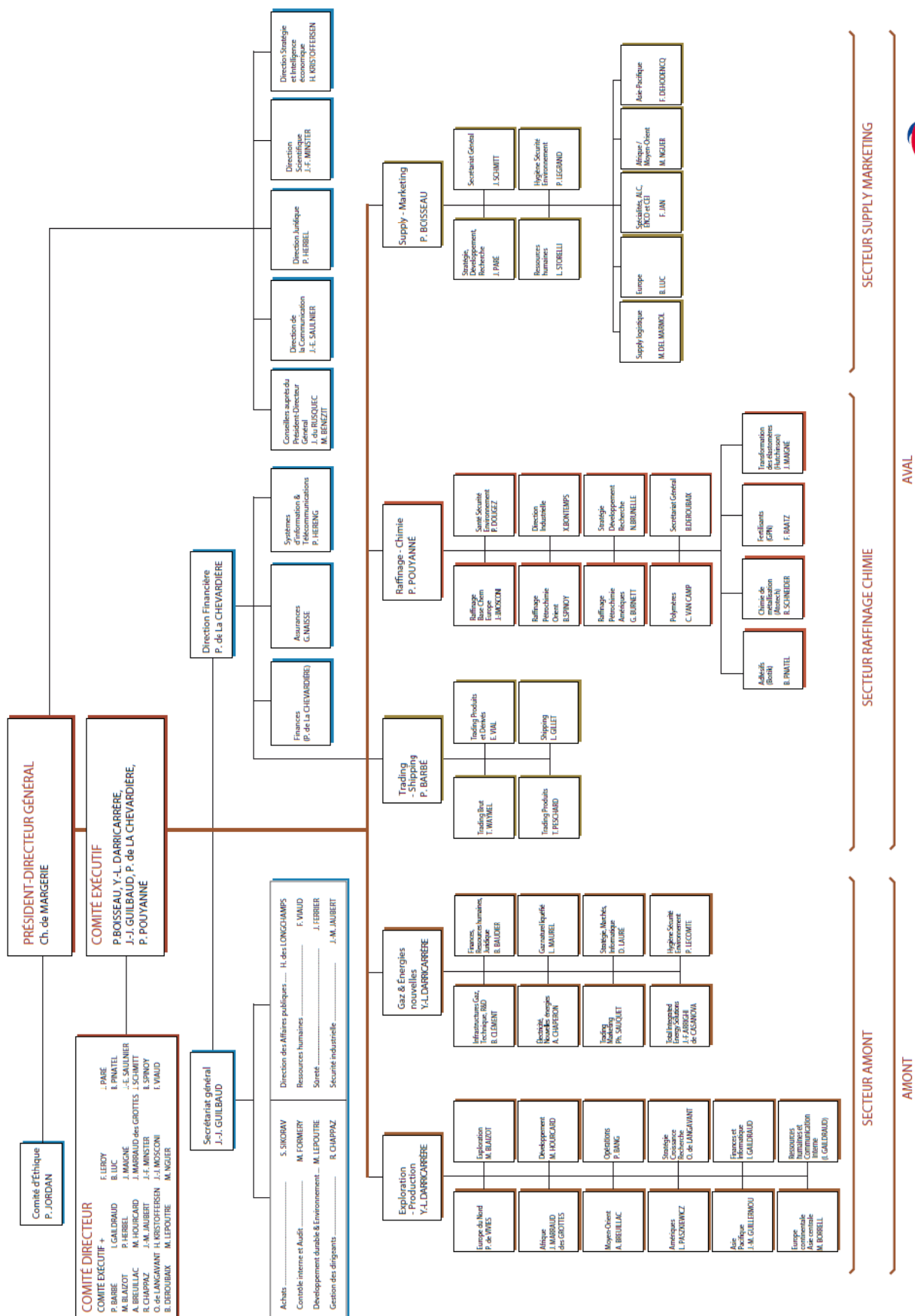
Annexe 2 : Evolution des coûts et utilisations des quais de chargement

Annexe 3 : Programmation

Annexe 4 : Evolution des chargements par quai

Annexe 1 : Organigramme du Groupe TOTAL

TOTAL BELGIUM se trouve dans la branche Aval du groupe TOTAL plus précisément dans le Secteur Supply-Marketing région Europe.



Annexe 2 : Evolution des coûts et utilisations des quais de chargement

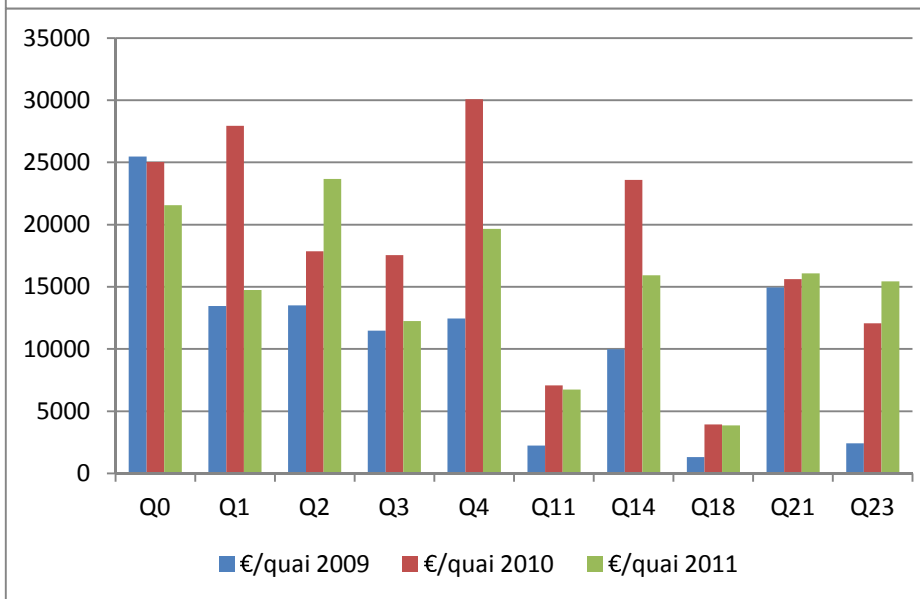
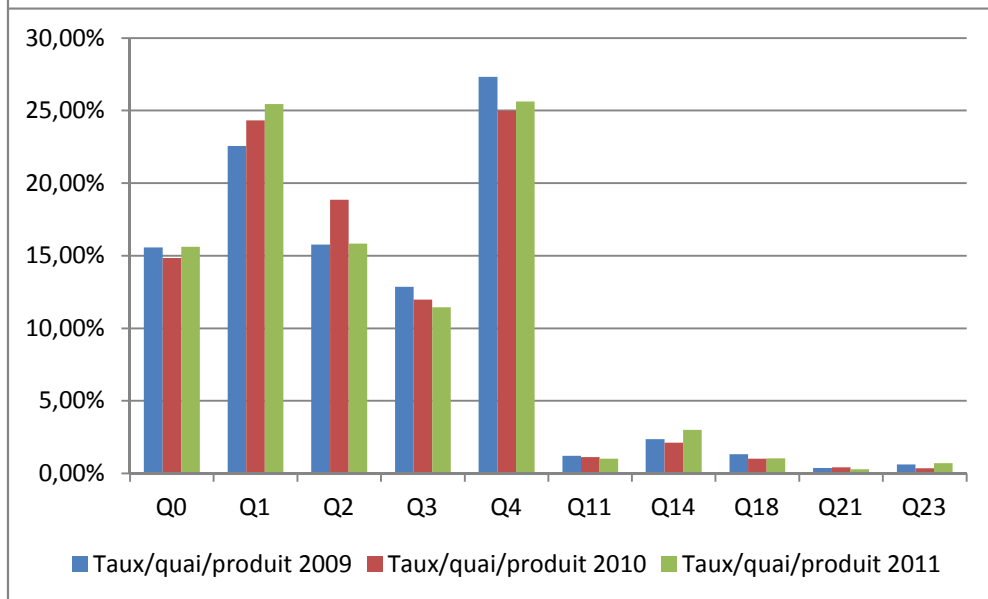
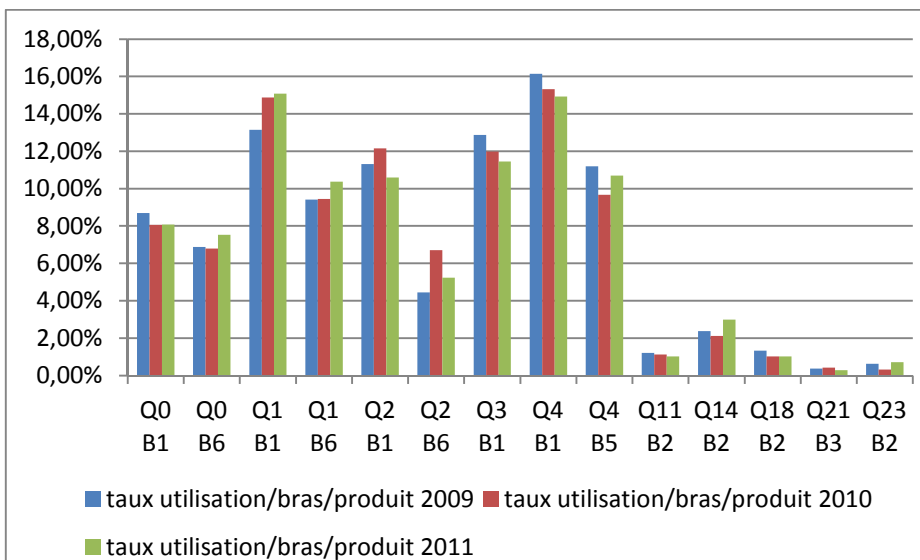
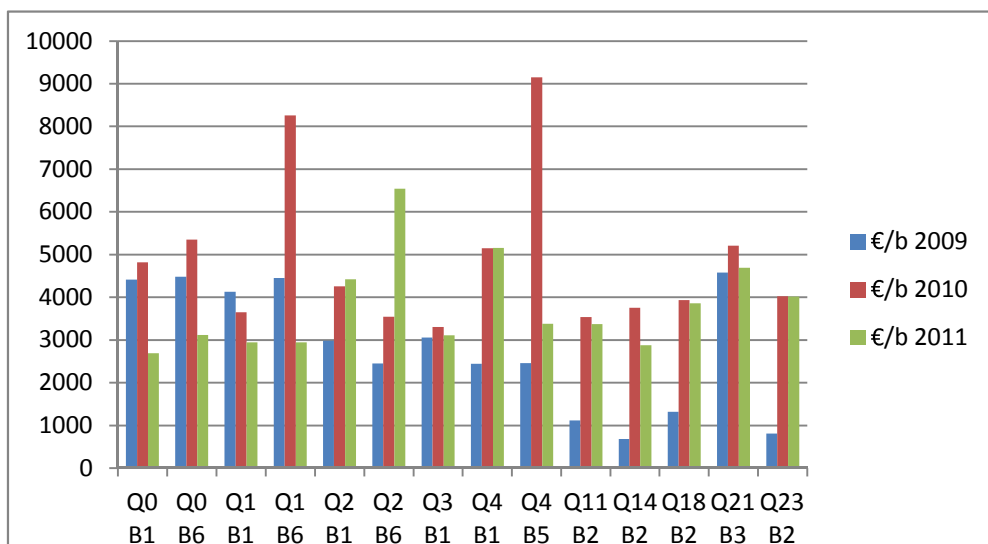
Pour tous les produits, les graphiques suivants sont disponibles :

- a) €/bras : totalité des dépenses pour ce bras de chargement
- b) Taux/bras/produit : volume sorti par ce bras relativement au volume total sorti pour ce produit
- c) Taux/quai/produit : somme des taux d'utilisation des bras pour le produit concerné sur le quai en question (si 2 bras : somme des deux taux)
- d) €/quai : totalité des dépenses pour ce quai (tous les bras confondus)

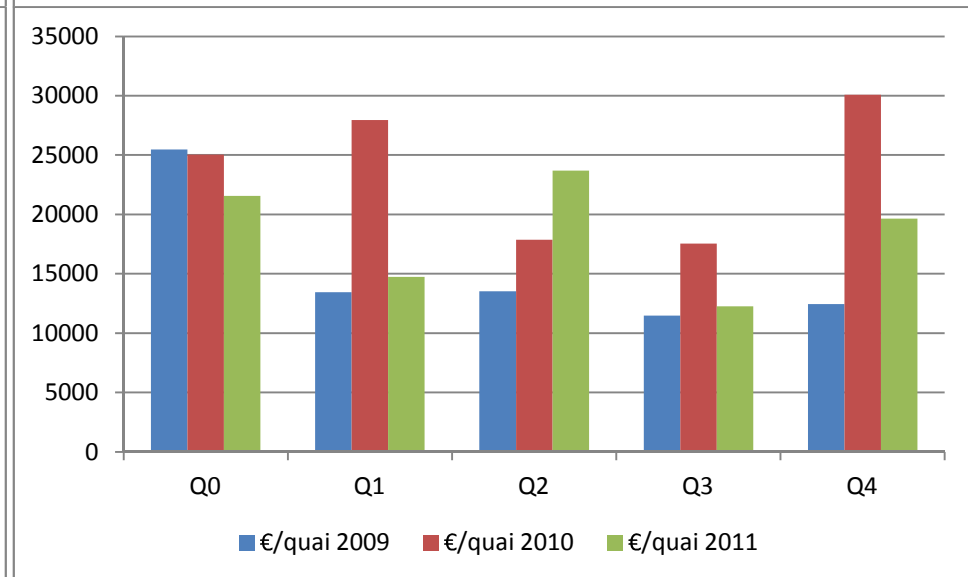
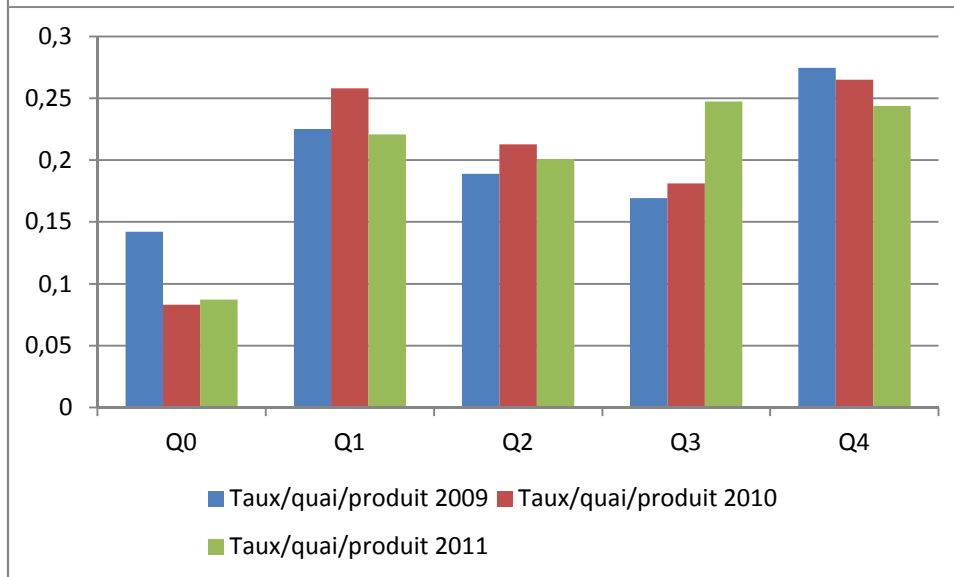
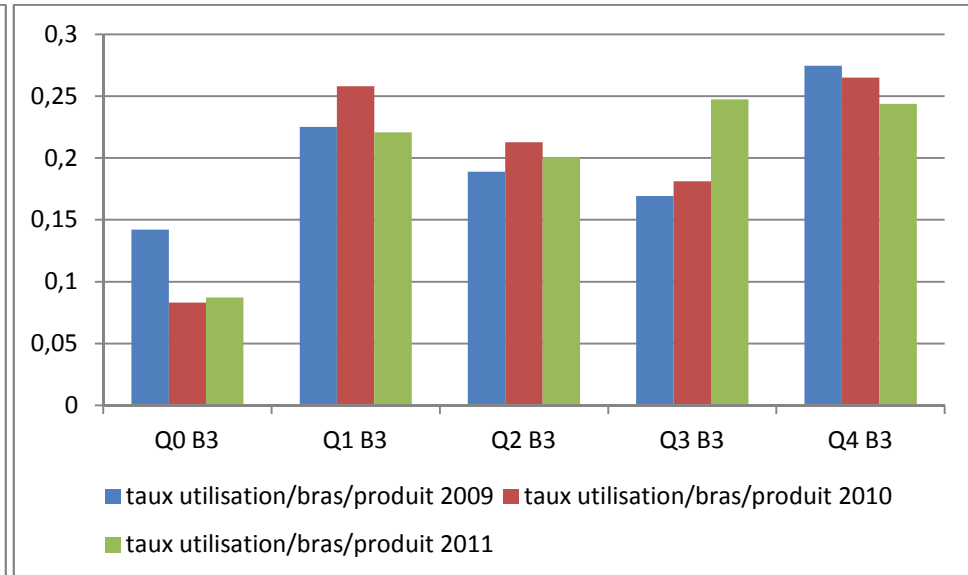
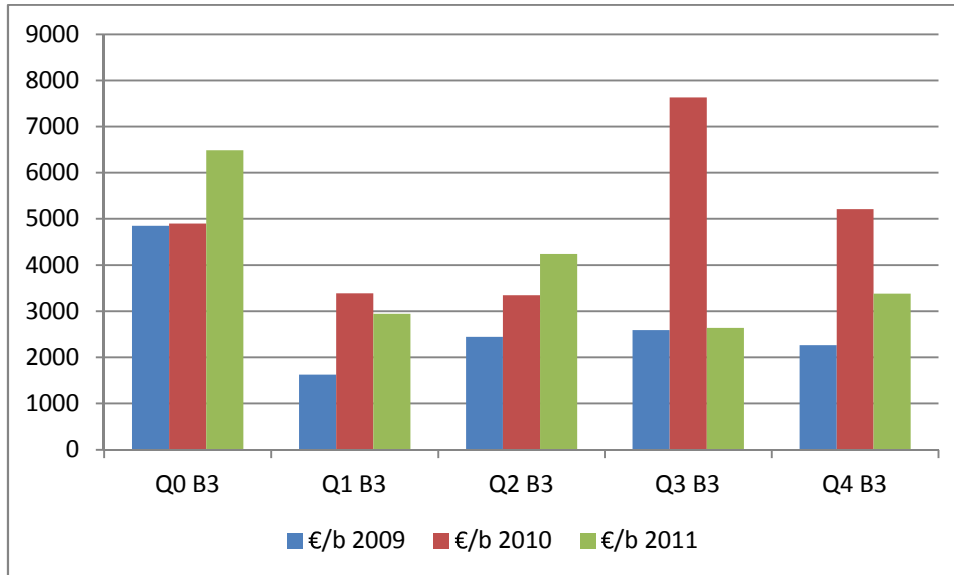
Table des matières

1. Diesel	2
2. Euro 95	3
3. E5	4
4. Super98	5
5. Gasoil de chauffage grand froid	6
6. Gasoil Extra.....	7
7. Pétrole	8
8. Gasoil de chauffage belge	9
9. Gasoil de chauffage français.....	10

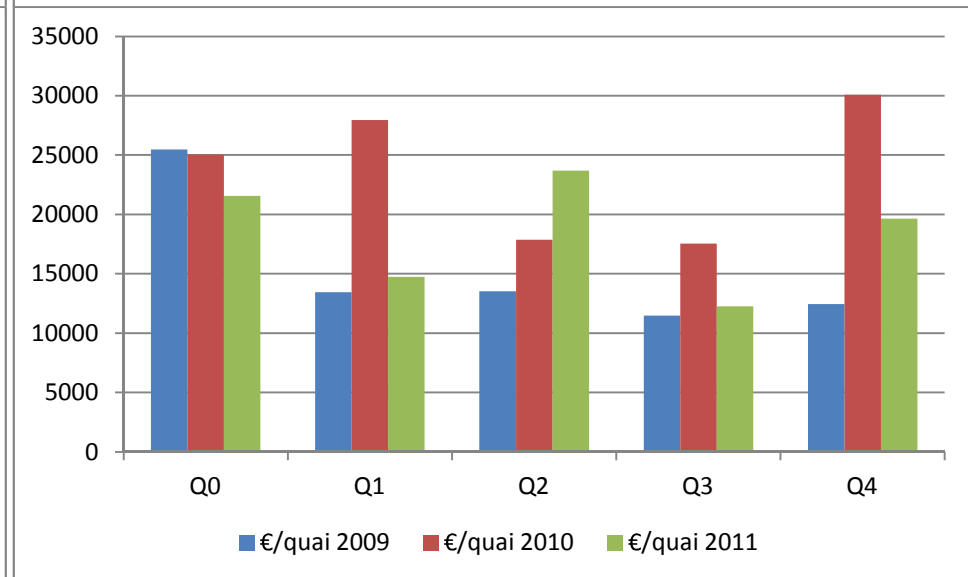
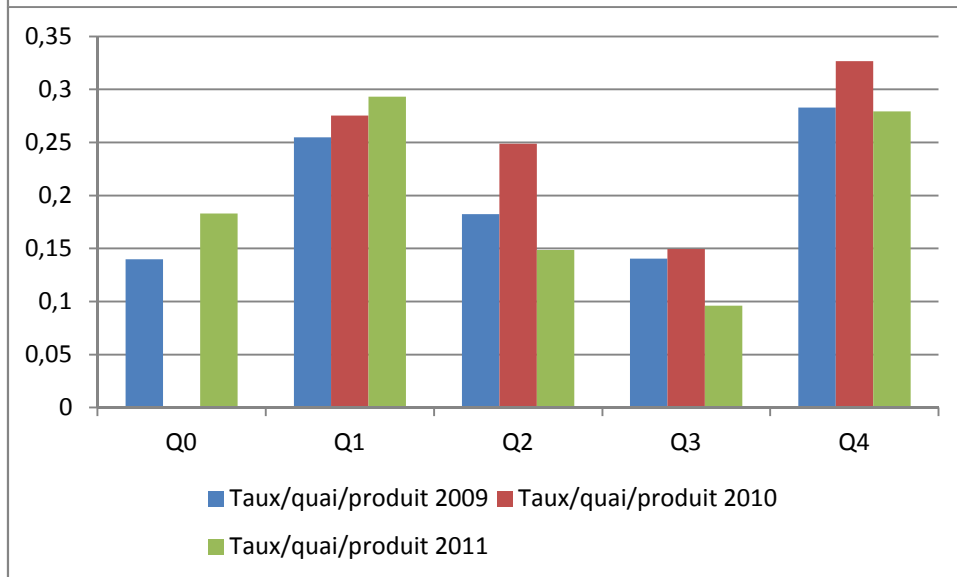
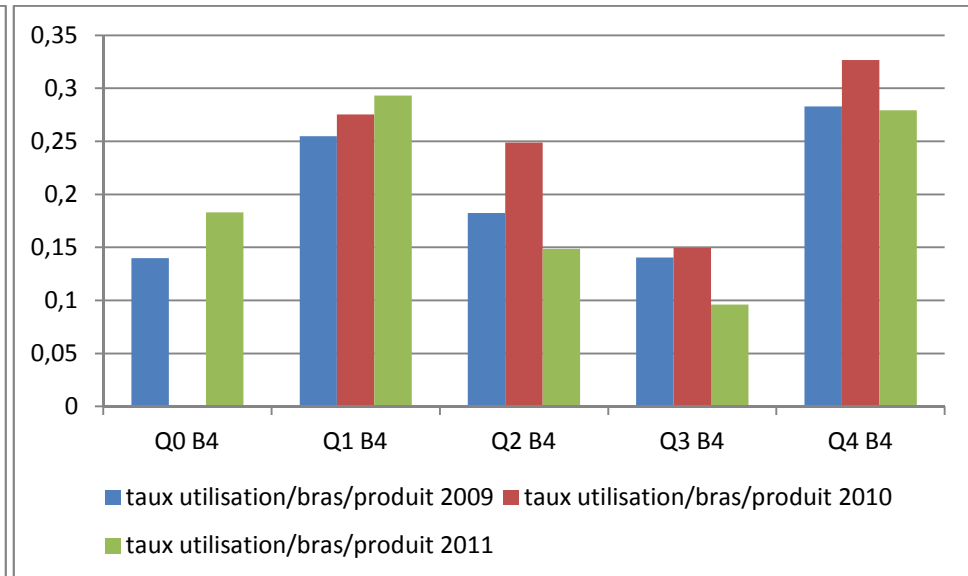
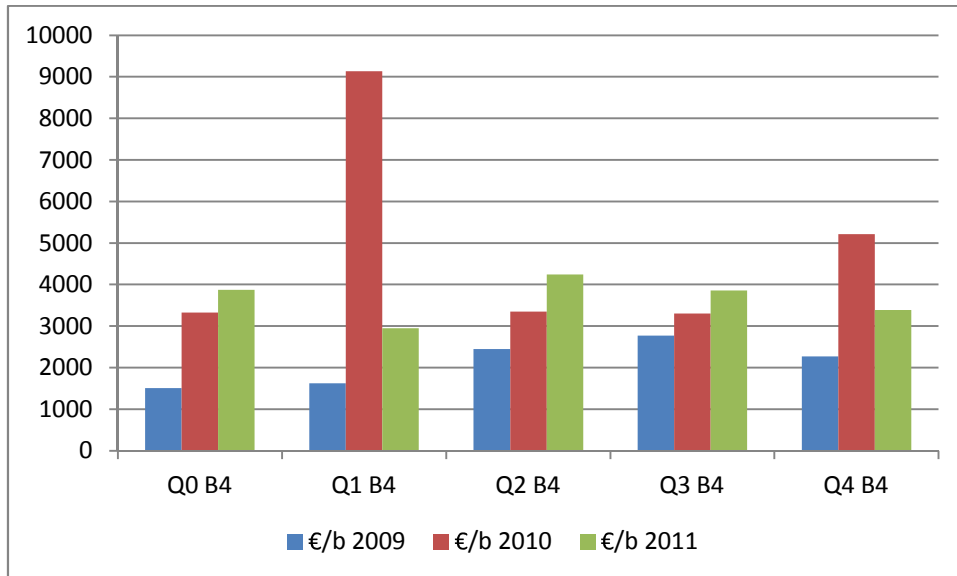
1. Diesel



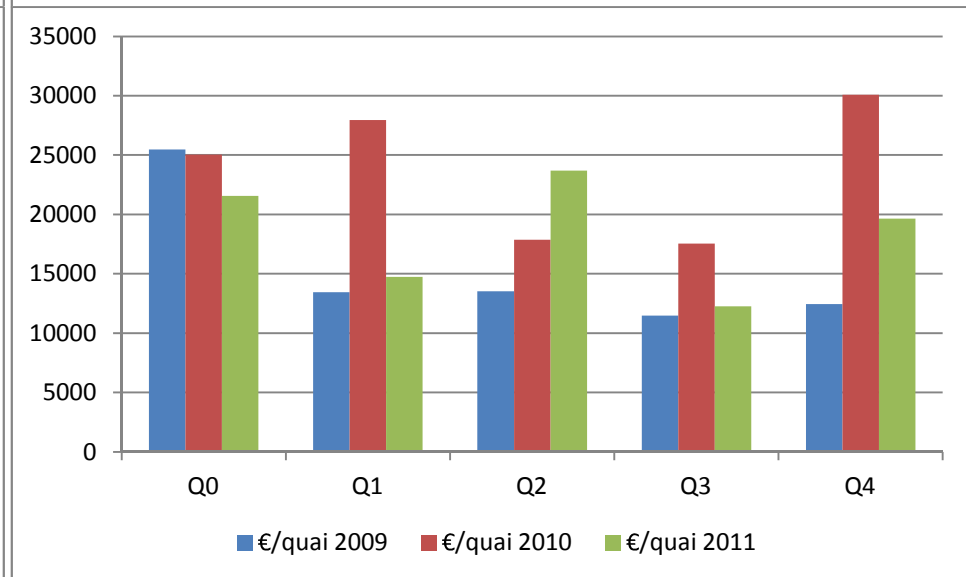
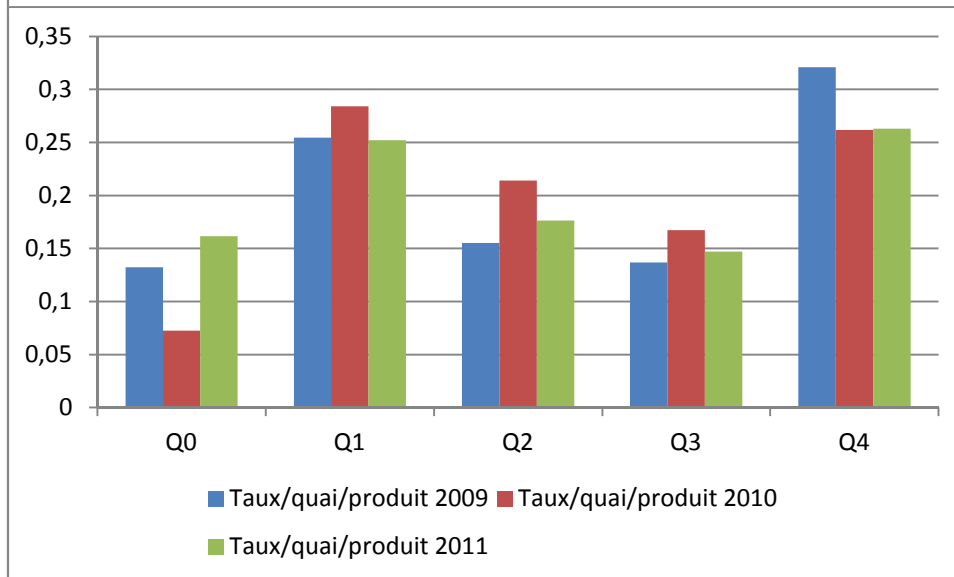
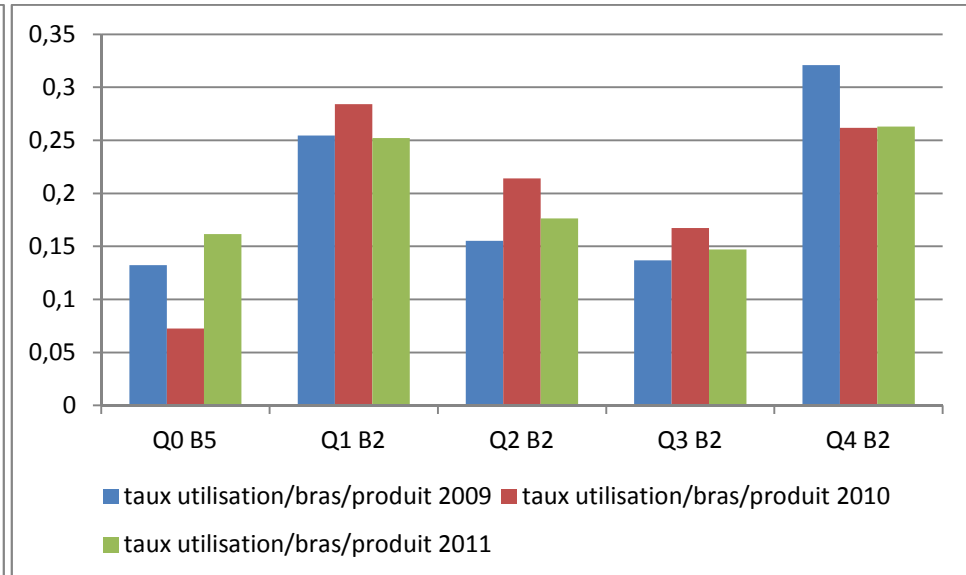
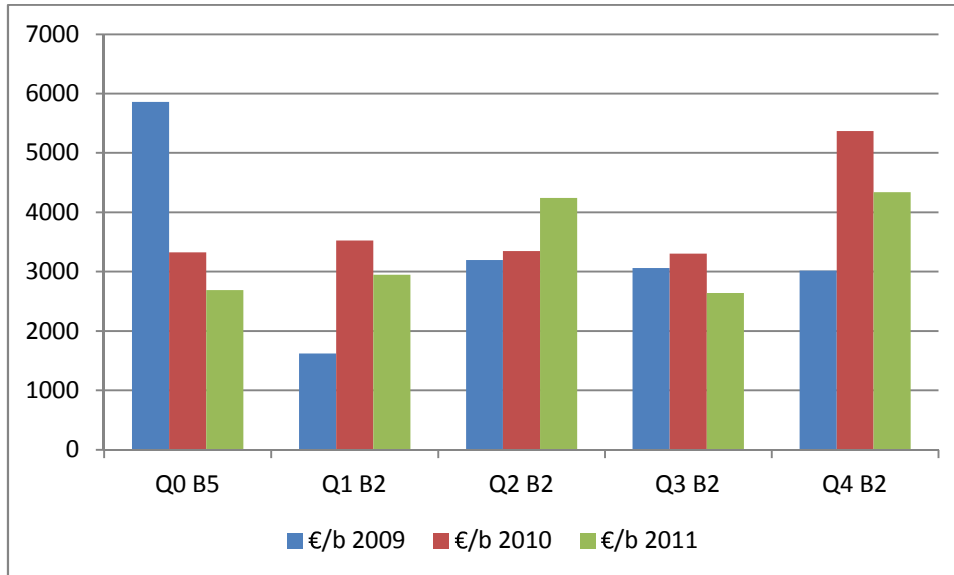
2. Euro 95



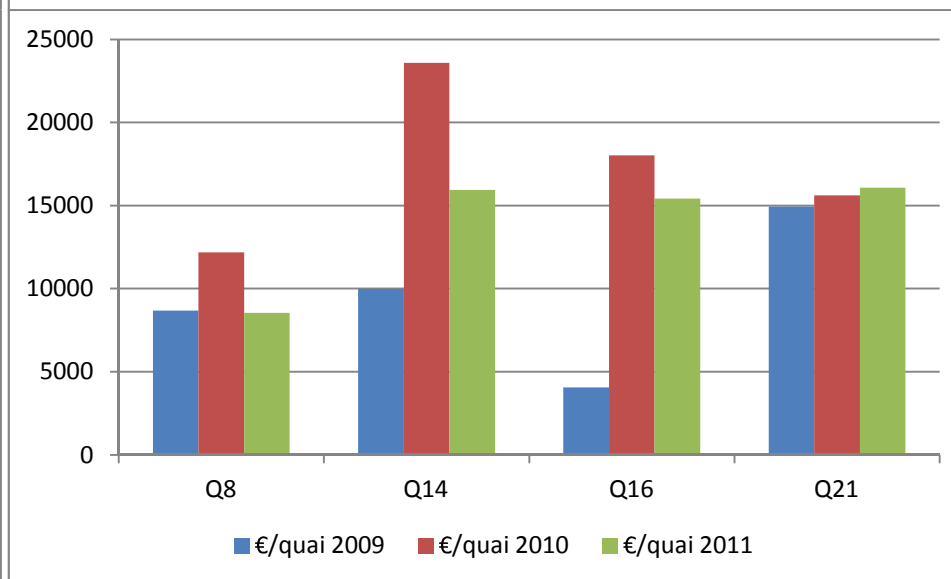
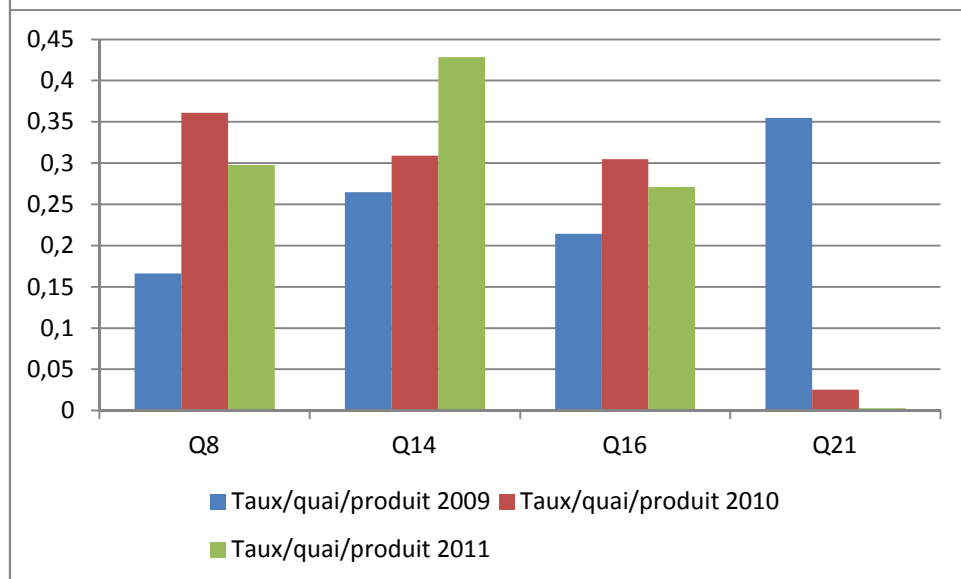
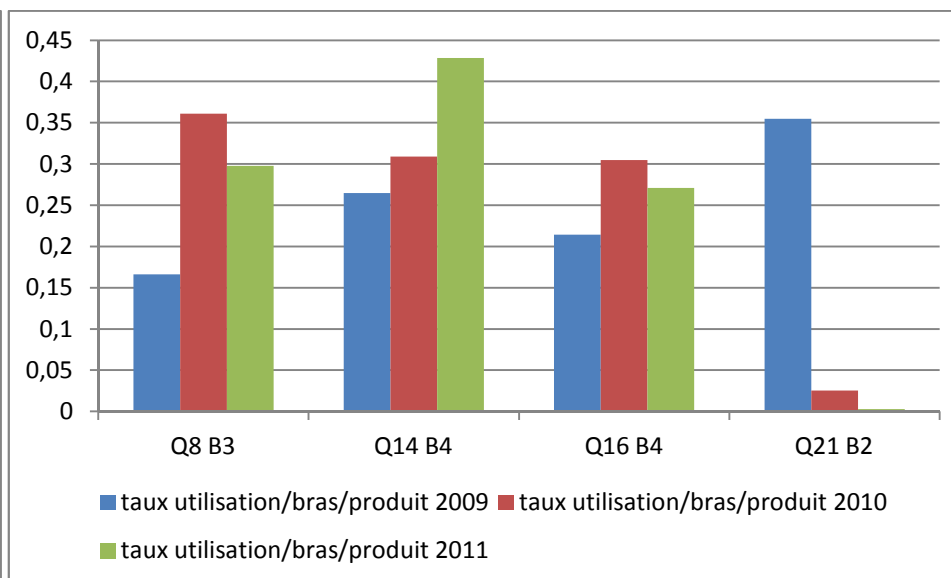
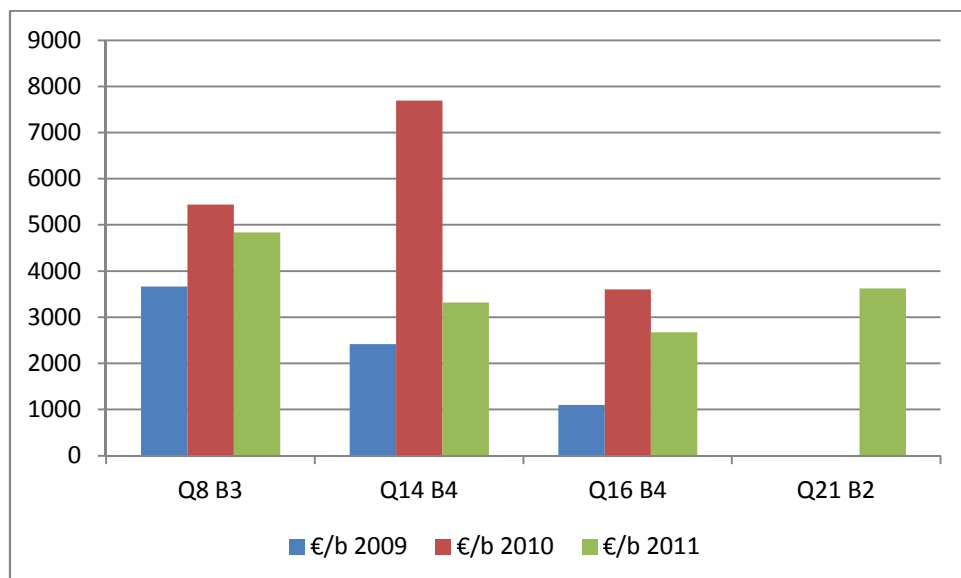
3. E5



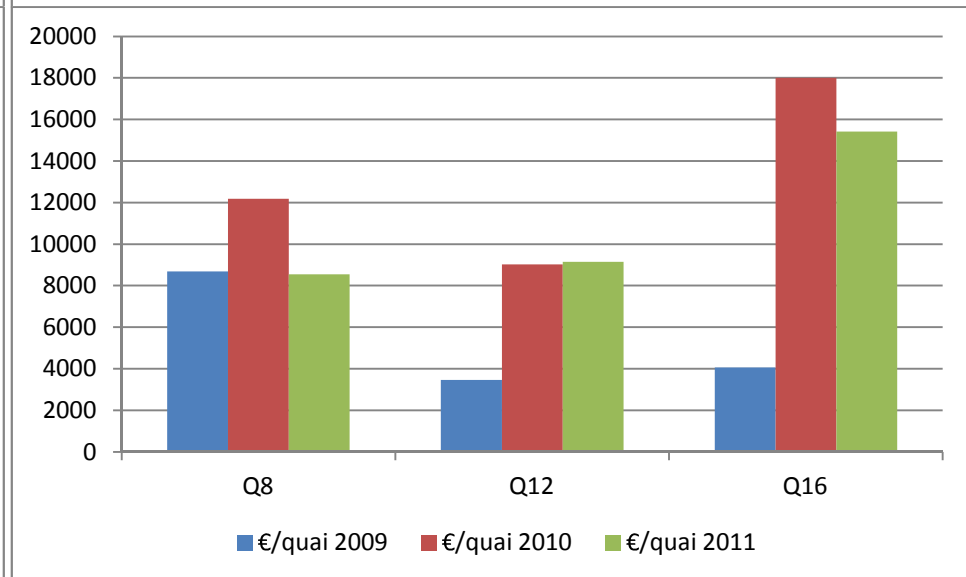
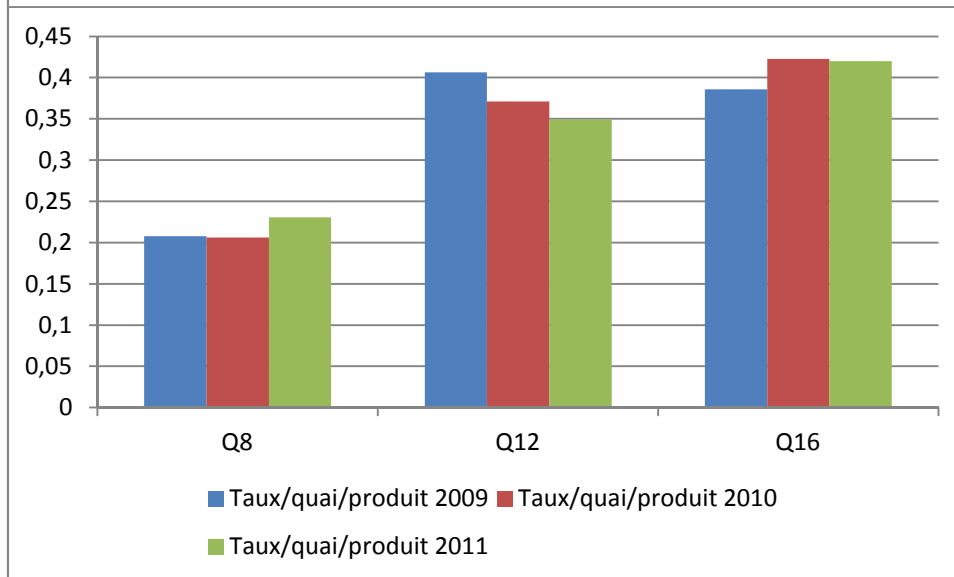
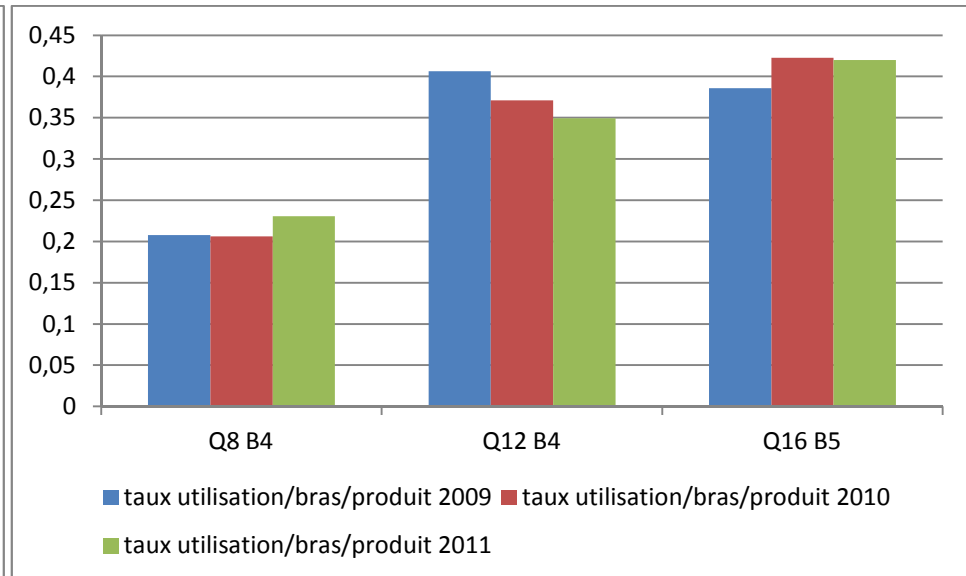
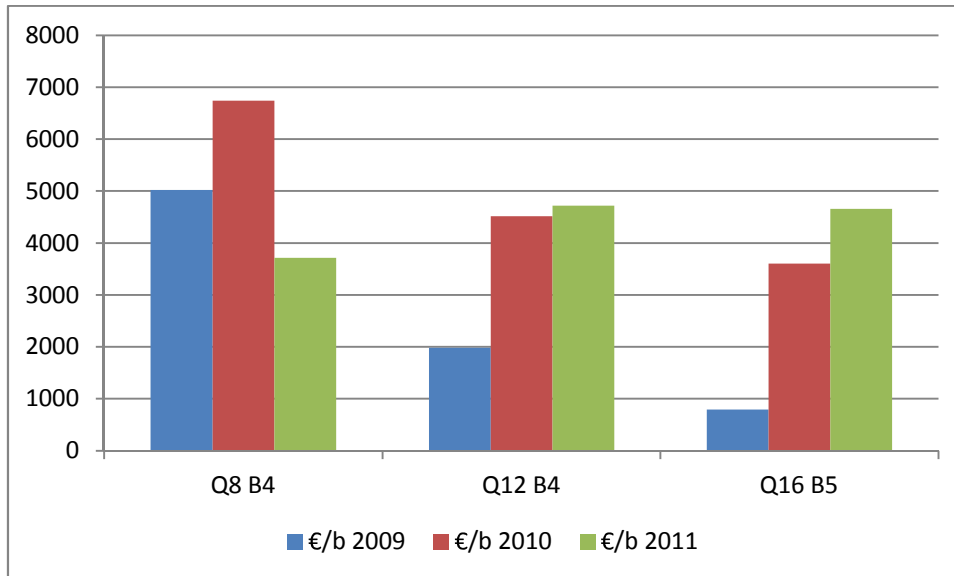
4. Super98



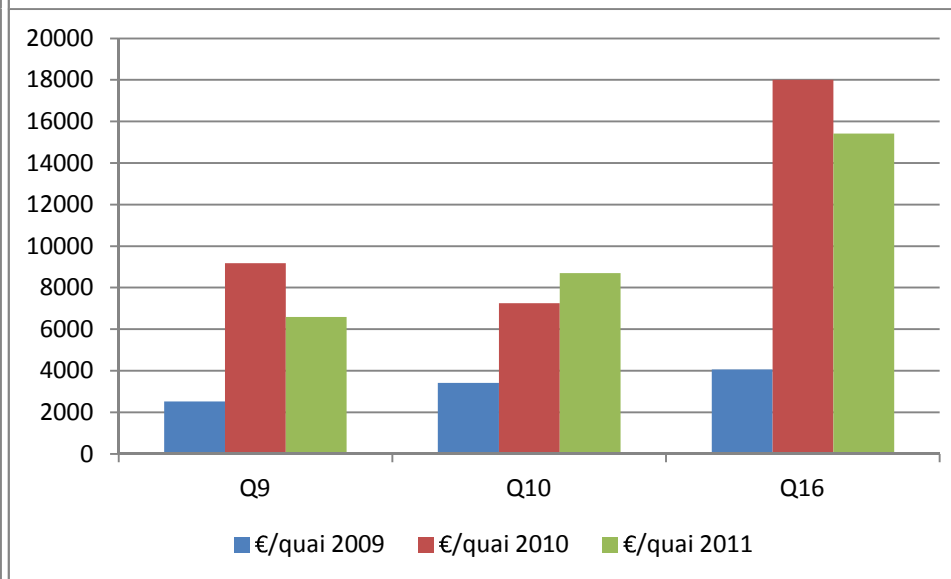
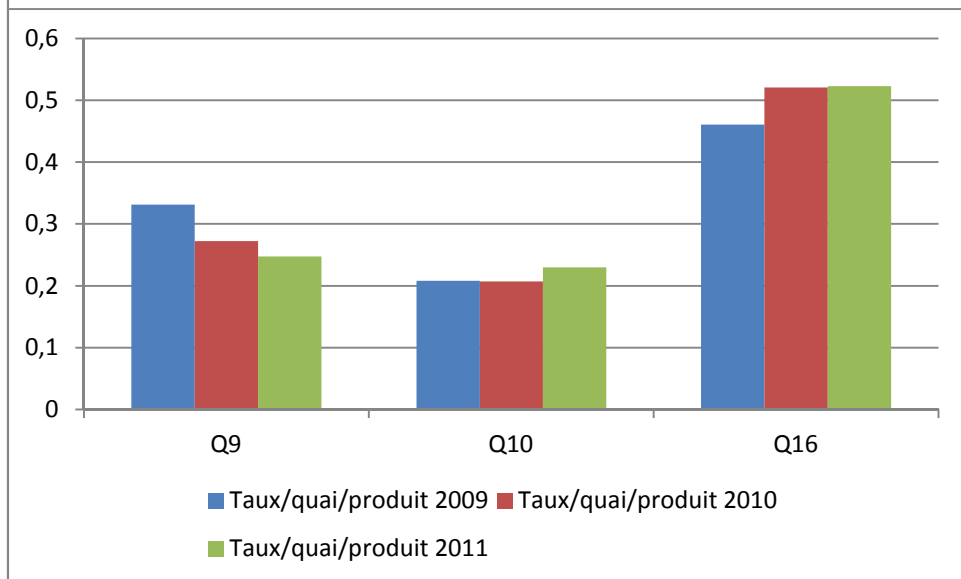
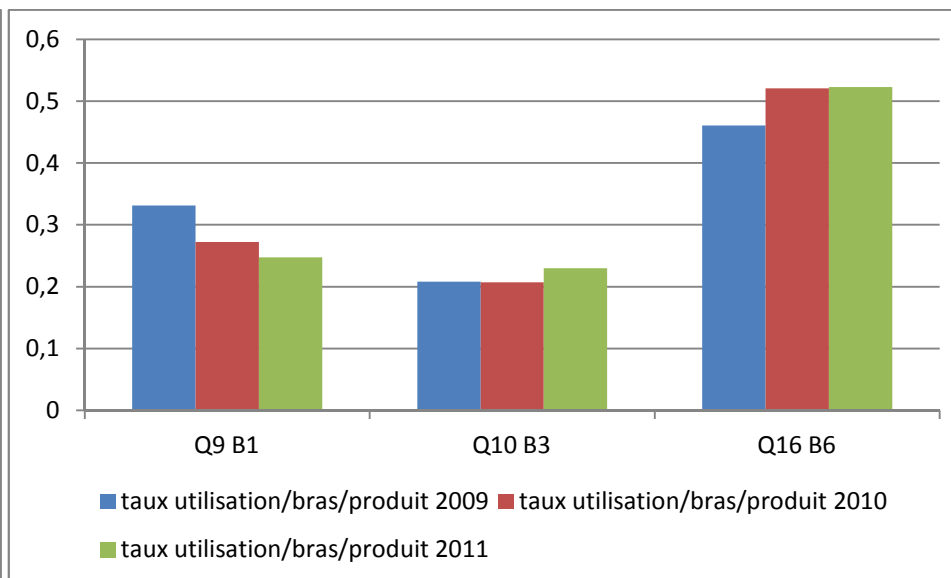
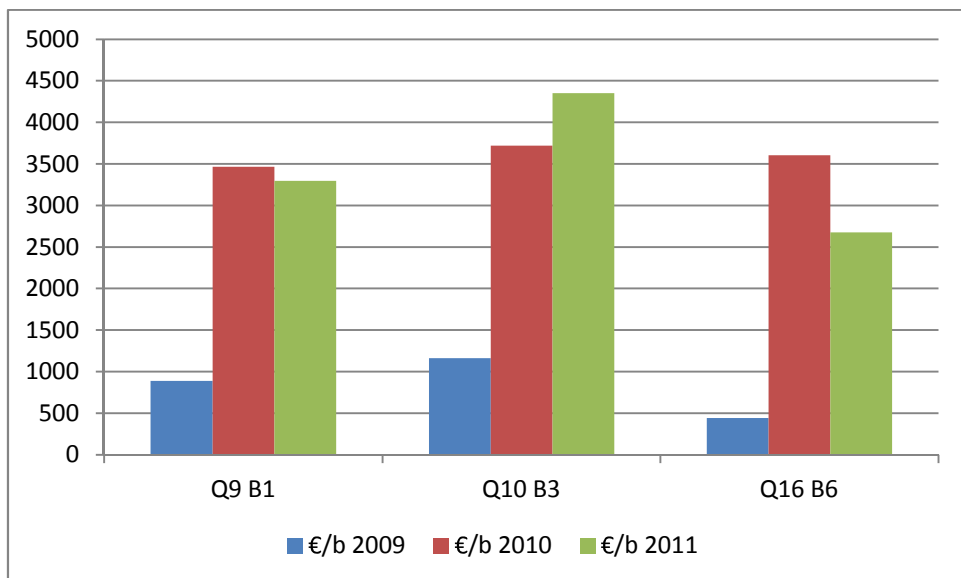
5. Gasoil de chauffage grand froid



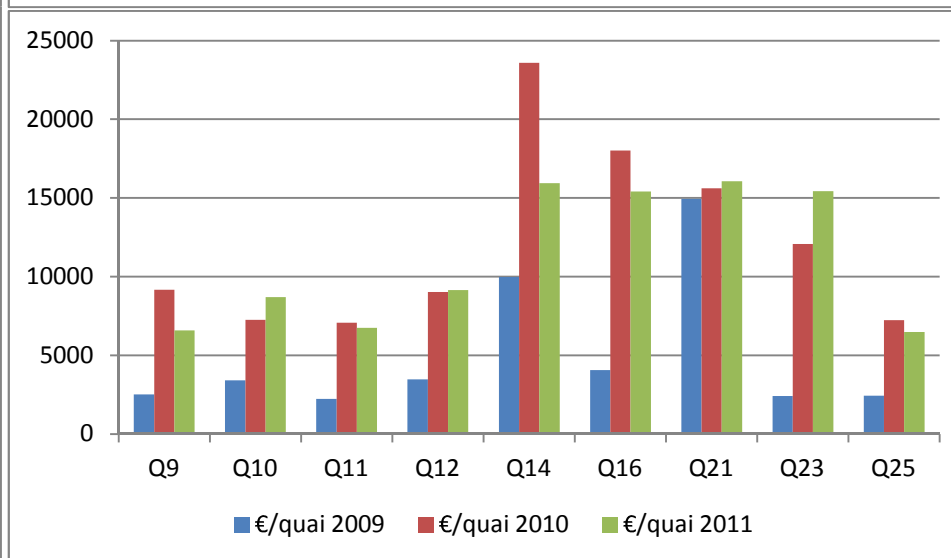
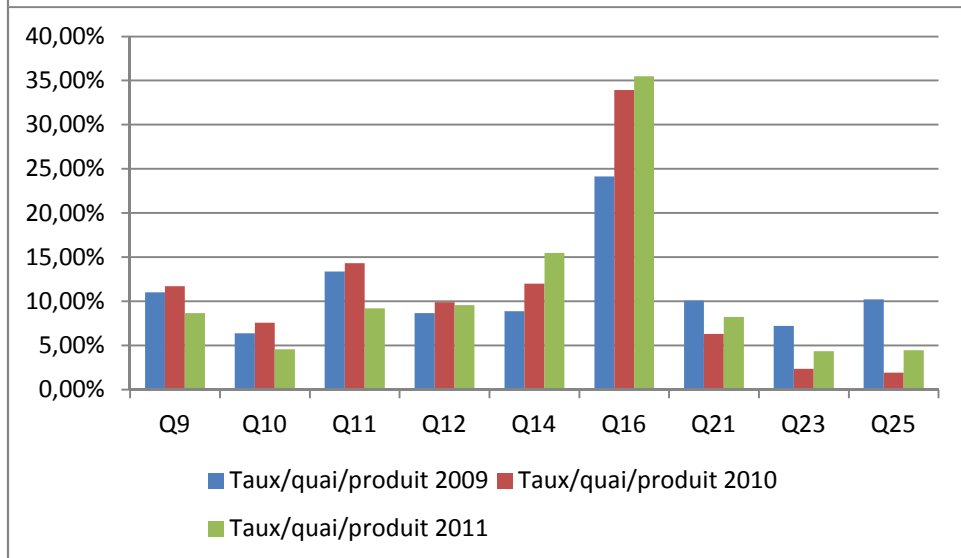
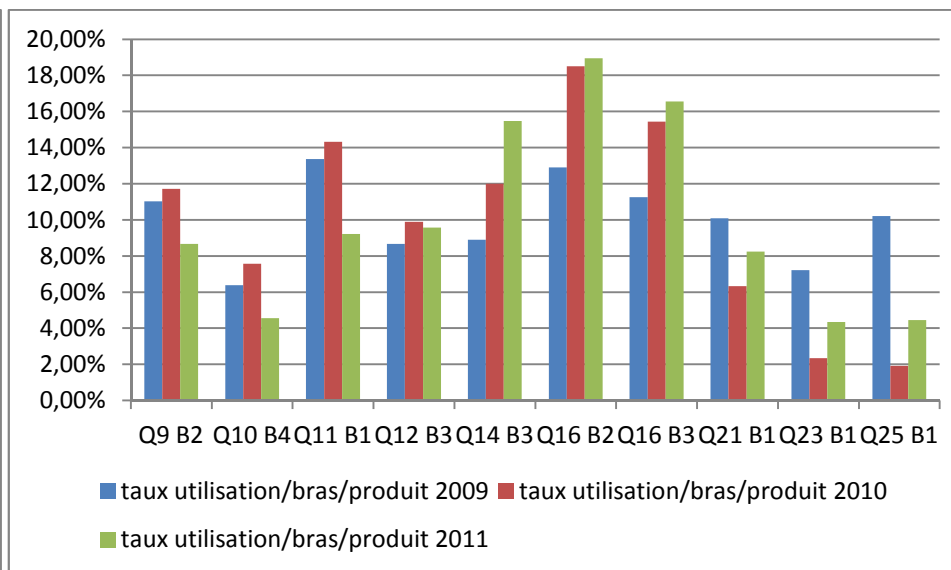
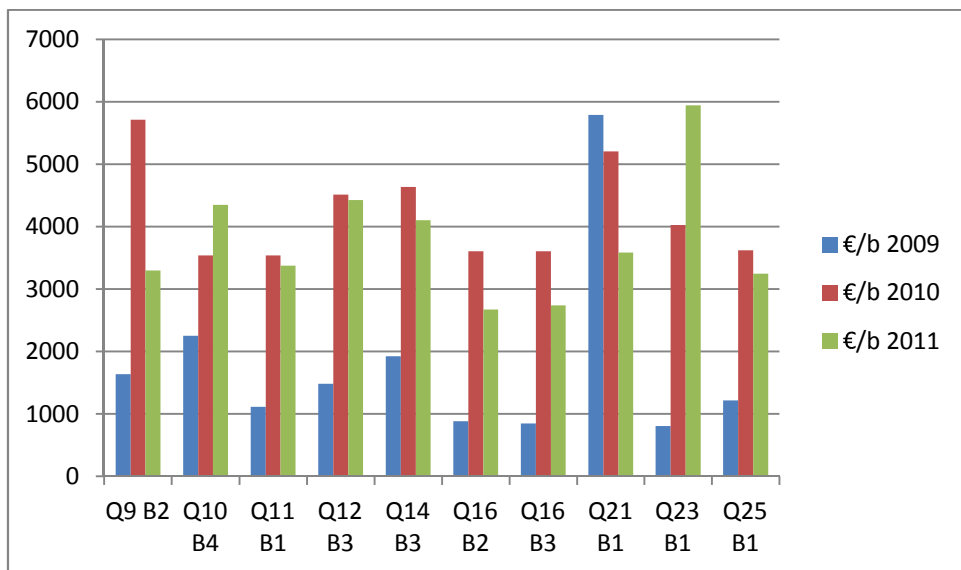
6. Gasoil Extra



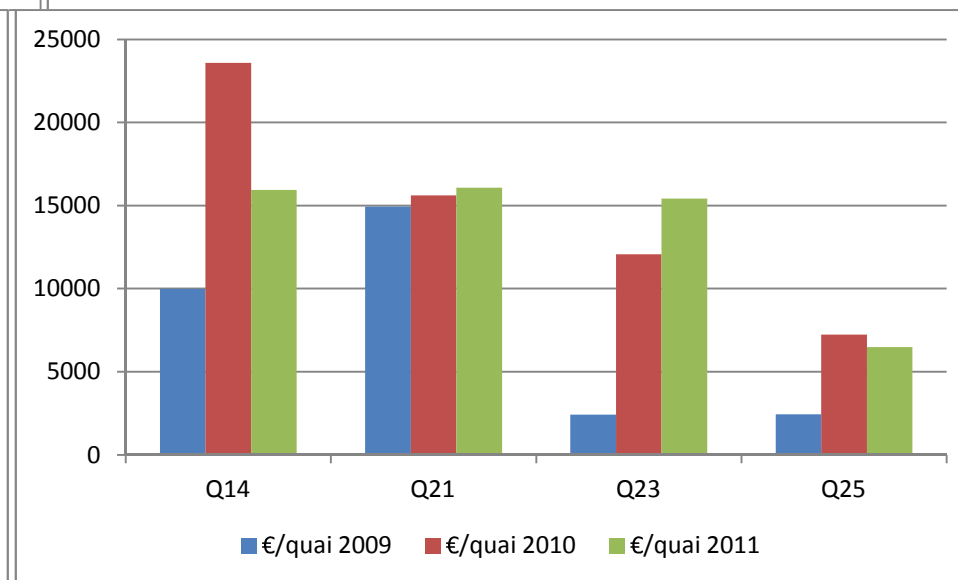
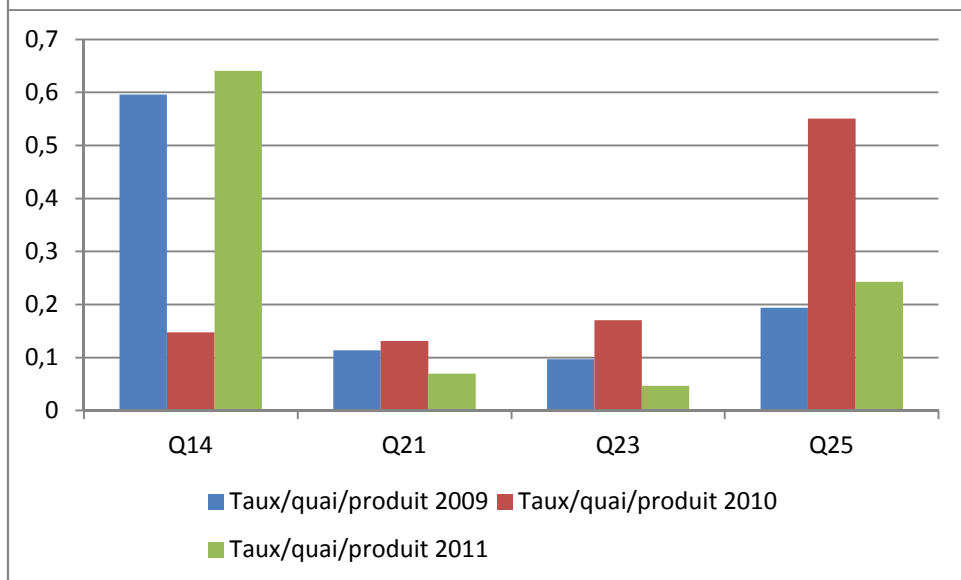
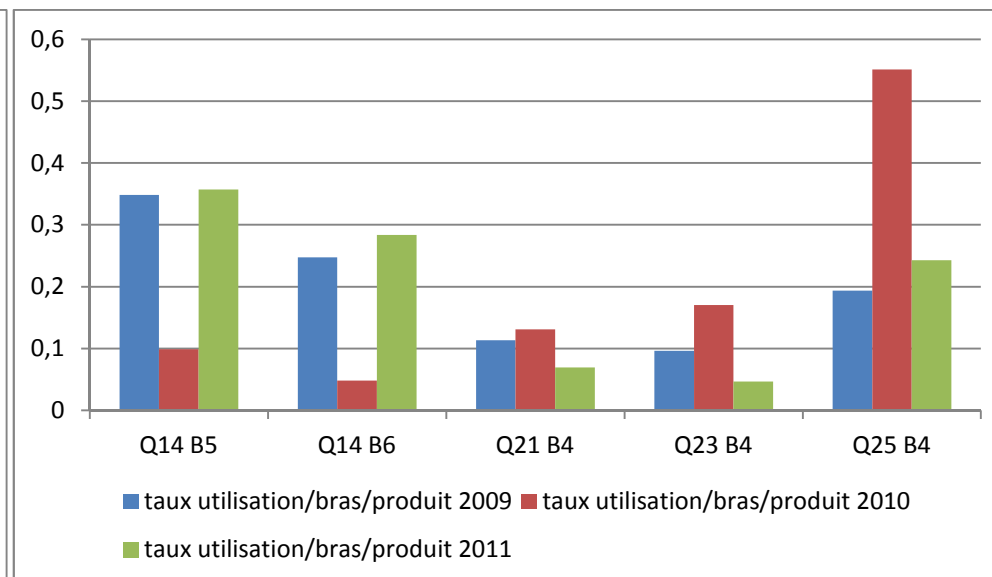
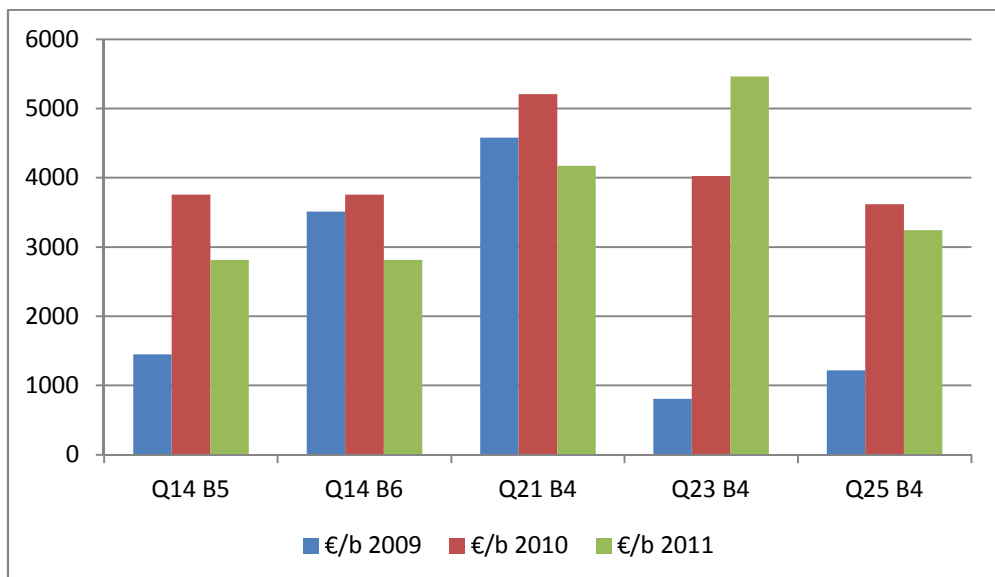
7. Pétrole



8. Gazoil de chauffage belge



9. Gasoil de chauffage français



Annexe 3 : Programmation

Cette annexe reprend les fonctions principales programmées afin d'extraire et traiter les diverses données. Ces fonctions ont été écrites en vue d'une utilisation sous Octave ou MatLab de MathWorks ©.

Table des matières

1. <i>c_l_pouranalyse</i> - Fonction de transfert vers Excel.....	2
2. <i>calc_file_q</i> - Fonction de calcul des files d'attente.....	4
3. <i>cgt_tot</i> - Fonction de calcul du nombre de chargements ainsi que les éventuelles simultanités	8
4. <i>cgt_tot_prod</i> - Fonction de calcul du nombre de chargements ainsi que les éventuelles simultanités (pour un produit particulier).....	9
5. <i>cgtsim</i> - Sous fonction de calcul du nombre de chargements simultanés	10
6. <i>cgtsim_AC1138</i> - Sous fonction de calcul du nombre de chargements simultanés par produit.....	15
7. <i>Chgtsimultanes</i> - Extraction du nombre de chargements simultanés pour le G3 et GD	21
8. <i>Cparli</i> – Calcul du coût par litre	24
9. <i>lit_produit</i> – Calcul des litrages par produit	25
10. <i>Moyenneentre</i> – Calcul des entrées sur site	26
11. <i>Moyenneentre_WE</i> – Calcul des entrées sur site sans les weekends	28
12. <i>Moyenneentre_WE_AC1138</i> – Calcul des entrées sur site sans les weekends configurable pour un produit particulier (ou plusieurs).....	30
13. <i>nb_bottom_bras</i> – Calcul chargements de diesel avec un ou deux bras (durant la même visite)	33
14. <i>nbchgt</i> – Calcul du nombre de chargements par demi-heure	36
15. <i>Nbrcgt2Bb</i> – Calcul de la durée des chargements à deux ou plusieurs bras/quais	39
16. <i>Nbrcgtprod</i> – Calcul de la durée des chargements à deux ou plusieurs bras/quais (boucle)	43
17. <i>nbvisites</i> – Calcul du nombre de visites (pour une journée)	44
18. <i>nbvisites_AC1138</i> – Calcul du nombre de visites configurable pour un ou plusieurs bras/quais (pour une journée)	48
19. <i>pourcentagebrasutilisation</i> – Calcul l'utilisation relative d'un produit pour un bras déterminé.....	52
20. <i>Tot_calc_file_q</i> – Calcul du nombre de camions devant attendre avant de charger	53

1. *c_l_pouranalyse* - Fonction de transfert vers Excel

```
function [ d ] = c_l_pouranalyse( cl09, cl10, cl11, cb09, cb10, cb11, sb09, sb10, sb11, t09, t10, t11 )
% Sortir un tableau pour intégrer dans Excel avec les coûts
% Sortie : tableau cou par litre (quai et bras)
% [donneesanalyse] =
% c-l_pouranalyse(cout_par_litre_2009, cout_par_litre_2010,
% cout_par_litre_2011,
% cout_bras_2009,cout_bras_2010,cout_bras_2011,sorties_l_bras_2009,sorties_
% l_bras_2010,sorties_l_bras_2011, Taux_l_2009, Taux_l_2010, Taux_l_2011);

format long g

d(6,57)=0;
p=0;
tmp1=0;

% -----
% reprise sorties par bras (colonnes 1, 2, 3)

for i=1:1:57
    d(1,i)=sb09(i);
    d(2,i)=sb10(i);
    d(3,i)=sb11(i);
end

% -----
% reprise des coûts par bras

for i=1:1:57
    d(4,i)=cb09(i);
    d(5,i)=cb10(i);
    d(6,i)=cb11(i);
end

% -----
% reprise des coûts au litre par bras

for i=1:1:57
    d(7,i)=cl09(i);
    d(8,i)=cl10(i);
    d(9,i)=cl11(i);
end

% -----
% reprise des taux d'utilisation (taux = sortie du produit concerné/sortie
% totale du produit concerné)

for i=1:1:57
    d(10,i)=t09(i);
    d(11,i)=t10(i);
    d(12,i)=t11(i);
```

```
end
```

```
d=d';
```

```
xlswrite('donnees_analyse.xls', d);
```

```
fprintf('fichier Excle crée du nom donnees_analyse.xls dans le répertoire actuel\n');
```

2. *calc_file_q* - Fonction de calcul des files d'attente

```
function [ file ] = calc_file_q( mois, jour)
% Détermine le nombre de camion ayant attendu au moins un camion
% le nombre indiqué n'est pas le nombre de minutes d'attente car l'heure
% d'arrivée sur site n'est pas disponible

global P1 P2 P3 P4 P5 P6 P7 P8 P9
global compo_Quais_bras
global tp

compo=compo_Quais_bras;
[L,C]=size(P1);
size(compo_Quais_bras);
flag(1)=0;
flag(48)=0;
pres(48,57)=0;
rr=0;

% Tri selon le numéro de visite (pour faire les tests sur une visite)
P1=sortrows(P1,12);
P2=sortrows(P2,12);
P3=sortrows(P3,12);
P4=sortrows(P4,12);
P5=sortrows(P5,12);
P6=sortrows(P6,12);
P7=sortrows(P7,12);
P8=sortrows(P8,12);
P9=sortrows(P9,12);

for pp=1:1:9

    if pp==1
        d=P1;
        size(d);
        pp;
    elseif pp==2
        d=P2;
        pp;
    elseif pp==3
        d=P3;
        pp;
    elseif pp==4
        d=P4;
        pp;
    elseif pp==5
        d=P5;
        pp;
    elseif pp==6
        d=P6;
        pp;
    elseif pp==7
        d=P7;
```

```

    pp;
elseif pp==8
    d=P8;
    pp;
elseif pp==9
    d=P9;
    pp;
end

file(1440,116)=0;
flag(31,24,60)=0; % a vérifier
[L,C]=size(d);
flag(1)=0;
duree=0;
f=0;
time=0;

for i=1:1:L
    if (i>1)% && (d(i,12)~=d(prec,12)) % le check sur la colonne 12 n'est pas bon car on retire alors les chargements
simultanés
        visite_j(L,4)=0;

        if d(i,1)==mois

            if d(i,2)==jour

                for j=1:1:17
                    % boucle sur les 17 quais

                    y=1;
                    quai=1;
                    while d(i,9)~=compo(y,1) || d(i,10)~=compo(y,2) % calcule pour l'élément en cours son quai (de 1 à 17)
                        y=y+1;

                        if(d(i,9)==compo(22,1) && d(i,10)==compo(22,2))
                            y=22;
                        end

                        % Pb Q3.6
                        if(d(i,9)==compo(21,1) && d(i,10)==compo(21,2))
                            quai=compo(y,4);

                            %OK
                        end

                        quai=compo(y,4);

                    end % while d(i,9)~=compo(y,1) || d(i,10)~=compo(y,2)

                    if quai==j && d(i,11)==0
%-----

```

```

        visite_j(i)=d(i,12);

    end % if quai==j && d(i,11)==0

    tmp=unique(visite_j);
    Ltmp=size(tmp)
    visite_j_u(Ltmp,4)= 0;

    visite_j_u(:,2)=tmp;
    visite_j_u(:,1)=quai;
    [Lv,Cv]=size(visite_j_u)

    clear Ltmp tmp

for z=1:1:Lv
    % boucle sur les visites
    visite_j_u(z,3) = 1439;
    visite_j_u(z,4) = 1;

    for r=1:1:L
        while d(r,12)==visite_j_u(z)

            q=1; % heure de début
            while(d(r,7)~=tp(q,1) || d(r,8)~=tp(q,2))
                % check sur l'heure de fin
                q=q+1;
            end

            qq=1;
            while(d(i,7)~=tp(qq,1) || d(i,8)~=tp(qq,2))
                % check sur l'heure de fin
                qq=qq+1;
            end % heure de fin

            visite_j_u(z,3) = min(1439,q);
            visite_j_u(z,4) = max(1,qq);

        end % while d(r,12)==visite_j_u(z)
    end %r=1:1:L

end % for z=1:1:Lv

for zz=1:1:(Lv-1)

    zzzz=zz+1;
    zzz=zz;
    flag=0;

    while (flag==0) && (zzzz<(Lv-1))
        flag = 1;

```

```

        if (visite_j_u(zzzz,3)-visite_j_u(zzz,2)<=6)
            visite_j_u(zzzz,4)=visite_j_u(zzzz,4)+1;
            flag=0;
        end
        zzz=zzz+1;
        zzzz=zzzz+1;

    end % while

    end % zz=1:1:Lv

    end %j=1:1:17
    file(Lv,4)=0;
    file=visite_j_u;
end

end

end

end

end

end

% if pp==9
%   d(1,:)
% end
%

d=0;

if pp==9
    % fprintf('Toutes les données sont passées \n');
end

end

```

3. *cgt_tot* - Fonction de calcul du nombre de chargements ainsi que les éventuelles simultanés

```
function [ evo ] = cgt_tot()
%Détermine le nombre de bras de chargement et Co en simultanés.

% pour la composition, voir le fichier excel relatif.

tic

global P1 P2 P3 P4 P5 P6 P7 P8 P9
global compo_Quais_bras
global tp dates

Le=0;
Ce=0;
[L,C]=size(dates);

evo(L,1440,86)=0;

for i=1:1:L

    mois=dates(i,1)
    jour=dates(i,2)
    evotmp=cgtsim(mois,jour);

    [Le,Ce]=size(evotmp);
    Cecpt=Ce+2;
    for j=3:1:Cecpt
        evo(i,:,1)=mois;
        evo(i,:,2)=jour;
        tt=j-2;
        evo(i,:,j)=evotmp(:,tt);
    end

end

fprintf('Fin \n');
toc
```

4. *cgt_tot_prod* - Fonction de calcul du nombre de chargements ainsi que les éventuelles simultanités (pour un produit particulier)

```
function [ evo ] = cgt_tot_prod()
%Détermine le nombre de bras de chargement et Co en simultanéité.

% pour la composition, voir le fichier excel relatif.

%

tic

global P1 P2 P3 P4 P5 P6 P7 P8 P9
global compo_Quais_bras
global tp dates

Le=0;
Ce=0;
[L,C]=size(dates);

evo(L,1440,86)=0;

for i=1:1:L

    mois=dates(i,1)
    jour=dates(i,2)
    evotmp=cgtsim_prod(mois,jour);

    [Le,Ce]=size(evotmp);
    Cecpt=Ce+2;
    for j=3:1:Cecpt
        evo(i,:,1)=mois;
        evo(i,:,2)=jour;
        tt=j-2;
        evo(i,:,j)=evotmp(:,tt);
    end

end

fprintf('Fin \n');
toc
```

5. *cgtsim* - Sous fonction de calcul du nombre de chargements simultanés

```
function [ file ] = cgtsim( mois, jour)
% calcule le nombre de chargements simultanés pour un seul jour
% avec beaucoup de choix : 84
```

```
global P1 P2 P3 P4 P5 P6 P7 P8 P9
global compo_Quais_bras
global tp
```

```
compo=compo_Quais_bras;
[L,C]=size(P1);
size(compo_Quais_bras);
flag(1)=0;
flag(48)=0;
pres(48,57)=0;
rr=0;
```

```
% Tri selon le numéro de visite (pour faire les tests sur une visite)
```

```
P1=sortrows(P1,12);
P2=sortrows(P2,12);
P3=sortrows(P3,12);
P4=sortrows(P4,12);
P5=sortrows(P5,12);
P6=sortrows(P6,12);
P7=sortrows(P7,12);
P8=sortrows(P8,12);
P9=sortrows(P9,12);
```

```
for pp=1:1:9
```

```
    if pp==1
        d=P1;
        size(d);
        pp;
    elseif pp==2
        d=P2;
        pp;
    elseif pp==3
        d=P3;
        pp;
    elseif pp==4
        d=P4;
        pp;
    elseif pp==5
        d=P5;
        pp;
    elseif pp==6
        d=P6;
        pp;
    elseif pp==7
```

```

        d=P7;
        pp;
    elseif pp==8
        d=P8;
        pp;
    elseif pp==9
        d=P9;
        pp;
    end

file(1440,84)=0;
flag(31,24,60)=0; % a vérifier
[L,C]=size(d);
flag(1)=0;
duree=0;
f=0;
time=0;

for i=1:1:L
    if d(i,11)==0 % s'assurer que l'on regarde produit pas additif
        % prec=i-1;
        if (i>1)% && (d(i,12)~=d(prec,12)) % le check sur la colonne 12 n'est pas bon car on retire alors les chargements
            simultanés

            if d(i,1)==mois

                if d(i,2)==jour

                    %-----

                    for q=1:1:1440
                        % boucle sur les minutes/heures de la journée
                        % =====
                        % incrément positif

                        if(d(i,3)==tp(q,1) && d(i,4)==tp(q,2))

                            % pour chaque produit, regarder le produit
                            % actuellement en charge

                            qq=1;
                            prod=compo(qq,3);
                            colonne_quai=compo(qq,4)+67;

                            while(d(i,7)~=tp(qq,1) || d(i,8)~=tp(qq,2))
                                % check sur l'heure de fin
                                qq=qq+1;
                            end

                            duree=qq-q;

                            % calculer la durée : faire le lien avec

```

```

% les q de arrivée et départ. La différence
% donne la durée. Ensuite faire une boucle
% sur l'incréméntation (avec maximim
% 23h59). Faire une boucle pour déterminer
% cela.

y=1;
prod=compo(y,3);
colonne_quai=compo(y,4)+67;

%         if(d(i,9)~=compo(21,1) || d(i,10)~=compo(21,2) )
%             d(i,:);
%         end

while d(i,9)~=compo(y,1) || d(i,10)~=compo(y,2)
    y=y+1;

    if(d(i,9)==compo(22,1) && d(i,10)==compo(22,2))
        y=22;

    end

%         if(d(i,9)==compo(21,1) && d(i,10)==compo(21,2))
%             y=17;
%         end

%         % Pb Q3.6
%         if(d(i,9)==compo(21,1) && d(i,10)==compo(21,2))
%             prod=compo(y,3);
%             colonne_quai=compo(y,4)+67;
%             %OK
%         end

        prod=compo(y,3);
        colonne_quai=compo(y,4)+67;
end

colonne_prod=prod;

file(q,colonne_prod)=file(q,colonne_prod)+1;
colonne_bras=y+10;
file(q,colonne_bras)=file(q,colonne_bras)+1;
file(q,colonne_quai)=file(q,colonne_quai)+1;

%         if (d(i,1)==3 && d(i,2)==9 && y==4 )
%             d(i,:);
%         end

if duree ~= 0
    for tm=1:1:(duree)

```

```

        tmi=q+tm;
        if tmi<=1440
            file(tmi,colonne_prod)=file(tmi,colonne_prod)+1;
            file(tmi,colonne_bras)=file(tmi,colonne_bras)+1;
            file(tmi,colonne_quai)=file(tmi,colonne_quai)+1;
        end
% =====
% incrément négatif
        if(d(i,7)==tp(q,1) && d(i,8)==tp(q,2))

            % pour chaque produit, regarder le produit
            % dont le chargement est fini

            y=1;
            prod=compo(y,3);
            colonne_quai=compo(y,4)+67;

%            if(d(i,9)~=compo(21,1) || d(i,10)~=compo(21,2) )
%                d(i,:)
%            end

            while d(i,9)~=compo(y,1) || d(i,10)~=compo(y,2)
                y=y+1;

                if(d(i,9)==compo(22,1) && d(i,10)==compo(22,2))
                    y=22;

                end

                if y>=58
                    fprintf('\n Y>=58 ');
                    y
                end

                prod=compo(y,3);
                colonne_quai=compo(y,4)+67;
            end

            colonne_prod=prod;
            file(q,colonne_prod)=file(q,colonne_prod)-1;
            colonne_bras=y+10;
            file(q,colonne_bras)=file(q,colonne_bras)-1;
            file(q,colonne_quai)=file(q,colonne_quai)-1;
        end
%-----
    end
end
end
end
end
end

```

```
        end
    end
    % if pp==9
    %     d(1,:)
    % end
    %

    d=0;

    if pp==9
        % fprintf('Toutes les données sont passées \n');
    end

    end
```

6. *cgtsim_AC1138* - Sous fonction de calcul du nombre de chargements simultanés par produit

```
function [ file ] = cgtsim_AC1138( mois, jour)
%Détermine le nombre de chargement simultanés en Excellium sur les bottoms.
% le nombre indiqué est le nombre de minutes où autant de bras charges.

% calcule le nombre de chargements simultanés pour un seul jour
% avec beaucoup de choix : 111 (voir fichier)
% les 84 premières colonnes sont identiques à l'autre : cgtsim

% modification pour F10 sur Q21-22 : dans fichier compo_bras : colonne 6
% est le F10 sur ce quai. + remplacement de compo(y,5) par compo(y,6)
% partout dans le code (= 2 fois). et changé à L73 file(1440,116)=0;
```

```
global P1 P2 P3 P4 P5 P6 P7 P8 P9
global compo_Quais_bras
global tp
```

```
compo=compo_Quais_bras;
[L,C]=size(P1);
size(compo_Quais_bras);
flag(1)=0;
flag(48)=0;
pres(48,57)=0;
rr=0;
```

```
% Tri selon le numéro de visite (pour faire les tests sur une visite)
P1=sortrows(P1,12);
P2=sortrows(P2,12);
P3=sortrows(P3,12);
P4=sortrows(P4,12);
P5=sortrows(P5,12);
P6=sortrows(P6,12);
P7=sortrows(P7,12);
P8=sortrows(P8,12);
P9=sortrows(P9,12);
```

```
for pp=1:1:9
```

```
    if pp==1
        d=P1;
        size(d);
        pp;
    elseif pp==2
        d=P2;
        pp;
    elseif pp==3
        d=P3;
```

```

        pp;
elseif pp==4
    d=P4;
    pp;
elseif pp==5
    d=P5;
    pp;
elseif pp==6
    d=P6;
    pp;
elseif pp==7
    d=P7;
    pp;
elseif pp==8
    d=P8;
    pp;
elseif pp==9
    d=P9;
    pp;
end

file(1440,134)=0;
flag(31,24,60)=0; % a vérifier
[L,C]=size(d);
flag(1)=0;
duree=0;
f=0;
time=0;
colonne_AC1138=85;

for i=1:1:L

    if (i>1)% && (d(i,12)~=d(prec,12)) % le check sur la colonne 12 n'est pas bon car on retire alors les chargements
simultanés
        if d(i,1)==mois
            if d(i,2)==jour
%-----
                for q=1:1:1440
                    % boucle sur les minutes/heures de la journée
% =====
                    % incrément positif

                    if(d(i,3)==tp(q,1) && d(i,4)==tp(q,2)) % check sur la date pour avoir la ligne sur les 1440

                        % pour chaque produit, regarder le produit
                        % actuellement en charge

                        qq=1;
                        prod=compo(qq,3);
                        colonne_quai=compo(qq,4)+67;

                        while(d(i,7)~=tp(qq,1) || d(i,8)~=tp(qq,2))

```

```

% check sur l'heure de fin
qq=qq+1;
end

duree=qq-q;

% calculer la durée : faire le lien avec
% les q de arrivée et départ. La différence
% donne la durée. Ensuite faire une boucle
% sur l'incrémement (avec maximim
% 23h59). Faire une boucle pour déterminer
% cela.

y=1;
prod=compo(y,3);
colonne_quai=compo(y,4)+67;

%           if(d(i,9)~=compo(21,1) || d(i,10)~=compo(21,2) )
%           d(i,:)
%           end

while d(i,9)~=compo(y,1) || d(i,10)~=compo(y,2)
    y=y+1;

    if(d(i,9)==compo(22,1) && d(i,10)==compo(22,2))
        y=22;

    end

%           if(d(i,9)==compo(21,1) && d(i,10)==compo(21,2))
%           y=17;
%           end

%           % Pb Q3.6
%           if(d(i,9)==compo(21,1) && d(i,10)==compo(21,2))
%           prod=compo(y,3);
%           colonne_quai=compo(y,4)+67;
%           %OK
%           end

%           prod=compo(y,3);
%           colonne_quai=compo(y,4)+67;
end
colonne_prod=prod;

if d(i,11)==0
    file(q,colonne_prod)=file(q,colonne_prod)+1;
    colonne_bras=y+10;
    file(q,colonne_bras)=file(q,colonne_bras)+1;
    file(q,colonne_quai)=file(q,colonne_quai)+1;

%           if (d(i,1)==3 && d(i,2)==9 && y==4 )

```

```

%           d(i,:)
%           end

if duree ~= 0
    for tm=1:1:(duree)

        tmi=q+tm;
        if tmi<=1440
            file(tmi,colonne_prod)=file(tmi,colonne_prod)+1;
            file(tmi,colonne_bras)=file(tmi,colonne_bras)+1;
            file(tmi,colonne_quai)=file(tmi,colonne_quai)+1;
            if (d(i,1)==3 && d(i,2)==9 && (tmi==619 || q==619) )
                fprintf('-----');
                d(i,:)
                file(tmi,colonne_prod)
            end
        end
    end
end
end

%           if file(q,colonne_bras)>=2
%           mois
%           jour
%           colonne_bras
%           q
%           d(i,:)
%           end

end

if d(i,11)==compo(y,6) && d(i,11)~=0
    fprintf('test');
    file(q,colonne_AC1138)=file(q,colonne_AC1138)+1;
    colonne_Qb_AC1138=y+85;
    file(q,colonne_Qb_AC1138)=file(q,colonne_Qb_AC1138)+1;
    colonne_quai_AC1138=compo(y,4)+111;
    file(q,colonne_quai_AC1138)=file(q,colonne_quai_AC1138)+1;

    if duree ~= 0
        for tm=1:1:(duree)

            tmi=q+tm;
            if tmi<=1440
                file(tmi,colonne_AC1138)=file(tmi,colonne_AC1138)+1;
                file(tmi,colonne_Qb_AC1138)=file(tmi,colonne_Qb_AC1138)+1;
                file(tmi,colonne_quai_AC1138)=file(tmi,colonne_quai_AC1138)+1;

            end
        end
    end
end

```

```

        end
    end

end

% =====
% incrément négatif
    if(d(i,7)==tp(q,1) && d(i,8)==tp(q,2))

        % pour chaque produit, regarder le produit
        % dont le chargement est fini
        y=1;
        prod=compo(y,3);
        colonne_quai=compo(y,4)+67;

%         if(d(i,9)~=compo(21,1) || d(i,10)~=compo(21,2) )
%             d(i,:);
%         end

        while d(i,9)~=compo(y,1) || d(i,10)~=compo(y,2)
            y=y+1;

            if(d(i,9)==compo(22,1) && d(i,10)==compo(22,2))
                y=22;

            end

            if y>=58
                fprintf('\n Y>=58 ');
                y
            end
            prod=compo(y,3);
            colonne_quai=compo(y,4)+67;
        end

        colonne_prod=prod;

        if (d(i,11)==0 )
            file(q,colonne_prod)=file(q,colonne_prod)-1;
            colonne_bras=y+10;
            file(q,colonne_bras)=file(q,colonne_bras)-1;
            file(q,colonne_quai)=file(q,colonne_quai)-1;

        end

        if (d(i,11)~=0 && d(i,11)==compo(y,6) )
            file(q,colonne_AC1138)=file(q,colonne_AC1138)-1;
            colonne_Qb_AC1138=y+85;
            file(q,colonne_Qb_AC1138)=file(q,colonne_Qb_AC1138)-1;

        end
    end
end

%-----

end

```

```

        end

    end

end

end

end

end

% if pp==9
%   d(1,:)
% end
%

d=0;

if pp==9
% fprintf('Toutes les données sont passées \n');
end

end

```

7. *Chgtsimultanes* - Extraction du nombre de chargements simultanés pour le G3 et GD

```
function [ m ] = Chgtsimultanes( cgt_sim )
% Calcule le nombre de chargements simultanés pour GD et G3
% Il faut introduire les données cgt_sim et ça donne le nombre de
% chargements simultanés par produit et en nombre de minutes.

tic

n_GD1=0;
n_GD2=0;
n_G31=0;
n_G32=0;

j=0;

for i=1:1:1440
    for c=0:1:180
        n_GD1=2*c+1;
        n_GD2=2*c+2;
        n_G31=2*c+3;
        n_G32=2*c+4;

        j=round(n_GD1/4)+1;
        %c'est le jour

        % i = la minute ; j = l'heure
        tmp(i,j,1)=cgt_sim(i,n_GD1)+cgt_sim(i,n_GD2);
        tmp(i,j,2)=cgt_sim(i,n_G31)+cgt_sim(i,n_G32);

    end

end

end

% fprintf('max1 = ');
% max(tmp(:,1))
% fprintf('/n ');
% fprintf('max2 = ');
% max(tmp(:,2))
% fprintf('/n ');
%
m(2,16)=0;
% somme des minutes où on a de 1 à 15 chargements simultanés
for t=1:j
    for i=1:1:1440
        %--
        if tmp(i,t,1)==0
            m(1,16)=m(1,16)+1;
        end
        if tmp(i,t,1)==1
```

```

        m(1,1)=m(1,1)+1;
    end
    if tmp(i,t,1)==2
        m(1,2)=m(1,2)+1;
    end
    if tmp(i,t,1)==3
        m(1,3)=m(1,3)+1;
    end
    if tmp(i,t,1)==4
        m(1,4)=m(1,4)+1;
    end
    if tmp(i,t,1)==5
        m(1,5)=m(1,5)+1;
    end
    if tmp(i,t,1)==6
        m(1,6)=m(1,6)+1;
    end
    if tmp(i,t,1)==7
        m(1,7)=m(1,7)+1;
    end
    if tmp(i,t,1)==8
        m(1,8)=m(1,8)+1;
    end
    if tmp(i,t,1)==9
        m(1,9)=m(1,9)+1;
    end
    if tmp(i,t,1)==10
        m(1,10)=m(1,10)+1;
    end
    if tmp(i,t,1)==11
        m(1,11)=m(1,11)+1;
    end
    if tmp(i,t,1)==12
        m(1,12)=m(1,12)+1;
    end
    if tmp(i,t,1)==13
        m(1,13)=m(1,13)+1;
    end
    if tmp(i,t,1)==14
        m(1,14)=m(1,14)+1;
    end
    if tmp(i,t,1)==15
        m(1,15)=m(1,15)+1;
    end

%--
    if tmp(i,t,2)==0
        m(2,16)=m(2,16)+1;
    end
    if tmp(i,t,2)==1
        m(2,1)=m(2,1)+1;
    end
    if tmp(i,t,2)==2
        m(2,2)=m(2,2)+1;

```

```

end
if tmp(i,t,2)==3
    m(2,3)=m(2,3)+1;
end
if tmp(i,t,2)==4
    m(2,4)=m(2,4)+1;
end
if tmp(i,t,2)==5
    m(2,5)=m(2,5)+1;
end
if tmp(i,t,2)==6
    m(2,6)=m(2,6)+1;
end
if tmp(i,t,2)==7
    m(2,7)=m(2,7)+1;
end
if tmp(i,t,2)==8
    m(2,8)=m(2,8)+1;
end
if tmp(i,t,2)==9
    m(2,9)=m(2,9)+1;
end
if tmp(i,t,2)==10
    m(2,10)=m(2,10)+1;
end
if tmp(i,t,2)==11
    m(2,11)=m(2,11)+1;
end
if tmp(i,t,2)==12
    m(2,12)=m(2,12)+1;
end
if tmp(i,t,2)==13
    m(2,13)=m(2,13)+1;
end
if tmp(i,t,2)==14
    m(2,14)=m(2,14)+1;
end
if tmp(i,t,2)==15
    m(2,15)=m(2,15)+1;
end

end

end

m=m';

toc

```

8. *Cparli* – Calcul du coût par litre

```
function [CL, CL1, CL2, CL3] = Cparli( c1,c2,c3,s1,s2,s3 )
% calcul du coût par litre par bras et remise en format colonne
% 4 arguments de sortie
%
for i=1:57

    CL(i,1) = c1(i)/s1(i);
    CL1(i) = c1(i)/s1(i);

end

for i=1:57

    CL(i,2) = c2(i)/s2(i);
    CL2(i) = c2(i)/s2(i);

end

for i=1:57

    CL(i,3) = c3(i)/s3(i);
    CL3(i) = c3(i)/s3(i);

end

CL1=CL1';
CL2=CL2';
CL3=CL3';
```

9. *lit_produit* – Calcul des litrages par produit

```
function [ P ] = lit_produit( S1,Compo )
%Détermine les litrages par produits pour une année
%Sort pour chaque produit le litrage

% P=0;
j=0;
i=0;

S1;
Compo;
P(1:12)=0;
for j=1:11

    for i=1:57
        if Compo(i,3)== j
            %a=P(j);
            %b=a+S1(i);
            i
            j
            P(j)
            S1(i)
            P(j)=P(j)+ S1(i);
        end
    end
end

%regroupement Extra et GD peut-être nécessaire

for j=1:11

    for i=1:57
        if Compo(i,3)== 1 || Compo(i,3)==7
            %a=P(j); b=a+S1(i); P(j)= b;
            P(12)=P(12)+S1(i);
        end
    end
end

P=P';
```

10. Moyenneentre – Calcul des entrées sur site

```
function [ moyenne, suivi] = Moyenneentre()
%Détermine le nombre d'entrées et de sorties sur un jour
%Fait une statistique sur la période

tic

daily(30)=0;
moyenne=0;
suivi=0;

% -----
% Janvier
for j=1:1:31

    if j==7||j==8||j==14||j==15||j==21||j==22||j==28||j==29
        Vflag2(j)=0;

    else
        %j
        [ Vflag2 ] = nbvisites(1, j );
        daily(j)=sum(Vflag2);
        moyenne=moyenne+sum(Vflag2);
    end

end

[Cj,lj] = max(daily);

suivi_j=daily;
clear daily;
clear Vflag2;
% -----
% Février
daily(29)=0;
for j=1:1:29

    if j==4||j==5||j==11||j==12||j==18||j==19||j==25||j==26
        Vflag2(j)=0;

    else
        %j
        [ Vflag2 ] = nbvisites(2, j );
        daily(j)=sum(Vflag2);
        moyenne=moyenne+sum(Vflag2);
    end

end

[Cf,lf] = max(daily);

suivi_f=daily;
clear daily;
```

```

clear Vflag2;

% -----
% Mars
daily(12)=0;
for j=1:1:12

    if j==2||j==3||j==10||j==11||j==17||j==18||j==24||j==25||j==31
        Vflag2(j)=0;
    else
        %j
        [ Vflag2 ] = nbvisites(3, j );
        daily(j)=sum(Vflag2);
        moyenne=moyenne+sum(Vflag2);
    end
end
suivi_m=daily;

[Cm,Im] = max(daily);

suivi=[suivi_j suivi_f suivi_m];
plot(suivi);
title('Evolution du nombre de camions chargés');
moyenne=moyenne/(31+29+12-25);
moyenne

Cj
lj
Cf
lf
Cm
lm

fprintf('Fin \n');
toc

```

11. *Moyenneentre_WE* – Calcul des entrées sur site sans les weekends

```
function [ suivi ] = Moyenneentre_WE()
%Détermine le nombre d'entrées et de sorties sur un jour du 02/01 au 12/03
%Fait une statistique sur la période
%Sans les WE!!

% [ suivi, sig ] = Moyenneentre_WE() pour la variance de 5h à 7h

tic

daily(30)=0;
moyenne=0;
suivi=0;

% -----
% Janvier
for j=1:1:31

    %j
    [ Vflag2 ] = nbvisites(1, j );
    % nbvisites(1,j) où 1 est le mois : ici janvier!
    daily(j)=sum(Vflag2);
    moyenne=moyenne+sum(Vflag2);
%     sig(j,1)=Vflag2(11);
%     sig(j,2)=Vflag2(12);
%     sig(j,3)=Vflag2(13);
%     sig(j,4)=Vflag2(14);
end

[Cj,lj] = max(daily);

suivi_j=daily;
clear daily;
clear Vflag2;
% -----
% Février
daily(29)=0;
for j=1:1:29

    %j
    [ Vflag2 ] = nbvisites(2, j );
    daily(j)=sum(Vflag2);
    moyenne=moyenne+sum(Vflag2);
%     l=j+31;
%     sig(l,1)=Vflag2(11);
%     sig(l,2)=Vflag2(12);
%     sig(l,3)=Vflag2(13);
%     sig(l,4)=Vflag2(14);
```

```

end

[Cf,lf] = max(daily);

suivi_f=daily;
clear daily;

clear Vflag2;

% -----
% Mars
daily(12)=0;

tttt=31;

for j=1:1:tttt

    %j
    [ Vflag2 ] = nbvisites(3, j );
    daily(j)=sum(Vflag2);
    moyenne=moyenne+sum(Vflag2);
%     l=j+31+29;
%     sig(l,1)=Vflag2(11);
%     sig(l,2)=Vflag2(12);
%     sig(l,3)=Vflag2(13);
%     sig(l,4)=Vflag2(14);

end
suivi_m=daily;

[Cm,lm] = max(daily);

figure(3);
suivi=[suivi_j suivi_f suivi_m];
bar(suivi);
title('Evolution du nombre de camions chargés');
moyenne1=moyenne/(31+29+31-26);
moyenne1
moyenne2=moyenne/(31+29+31);
moyenne2

fprintf('Fin \n');
toc

```

12. *Moyenneentre_WE_AC1138* – Calcul des entrées sur site sans les weekends configurable pour un produit particulier (ou plusieurs)

```
function [daily_moyenne_1, suivi] = Moyenneentre_WE_AC1138()
%Détermine le nombre d'entrées et de sorties sur un jour du 02/01 au 12/03
%Fait une statistique sur la période
% [ daily_moyenne_1 ] = Moyenneentre_WE_prod() => pour la moyenne
% [ suivi ] = Moyenneentre_WE_prod() => pour l'évolution sur les 92 jours
```

```
tic
```

```
daily(30)=0;
moyenne=0;
suivi=0;
daily1_moyenne(48)=0;
daily2_moyenne(48)=0;
daily3_moyenne(48)=0;
% -----
% Janvier
    for j=1:1:31

        %j
        [ Vflag2, total2 ] = nbvisites_AC1138(1, j );
        % nbvisites_AC1138(1,j) où 1 est le mois : ici janvier!
        daily(j)=sum(Vflag2);
        moyenne=moyenne+sum(Vflag2);
        daily1_moyenne(:)=daily1_moyenne(:)+(Vflag2(:));
```

```
end
```

```
[Cj,lj] = max(daily);
```

```
suivi_j=daily;
clear daily;
clear Vflag2;
clear total2;
% -----
% Février
daily(29)=0;
for j=1:1:29
```

```
    %j
    [ Vflag2, total2 ] = nbvisites_AC1138(2, j );
    daily(j)=sum(Vflag2);
    moyenne=moyenne+sum(Vflag2);
    daily2_moyenne(:)=daily2_moyenne(:)+(Vflag2(:));
```

```
end
```

```
[Cf,lf] = max(daily);
```

```

suivi_f=daily;
clear daily;

clear Vflag2;
clear total2;

% -----
% Mars
daily(12)=0;
for j=1:1:31

    %j
    [ Vflag2, total2 ] = nbvisites_AC1138(3, j );
    daily(j)=sum(Vflag2);
    moyenne=moyenne+sum(Vflag2);
    daily3_moyenne(:)=daily3_moyenne:)+(Vflag2);

end
suivi_m=daily;

[Cm,lm] = max(daily);

suivi=[suivi_j suivi_f suivi_m];
figure(1) ;
bar(suivi);
total_trimestre = sum(suivi);
title(['Evolution du nombre de camions chargés ',num2str(total_trimestre) ]);
moyenne1=moyenne/(31+29+31-26);
moyenne1
moyenne2=moyenne/(31+29+31);
moyenne2

daily_moyenne_0=[daily1_moyenne daily2_moyenne daily3_moyenne];
daily_moyenne_1=(daily1_moyenne + daily2_moyenne + daily3_moyenne)/(31+29+31-26);
daily_moyenne=daily_moyenne_0/(31+29+31-26)
figure(2);
title('Nombre moyen 3 X 1mois d entree moyenne par demi-heure ');
bar(daily_moyenne);
sum(daily_moyenne)
figure(3);
bar(daily_moyenne_1); % moyenne des heures d'entrée par demi-heure sur les 3 mois
daily_moyenne_1=daily_moyenne_1';

% fprintf(fid,'Nombre max de camions en janvier : %6f constaté le %6f \n', 2,4);
% Cj
% lj
% Cf
% lf
% Cm
% lm

```

```
suivi=suivi';
```

```
fprintf('Fin \n');
```

```
toc
```

13. *nb_bottom_bras* – Calcul chargements de diesel avec un ou deux bras (durant la même visite)

```
function [ flag ] = nb_bottom_bras( )  
% calcule le nombre de chargements avec un ou deux bras de diesel
```

```
global P1 P2 P3 P4 P5 P6 P7 P8 P9  
global compo_Quais_bras
```

```
compo=compo_Quais_bras;  
[L,C]=size(P1);  
size(compo_Quais_bras);
```

```
% Tri selon le numéro de visite (pour faire les tests sur une visite)
```

```
P1=sortrows(P1,12);  
P2=sortrows(P2,12);  
P3=sortrows(P3,12);  
P4=sortrows(P4,12);  
P5=sortrows(P5,12);  
P6=sortrows(P6,12);  
P7=sortrows(P7,12);  
P8=sortrows(P8,12);  
P9=sortrows(P9,12);
```

```
for pp=1:1:9
```

```
    if pp==1  
        d=P1;  
        size(d);  
        pp;  
    elseif pp==2  
        d=P2;  
        pp;  
    elseif pp==3  
        d=P3;  
        pp;  
    elseif pp==4  
        d=P4;  
        pp;  
    elseif pp==5  
        d=P5;  
        pp;  
    elseif pp==6  
        d=P6;  
        pp;  
    elseif pp==7  
        d=P7;  
        pp;  
    elseif pp==8  
        d=P8;  
        pp;
```

```

elseif pp==9
    d=P9;
    pp;
end

[L,C]=size(d);

visites = [ P1(:,12); P2(:,12); P3(:,12); P4(:,12); P5(:,12); P6(:,12); P7(:,12); P8(:,12); P9(:,12); ];
visites=unique(visites);
[Lv,Cv]=size(visites);
flag(Lv,6)=0;

for i=1:1:L % boucle sur les visites
    for k=1:1:Lv
        if (visites(k)==d(i,12))

            if ( (d(i,9)== 1 && d(i,10)== 1 && d(i,11)==0))
                flag(k,1)=flag(k,1)+1;
            end
            if ( (d(i,9)== 1 && d(i,10)== 6 && d(i,11)==0))
                flag(k,2)=flag(k,2)+1;
            end

            if ( (d(i,9)== 2 && d(i,10)== 1 && d(i,11)==0))
                flag(k,3)=flag(k,3)+1;
            end
            if ( (d(i,9)== 2 && d(i,10)== 6 && d(i,11)==0))
                flag(k,4)=flag(k,4)+1;
            end

            if ( (d(i,9)== 4 && d(i,10)== 1 && d(i,11)==0))
                flag(k,5)=flag(k,5)+1;
            end
            if ( (d(i,9)== 4 && d(i,10)== 5 && d(i,11)==0))
                flag(k,6)=flag(k,6)+1;
            end
        end
    end

    end %for k
end %for i

if pp==9
    fprintf('Toutes les données sont passées \n');
end

end %for P1 etc...

cgt_q(Lv,6)=0;
for i=1:1:Lv
    if flag (i,1)>=1 && flag(i,2)==0
        cgt_q(i,1)=1;
    end
end

```

```
end
if flag (i,1)==0 && flag(i,2)>=1
    cgt_q(i,2)=1;
end
```

14. *nbchgt* – Calcul du nombre de chargements par demi-heure

```
function [ flag, pres ] = nbchgt( mois, jour)
% calcule le nombre d'arrivée sur 30 min pour un bras (donc plusieurs bras
% par visite!)
% Demande fichier de donnée
% pres qui sort est en fait 48 lignes qui reprennent les 1/2h par 1/2h sur
% la journée
% Flag reprend les chargements effectués 30min par 30min
tic

global P1 P2 P3 P4 P5 P6 P7 P8 P9
global P1t P2t P3t P4t P5t P6t P7t P8t P9t
global compo_Quais_bras

% P1=sortrows(P1,12);
% P2=sortrows(P2,12);
% P3=sortrows(P3,12);
% P4=sortrows(P4,12);
% P5=sortrows(P5,12);
% P6=sortrows(P6,12);
% P7=sortrows(P7,12);
% P8=sortrows(P8,12);
% P9=sortrows(P9,12);

compo=compo_Quais_bras;
[L,C]=size(P1);
size(compo_Quais_bras)
flag(1)=0;
flag(48)=0;
pres(48,57)=0;

for pp=1:1:9

    if pp==1
        d=P1;
        size(d)
        pp
    elseif pp==2
        d=P2;
        pp
    elseif pp==3
        d=P3;
        pp
    elseif pp==4
        d=P4;
        pp
    elseif pp==5
        d=P5;
        pp
    elseif pp==6
        d=P6;
        pp
```

```

elseif pp==7
    d=P7;
    pp
elseif pp==8
    d=P8;
    pp
elseif pp==9
    d=P9;
    pp
end

[L,C]=size(d)
flag(1)=0;

for i=1:1:L
    if d(i,11)==0

        if d(i,1)==mois

            if d(i,2)==jour

                for h=0:1:23

                    if d(i,3)==h

                        if (d(i,4)>= 0) && (d(i,4)<=30)
                            y=2*h+1;
                            flag(y)=flag(y)+1;

                            % mise en mémoire du quai
                            q=1;
                            while d(i,9)~=compo(q,1) || d(i,10)~=compo(q,2)
                                q=q+1; % le problème vient du quai 1 B5 ???
                                if q==58
                                    fprintf('ouff');
                                    i
                                    d(i,13)
                                    d(i,14)
                                end
                                %
                                if (d(i,13)==1 || d(i,14)==5 ) %
                                %
                                mettre condition pour que la
                                %
                                boucle soit sorite en attribuant
                                %
                                au bras de 98 du Q1
                                %
                                break

                                end
                                pres(y,q)=pres(y,q)+1;

                                %
                                for q=1:1:57

```

```

%           if d(i,13)==compo(q,1)
%           if d(i,14)==compo(q,2)
%               pres(y,q)=pres(y,q)+1;
%           else
%               % s'assurer que l'on a le bon bras!
%               if d(i,14)==compo(q+1,2)
%                   pres(y,q+1)=pres(y,q+1)+1;

end
if (d(i,4)>30 ) && (d(i,4)<=60)
    y=2*h+2;
    flag(y)=flag(y)+1;

% mise en mémoire du quai
q=1;
while (d(i,9)~=compo(q,1)) || (d(i,10)~=compo(q,2))
    q=q+1;

    if q==58
        fprintf('ouff');
        i
        d(i,9)
        d(i,10)
        pp
    end

end

end
pres(y,q)=pres(y,q)+1;
end
end
end
end
end
end
end

d=0;
end

flag=flag';

figure(1);
bar(flag);
title(['Entrées sur site en date du ', num2str(jour),' - ', num2str(mois),' - 2012']);

toc

```

15. *Nbrcgt2Bb* – Calcul de la durée des chargements à deux ou plusieurs bras/quais

```
function [ report ] = Nbrcgt2Bb( evo )
%Détermine le nombre de chargement simultanés de GD sur un bottom ayant
%deux bras.
% le nombre indiqué est le nombre de minutes où autant de bras charges.
% Donc, on ne cumul pas. Pour avoir au moins 2 bras, il faut additionner 2
% bras, 3 bras et les suivants!!
```

```
tic
```

```
[L,C,H]=size(evo);
report(5,18)=0;
report(5,:)=0;
```

```
for i=1:1:L
```

```
    % boucle sur toutes les dates
```

```
        for l=1:1:1440
```

```
            % boucle sur les minutes
```

```
                for (k=1:1:5)
```

```
                    % boucle jusque 2 car le nombre de bras max en même temps est 2
```

```
                    % et un de sécurité
```

```
                        if((evo(i,l,13)+evo(i,l,18))==k )
```

```
                        % cas Q01 ou Q06
```

```
                            j=1;
```

```
                            report(k,j)=report(k,j)+1;
```

```
                            % ligne : utilisation 1 ou bras
```

```
                            % colonne : le cas considéré
```

```
%                            if ((evo(i,l,13)+evo(i,l,18))==3)
```

```
%                            evo(i,l,13)
```

```
%                            evo(i,l,18)
```

```
%                            i
```

```
%                            l
```

```
%                            end
```

```
                        end
```

```
%                        if((evo(i,l,13)+evo(i,l,18))==2 )
```

```
%                        % cas Q01 et Q06
```

```
%                        j=2;
```

```
%                        report(k,j)=report(k,j)+1;
```

```
%                        end
```

```
                        if((evo(i,l,19)+evo(i,l,23))==k )
```

```
%                        % cas Q11 ou Q16
```

```
                            j=2;
```

```
                            report(k,j)=report(k,j)+1;
```

```
                        end
```

```
%                        if((evo(i,l,19)+evo(i,l,23))==2 )
```

```
%                        % cas Q11 et Q16
```

```
%                        j=4;
```

```

%         report(k,j)=report(k,j)+1;
%     end
%     if((evo(i,l,24)+evo(i,l,28))==k)
%     % cas Q21 ou Q26
%         j=3;
%         report(k,j)=report(k,j)+1;
%     end
%     if((evo(i,l,24)+evo(i,l,28))==2 )
%     % cas Q21 et Q26
%         j=6;
%         report(k,j)=report(k,j)+1;
%     end
%     if((evo(i,l,29)+evo(i,l,33))==k )
%     % cas Q31 ou Q36
%         j=4;
%         report(k,j)=report(k,j)+1;
%     end
%     if((evo(i,l,29)+evo(i,l,33))==2 )
%     % cas Q31 et Q36
%         j=8;
%         report(k,j)=report(k,j)+1;
%     end
%     if((evo(i,l,34)+evo(i,l,38))==k )
%     % cas Q41 ou Q45
%         j=5;
%         report(k,j)=report(k,j)+1;
%     end
%     if((evo(i,l,34)+evo(i,l,38))==2 )
%     % cas Q41 et Q45
%         j=10;
%         report(k,j)=report(k,j)+1;
%     end
%     if((evo(i,l,19)+evo(i,l,23)+(evo(i,l,34)+evo(i,l,38)))==k )
%     % cas Q11 et Q14 et Q41 et Q45
%         j=6;
%         report(k,j)=report(k,j)+1;
%     end

%     if((evo(i,l,24)+evo(i,l,28)+(evo(i,l,29)+evo(i,l,32)))==k )
%     % cas Q21 et Q26 et Q31 et Q36
%         j=7;
%         report(k,j)=report(k,j)+1;
%     end

%     if((evo(i,l,19)>=1 || evo(i,l,23)>=1)&& (evo(i,l,34)>=1 || evo(i,l,38)>=1) )
%     % cas Q1 et Q4 en même tps si GD

%         k=1;
%         j=8;
%         report(k,j)=report(k,j)+1;
%     end

%     if((evo(i,l,13)>=1 || evo(i,l,18)>=1)&&(evo(i,l,19)>=1 || evo(i,l,23)>=1)&& (evo(i,l,34)>=1 || evo(i,l,38)>=1)&&
evo(i,l,73)==0 )

```

```

% cas Q0, Q1 et Q4 en même tps en GD mais Q3(totalité)
% libre pour voir si intéressant de mettre Excellium au
% Q2

    k=1;
    j=9;
    report(k,j)=report(k,j)+1;
end

%=====
%Alternative au quai 21

    if(evo(i,l,84)>=1 )
% cas Q11 et Q23 en même tps (

        k=1;
        j=10;
        report(k,j)=report(k,j)+1;
    end

    if((evo(i,l,85)>=1 ) )
% cas Q8 et Q21B2 en même tps (alternative au Q21) cas
% du G33 => minutes OK pour passer sur Q8

        k=1;
        j=11; % colonne
        report(k,j)=report(k,j)+1;
    end

    if((evo(i,l,86)>=1 ) )
% cas Q23-24

        k=1;
        j=12;
        report(k,j)=report(k,j)+1;
    end

    if((evo(i,l,78)>=1 ) )
% cas Q21B1 ou Q21B3 et Q11 en même tps (alternative au
% Q21) cas pour G3 et GD en même tps => minutes OK pour
% passer sur Q8

        k=1;
        j=13;
        report(k,j)=report(k,j)+1;
    end

    if((evo(i,l,79)>=1 ) )
% cas Q21B4 et Q23 ou Q25 en même tps (alternative au
% Q21) cas pour FOD => minutes HS pour passer sur Q8

        k=1;
        j=14;
        report(k,j)=report(k,j)+1;
    end

```

```

end

if((evo(i,l,64)>=1 )&& (evo(i,l,85)+evo(i,l,86)==0 ) )
    % cas Q21B4 et Q23 ou Q25 libres en même tps (alternative au
    % Q21) cas pour FOD => minutes OK pour passer sur Q8
    k=1;
    j=15;
    report(k,j)=report(k,j)+1;
end

if((evo(i,l,64)>=1 )&& (evo(i,l,85)+evo(i,l,86)==0 ) )
    % cas Q21B4 et Q23 ou Q25 libres en même tps (alternative au
    % Q21) cas pour FOD => minutes OK pour passer sur Q8
    k=1;
    j=16;
    report(k,j)=report(k,j)+1;
end

if((evo(i,l,75)>=1 ) )
    % cas Q8 nombre de minutes utilisées
    k=1;
    j=17;
    report(k,j)=report(k,j)+1;
end

if((evo(i,l,86)>=1 ) )
    % cas Q25-26 nombre de minutes utilisées

    k=1;
    j=18;
    report(k,j)=report(k,j)+1;
end

%          %cas Q11 nombre de minutes utilisées 8257
%          if((evo(i,l,78)>=1 ) )
%          k=1;
%          j=16;
%          report(k,j)=report(k,j)+1;
%          end
% %
%          if((evo(i,l,85)>=1 ) )
%          % cas Q23 nombre de minutes utilisées 4749min
%          k=1;
%          j=17;
%          report(k,j)=report(k,j)+1;
%          end

    end
end

end
fprintf('Fin \n');
toc

```

16. *Nbrcgtprod* – Calcul de la durée des chargements à deux ou plusieurs bras/quais (boucle)

```
function [ report ] = Nbrcgtprod( evo )
%Détermine le nombre de chargement simultanés au total à partir du fichier
%sorti par cgt_tot sur la période
% le nombre indiqué est le nombre de minutes où autant de bras charges.
% Donc, on ne cumul pas. Pour avoir au moins 2 bras, il faut additionner 2
% bras, 3 bras et les suivants!!

tic

[L,C,H]=size(evo);
report(14,10)=0;

for i=1:1:L
    % boucle sur toutes les dates

    for j=1:1:10
        jj=j+2;
        % boucle sur les 10 produits

        for l=1:1:1440
            % boucle sur les minutes

            for (k=1:1:14)
                % boucle jusque 14 car le nombre de bras max en même temps est
                % 10 donc un peu de marge pour vérifier les calculs

                if(evo(i,l,jj)==k)
                    report(k,j)=report(k,j)+1;

                end
            end
        end
    end

end
end
fprintf('Fin \n');
toc
```

17. nbvisites – Calcul du nombre de visites (pour une journée)

```
function [ flag ] = nbvisites( mois, jour)
% calcule le nombre d'arrivée sur 30 min sur site
% Demande fichier de donnée
% pres qui sort est en fait 48 lignes qui reprennent les 1/2h par 1/2h sur
% la journée
% Flag reprend les entrées 30min par 30min
% pres : reprend le premier chargement de la visite -> n'a pas de sens!!
%     function [ flag, pres ] = nbvisites( mois, jour)
%     [ flag,total ] = nbvisites( mois, jour)

global P1 P2 P3 P4 P5 P6 P7 P8 P9
global compo_Quais_bras

compo=compo_Quais_bras;
[L,C]=size(P1);
size(compo_Quais_bras);
flag(1)=0;
flag(48)=0;
pres(48,57)=0;
rr=0;

% Tri selon le numéro de visite (pour faire les tests sur une visite)
P1=sortrows(P1,12);
P2=sortrows(P2,12);
P3=sortrows(P3,12);
P4=sortrows(P4,12);
P5=sortrows(P5,12);
P6=sortrows(P6,12);
P7=sortrows(P7,12);
P8=sortrows(P8,12);
P9=sortrows(P9,12);

for pp=1:1:9

    if pp==1
        d=P1;
        size(d);
        pp;
    elseif pp==2
        d=P2;
        pp;
    elseif pp==3
        d=P3;
        pp;
    elseif pp==4
        d=P4;
        pp;
    elseif pp==5
        d=P5;
        pp;
    elseif pp==6
```

```

        d=P6;
        pp;
    elseif pp==7
        d=P7;
        pp;
    elseif pp==8
        d=P8;
        pp;
    elseif pp==9
        d=P9;
        pp;
    end

[L,C]=size(d);
flag(1)=0;

for i=1:1:L
    if d(i,11)==0 % s'assurer que l'on regarde produit pas additif
        prec=i-1;
        if (i>1) && (d(i,12)~=d(prec,12))
            if d(i,1)==mois

                if d(i,2)==jour

                    for h=0:1:23

                        if d(i,3)==h
%fprintf('ouff \n');
                            % ici on regarde par tranche de 30min
                            if (d(i,4)>= 0) && (d(i,4)<=30)
                                y=2*h+1;
                                flag(y)=flag(y)+1;

                                % mise en mémoire du quai
                                q=1;
                                while d(i,9)~=compo(q,1) || d(i,10)~=compo(q,2)
                                    q=q+1; % le problème vient du quai 1 B5 ???
                                    if q==58
                                        fprintf('ouff');
                                        i
                                        d(i,:)
                                        d(i,9)
                                        d(i,10)
                                    end
                                end
                                pres(y,q)=pres(y,q)+1;

                                %
                                for q=1:1:57
                                    %
                                    if d(i,13)==compo(q,1)
                                        %
                                        if d(i,14)==compo(q,2)
                                            %
                                            pres(y,q)=pres(y,q)+1;
                                        %
                                        else
                                            %
                                            % s'assurer que l'on a le bon bras!
                                            %
                                            if d(i,14)==compo(q+1,2)

```



```
flag=flag';

% figure(1);
total=sum(flag)
%
%
bar(flag);
title(['Entrées sur site en date du ', num2str(jour),' - ', num2str(mois),' - 2012 Nombre total d entrées : ', num2str(total)]);
```

18. nbvisites_AC1138 – Calcul du nombre de visites configurable pour un ou plusieurs bras/quais (pour une journée)

```
function [ flag,total ] = nbvisites_AC1138( mois, jour)
% calcule le nombre d'arrivée sur 30 min sur site pour AC1138 ou GD
% Demande fichier de donnée
% pres qui sort est en fait 48 lignes qui reprennent les 1/2h par 1/2h sur
% la journée
% Flag reprend les entrées 30min par 30min
% pres : reprend le premier chargement de la visite -> n'a pas de sens!!
%     function [ flag, pres ] = nbvisites( mois, jour)
```

```
% Attention : ici pour AC1138 donc vérifier condition!
```

```
global P1 P2 P3 P4 P5 P6 P7 P8 P9
global compo_Quais_bras
```

```
compo=compo_Quais_bras;
[L,C]=size(P1);
size(compo_Quais_bras);
flag(1)=0;
flag(48)=0;
pres(48,57)=0;
rr=0;
```

```
% Tri selon le numéro de visite (pour faire les tests sur une visite)
```

```
P1=sortrows(P1,12);
P2=sortrows(P2,12);
P3=sortrows(P3,12);
P4=sortrows(P4,12);
P5=sortrows(P5,12);
P6=sortrows(P6,12);
P7=sortrows(P7,12);
P8=sortrows(P8,12);
P9=sortrows(P9,12);
```

```
for pp=1:1:9
```

```
    if pp==1
        d=P1;
        size(d);
        pp;
    elseif pp==2
        d=P2;
        pp;
    elseif pp==3
        d=P3;
        pp;
    elseif pp==4
        d=P4;
        pp;
    elseif pp==5
        d=P5;
```

```

    pp;
elseif pp==6
    d=P6;
    pp;
elseif pp==7
    d=P7;
    pp;
elseif pp==8
    d=P8;
    pp;
elseif pp==9
    d=P9;
    pp;
end

[L,C]=size(d);
flag(1)=0;
quai=1; % quai à analyser
k=1;
ki=0;
v(1)=d(1,12);
for i=1:1:L

    tt=i-1;
    if (i>1 && d(i,12)~=d(tt,12))
        ki=1;
        end % test pour tenir compte de plusieurs produits sur la même visite!!

        %if ( (d(i,9)== 0 && d(i,11)==0) || (d(i,9)== 1 && d(i,11)==0) || (d(i,9)== 2 && d(i,11)==0) || (d(i,9)== 3 && d(i,11)==0) ||
        (d(i,9)== 4 && d(i,11)==0) ) % s'assurer que ce ne sont queles quais
            if ( (d(i,9)== quai && d(i,10)== 1 && d(i,11)==0) || (d(i,9)== quai && d(i,10)== 2 && d(i,11)==0) || (d(i,9)== quai &&
            d(i,10)== 3 && d(i,11)==0) || (d(i,9)== quai && d(i,10)== 4 && d(i,11)==0) || (d(i,9)== quai && d(i,10)== 5 && d(i,11)==0) ||
            (d(i,9)== quai && d(i,10)== 6 && d(i,11)==0) ) % s'assurer que les additifs
                %
                fprintf('test1 \n');
                %
                d(i,12)
                %
                v(k)
                if (i>1) && (ki==1) % ki==1 : on regarde si nouvelle visite
                    %if (i>1) && (d(i,12)~=v(k))
                    % fprintf('test2 \n');
                    ki=0;

                    if d(i,1)==mois

                        if d(i,2)==jour

                            for h=0:1:23

                                if d(i,3)==h
                                    %fprintf('ouff \n');
                                    % ici on regarde par tranche de 30min
                                    if (d(i,4)>= 0) && (d(i,4)<=30)
                                        y=2*h+1;
                                        flag(y)=flag(y)+1;
                                        d(i,:);

```



```

        end

    end

end

end

end

end
end
end

% if pp==9
%     d(1,:)
% end
%

d=0;

if pp==9
    fprintf('Toutes les données sont passées \n');
end

end

flag=flag';

figure(4);
total=sum(flag);
%
%
bar(flag);
title(['Entrées sur site en date du ', num2str(jour),' - ', num2str(mois),' - 2012 Nombre total d entrées : ', num2str(total),' sur le quai ', num2str(quai)]);

```

19. *pourcentagebrasutilisation* – Calcul l'utilisation relative d'un produit pour un bras déterminé

```
function [ Taux, stock ] = pourcentagebrasutilisation(Sorties, Compo, LparProduit )
%détermine le % de produit par ce bras pour l'année désignée
% divise les sorties par la quantité totale par ce bras
% objectif définir l'importance d'un bras pour un produit

for i=1:57
    Taux(i)=0;
    %pour chacun des bras
    % il faut voir le produit et comparer par rapport au total du produit
    for j=1:11
        if Compo(i,3)==j
            % i
            % j
            % Sorties(j)
            % LparProduit(j)
            Taux(i)=Sorties(i)/LparProduit(j);
            Taux(i);
        end
    end
end

% stockage en regroupant par produit
Sotck(1,1,1)=0;
for i=1:57
    for j=1:11
        if Compo(i,3)==j
            fprintf('i et j');
            %      i
            %      j
            stock(i,1)=i; % dans la première colonne : numéro du bras
            stock(i,2)=j; % dans la deuxième colonne : le produit
            stock(i,3)=Taux(i); % dans la troisième colonne : le taux

        end
    end
end

Taux=Taux';

%[] = figureprod(stock, an )
```

20. *Tot_calc_file_q* – Calcul du nombre de camions devant attendre avant de charger

```

function [ report ] = Tot_calc_file_q( evo )
% Détermine le nombre de fois où un camion est arrivé dans un quai dans les
% 8 minutes après la fin du chargement précédent.

global P1 P2 P3 P4 P5 P6 P7 P8 P9
global compo_Quais_bras
global tp dates
tic

report(2,19)=0;
report1(2,19)=0;

[L, C]=size(dates);

for i=1:1:L
    % boucle sur toutes les dates

    jour=dates(i,1)
    mois=dates(i,2)

    for q=70:1:86
        % boucle sur les quais
        qq=q-69;
        qq;
        for m=1:1:(1440-6)

            m1=m+1;
            m2=m+2;
            m3=m+3;
            m4=m+4;
            m5=m+5;
            m6=m+6;
            m7=m+7;
            m8=m+8;
            m9=m+9;
            m10=m+10;

            if (evo(i,m,q)==1 && (evo(i,m1,q)+evo(i,m2,q)+evo(i,m3,q)==0) )

%                compteur(i,qq,1)==compteur(i,qq,1)+1;
%                report(1,qq)=report(1,qq)+1;
%                fprintf('test1\n');

            if ((evo(i,m4,q)+evo(i,m5,q)+evo(i,m6,q)+evo(i,m7,q)+evo(i,m8,q))~=0)
%                compteur(i,qq,2)==compteur(i,qq,2)+1;
%                report(2,qq)=report(2,qq)+1;

            % -----
            % Condition : si fin de chargement avec Extra alors a attendu
            % sur le Q8!!

```

```

%      if ( (evo(i,m4,q)+evo(i,m5,q)+evo(i,m6,q)+evo(i,m7,q)+evo(i,m8,q))~=0)

% -----

% fprintf('\n test2\n');
end % if evo 2

end % if evo 1

end % m=1:1:(1440-6)
end % q=70:1:86

end % for i=1:1:L

fprintf('Fin \n');
toc

```

Annexe 4 : Evolution des chargements par quai

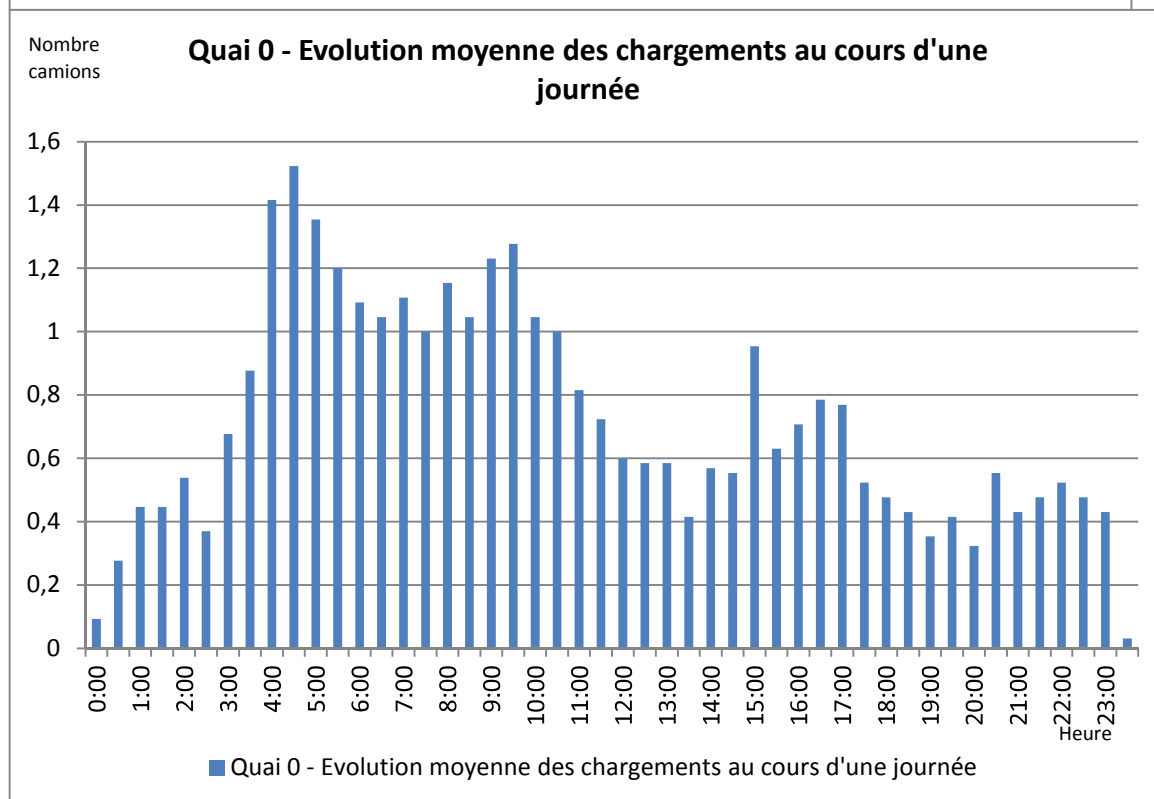
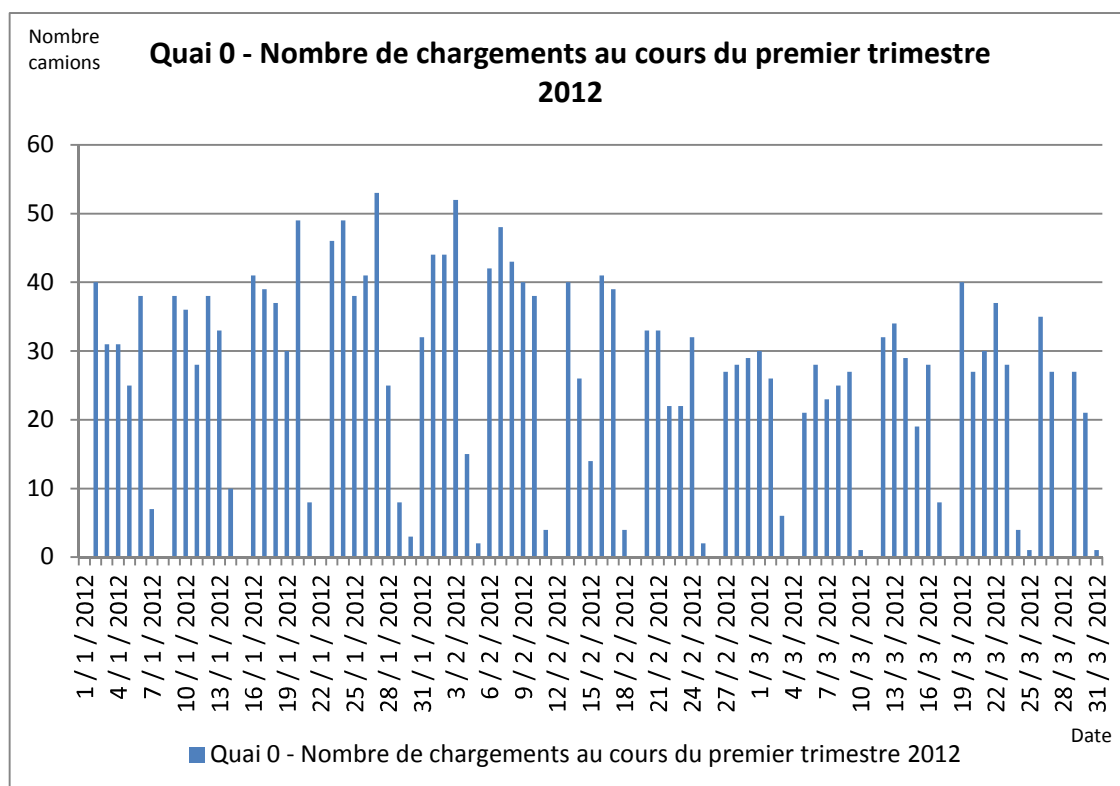
Cette annexe reprend pour tous les quais :

- L'évolution des chargements au cours du premier trimestre 2012,
- L'évolution moyenne des chargements journaliers, au cours de la journée (moyenne réalisée sur le premier trimestre 2012).

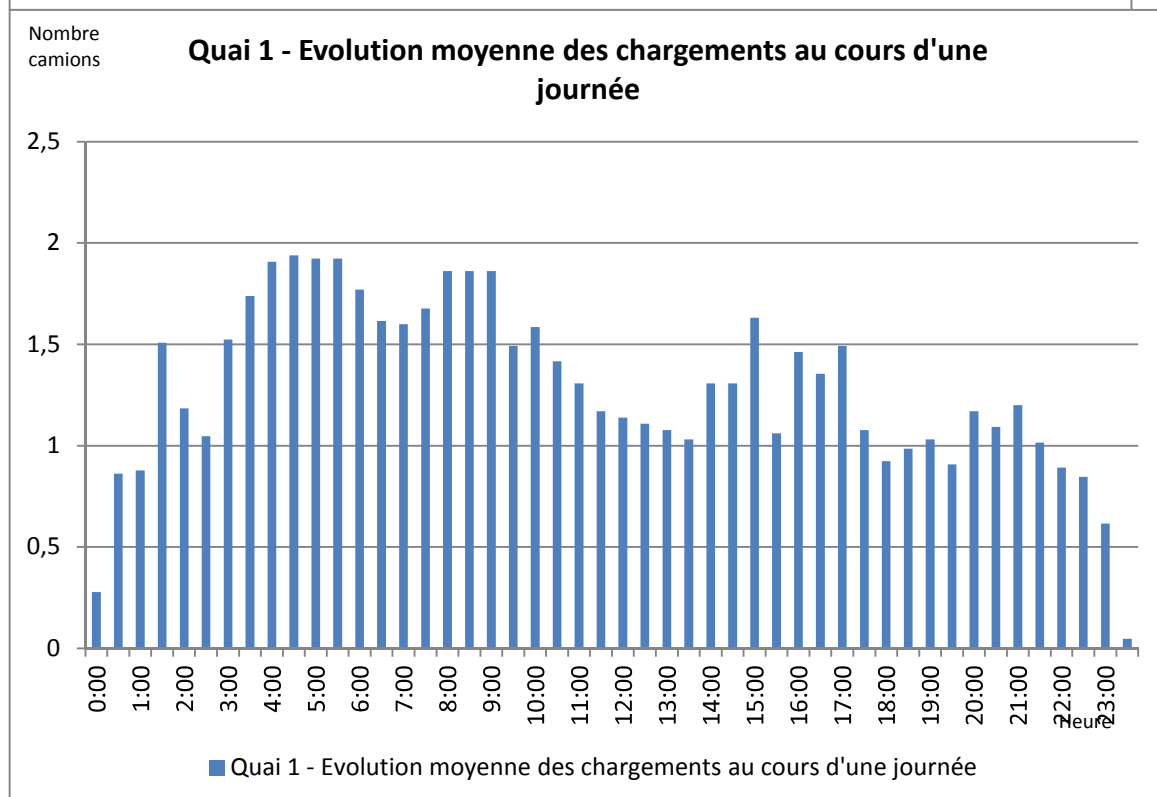
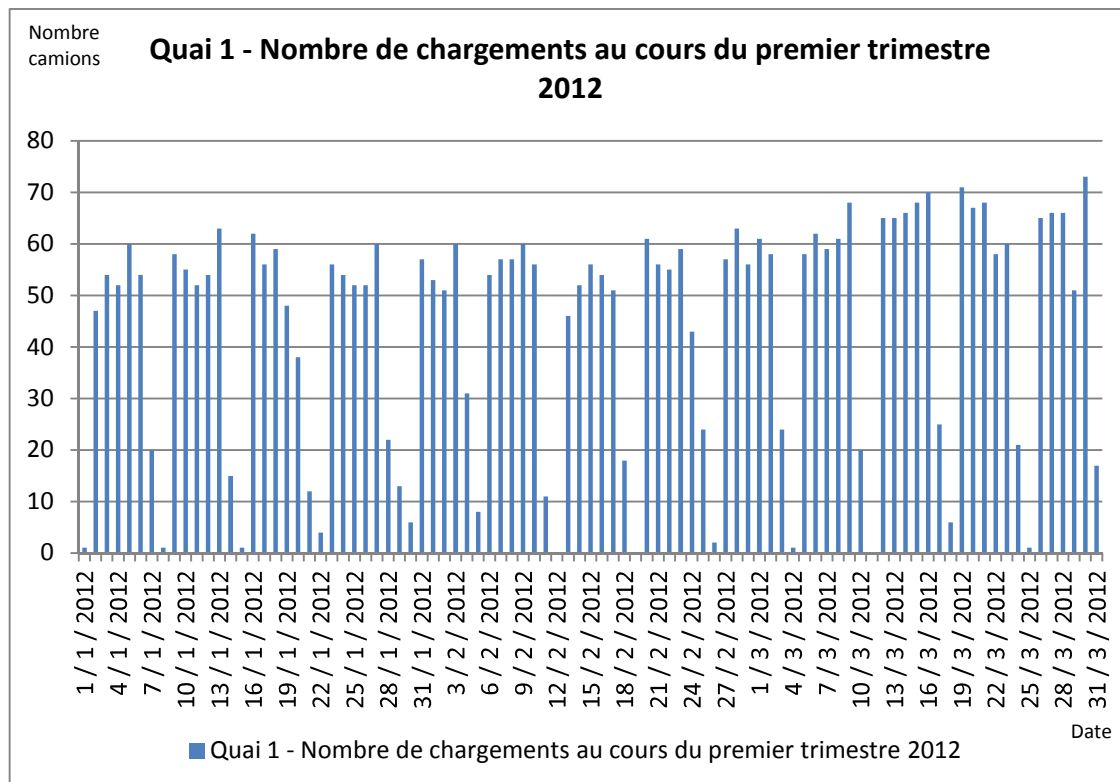
Table des matières

1. Quai 0	2
2. Quai 1	3
3. Quai 2	4
4. Quai 3	5
5. Quai 4	6
6. Quai 8	7
7. Quai 9	8
8. Quai 10	9
9. Quai 11	10
10. Quai 12	11
11. Quai 14	12
12. Quai 16	13
13. Quai 18	14
14. Quai 21-22	15
15. Quai 23-24	16
16. Quai 25-26	17

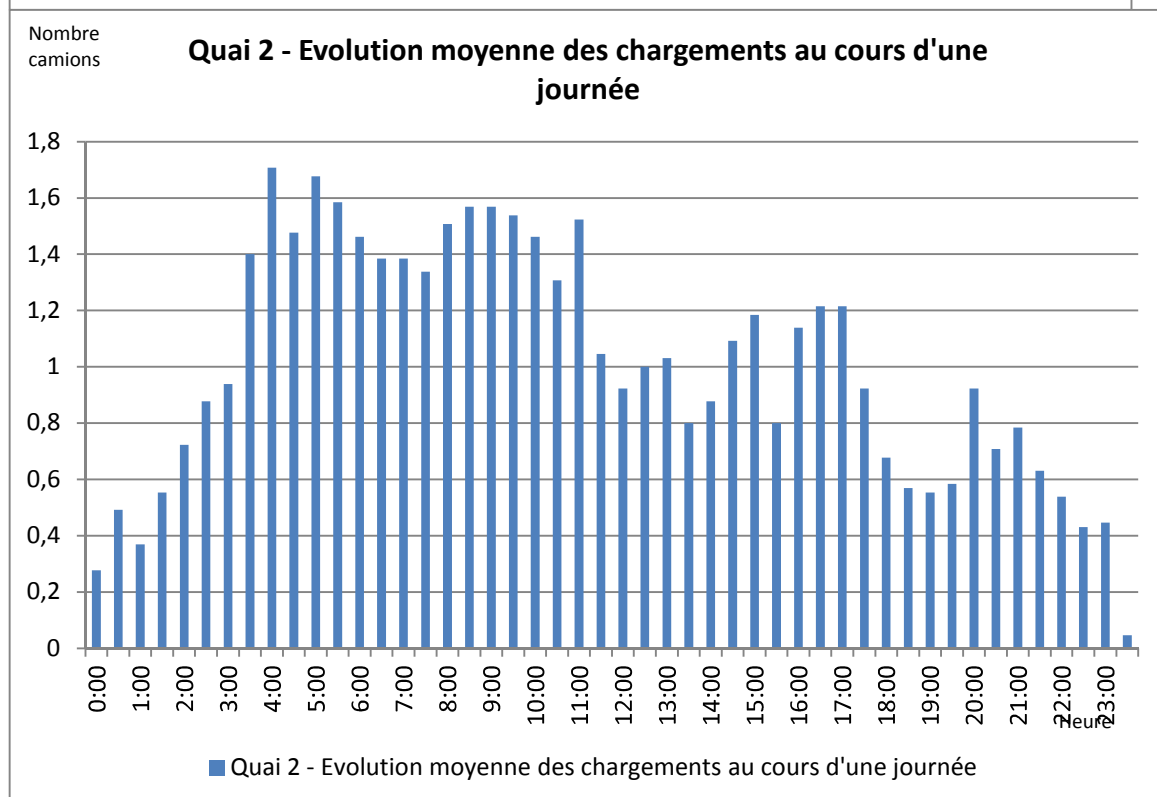
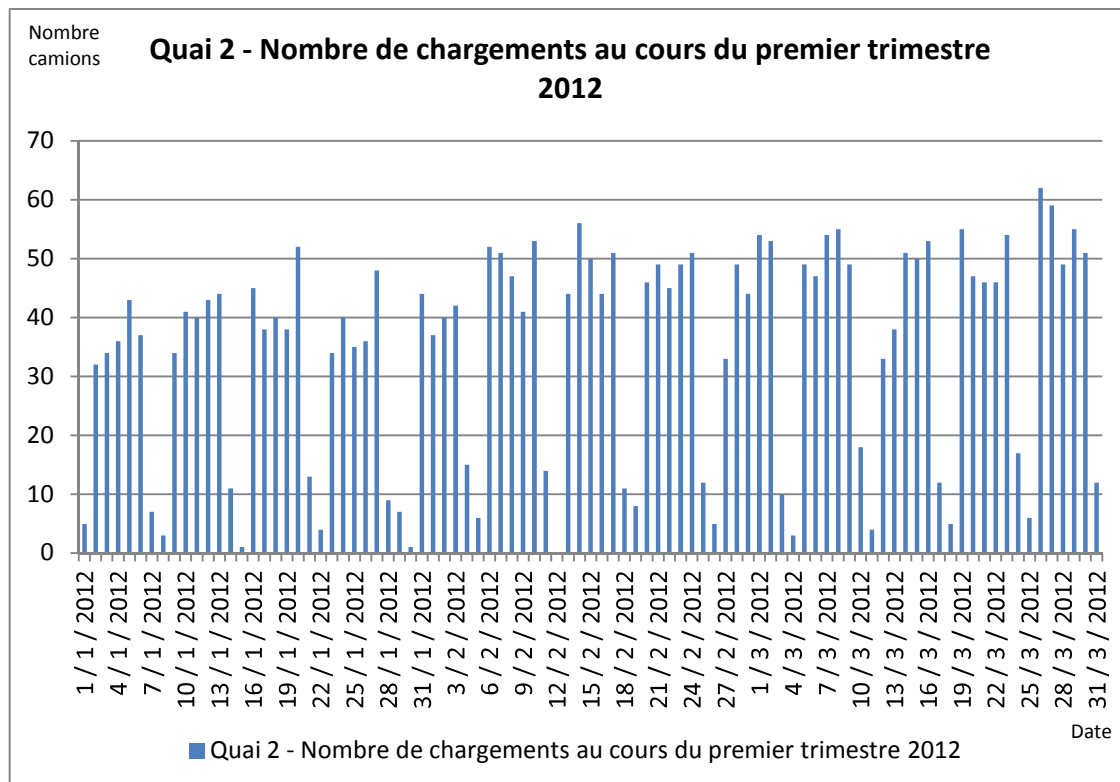
1. Quai 0



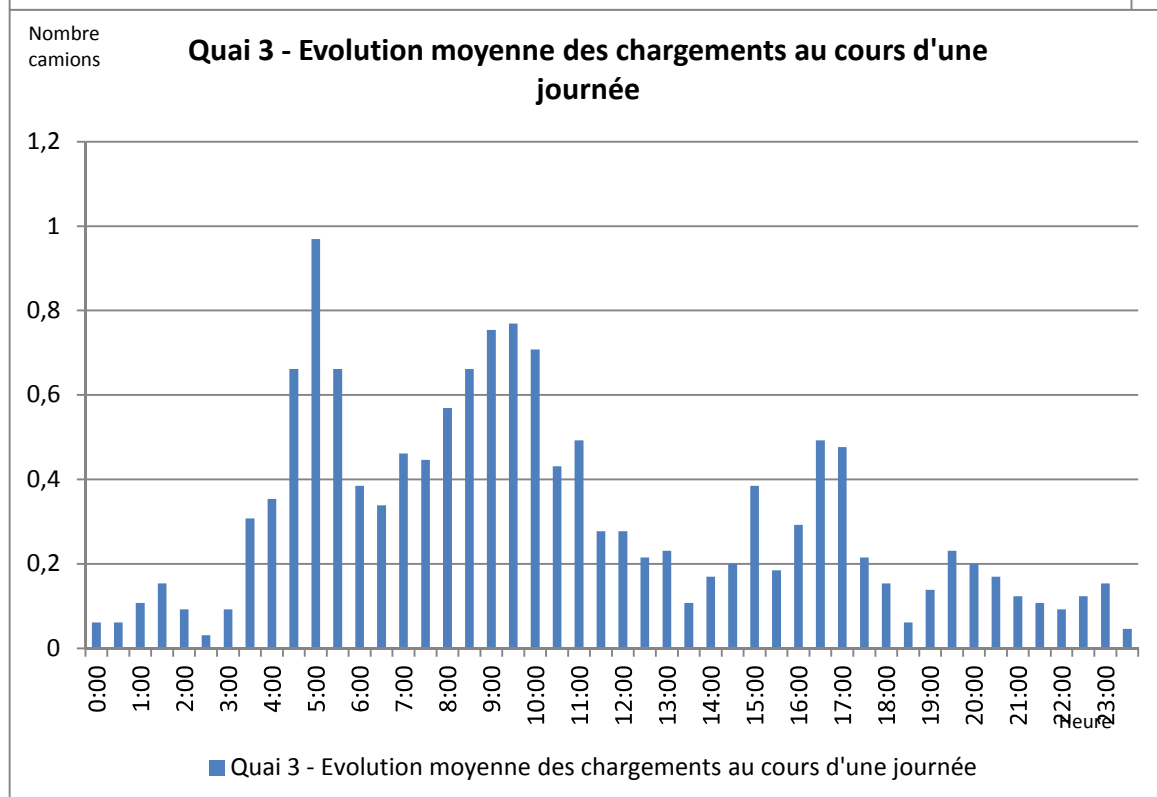
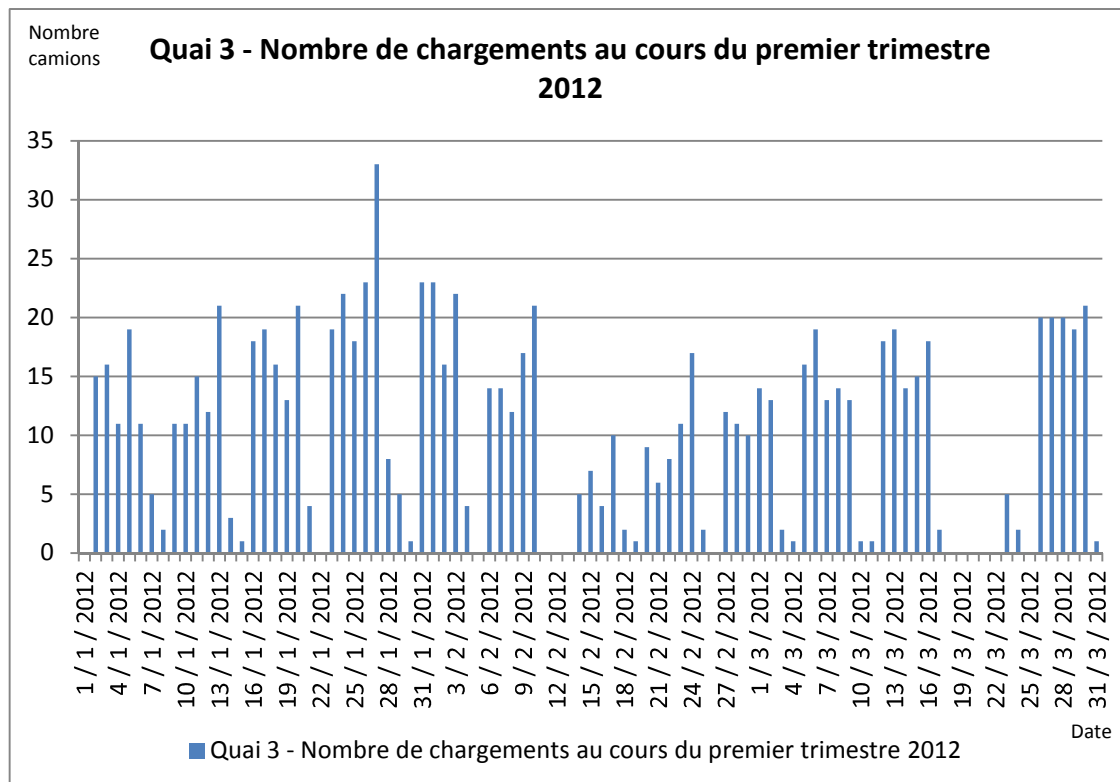
2. Quai 1



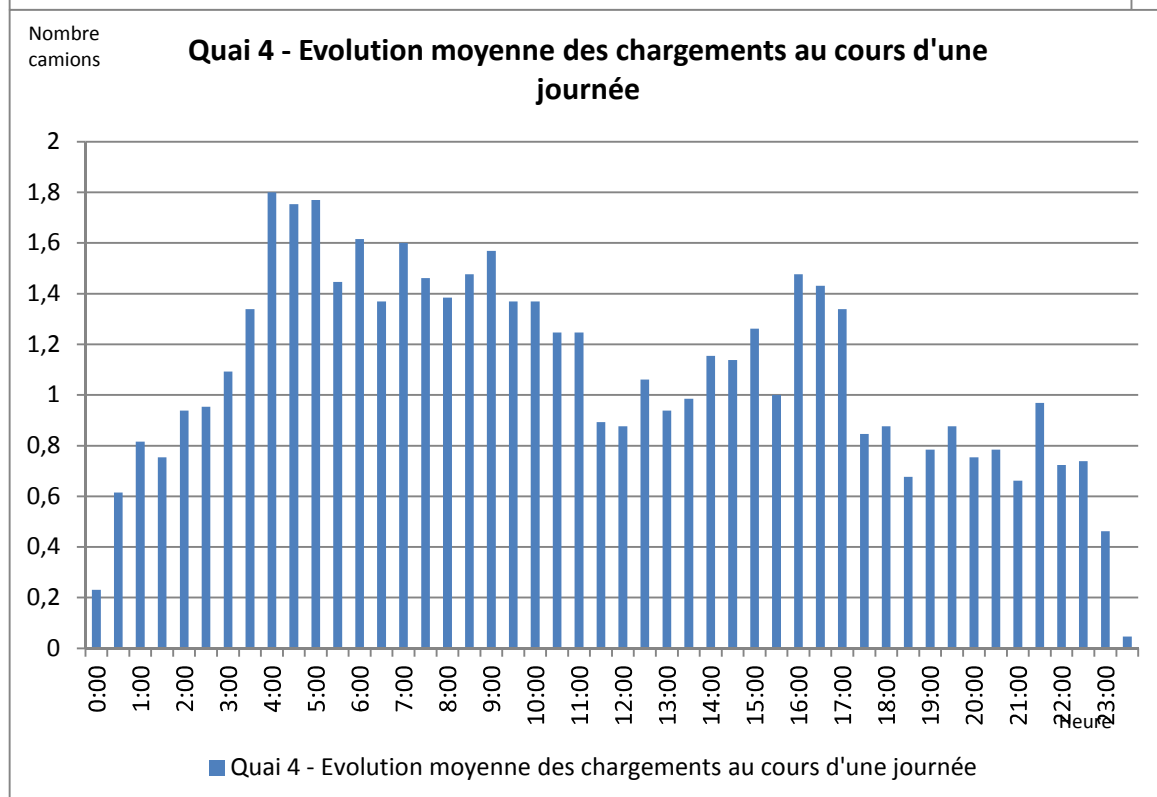
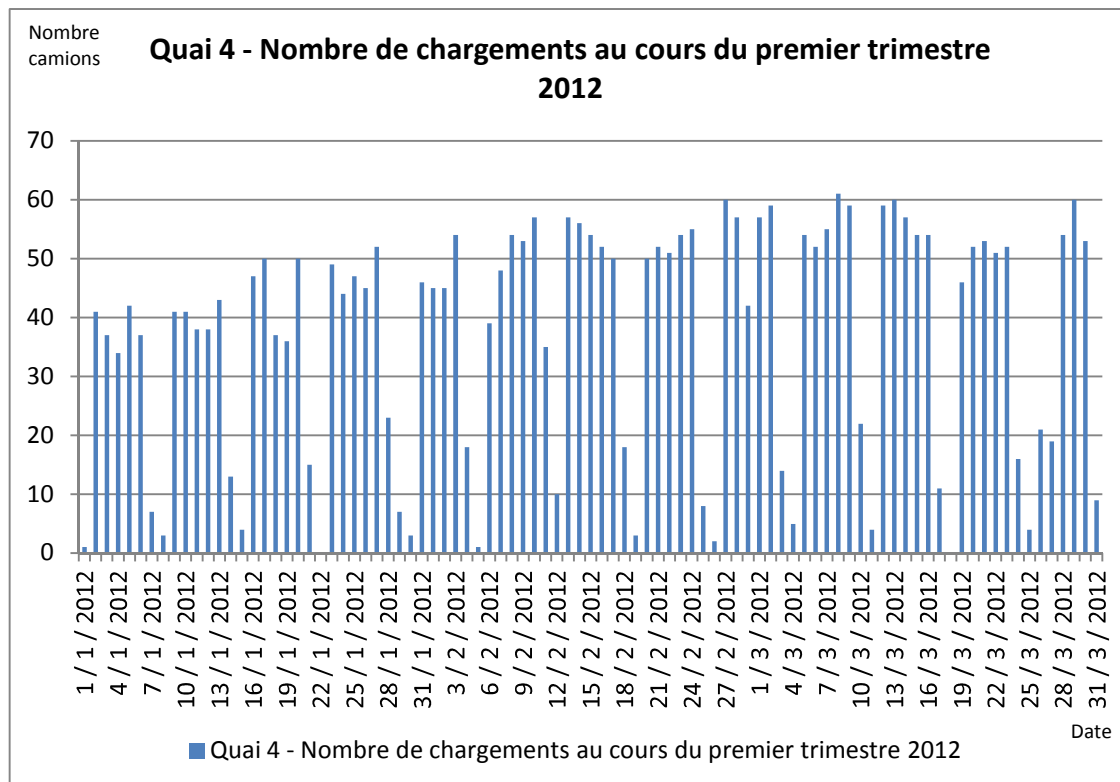
3. Quai 2



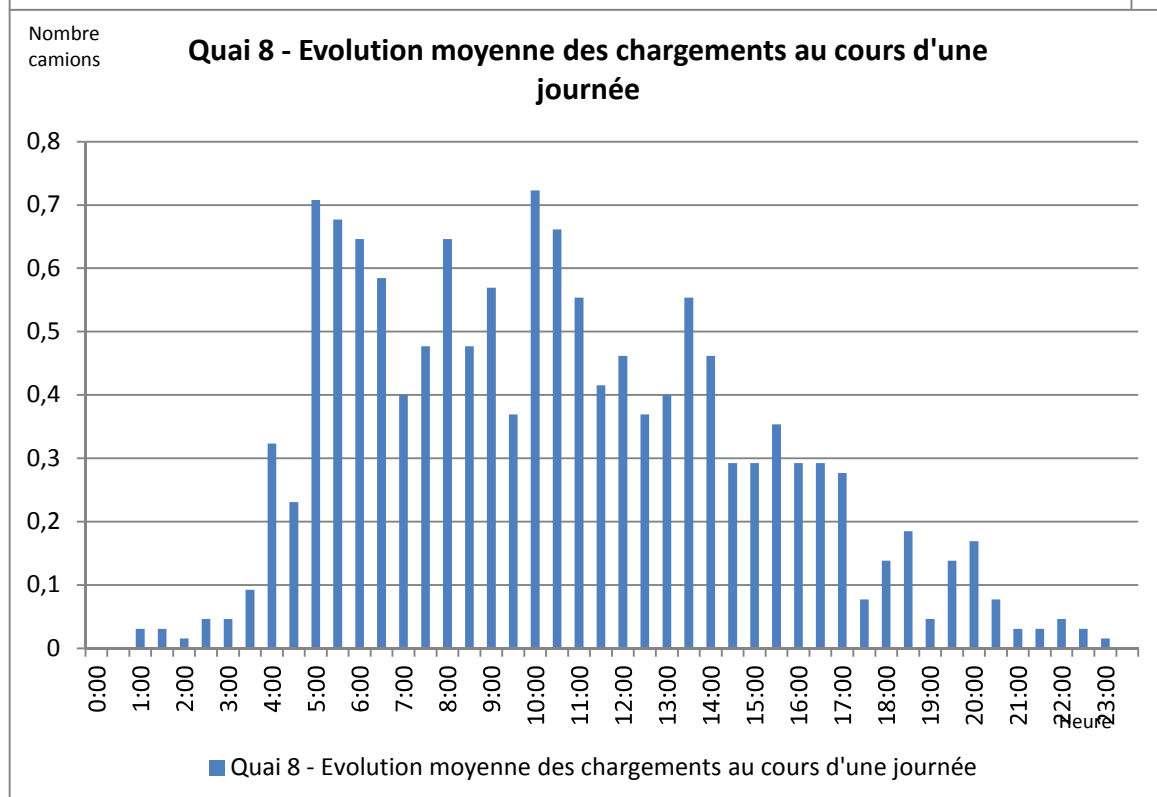
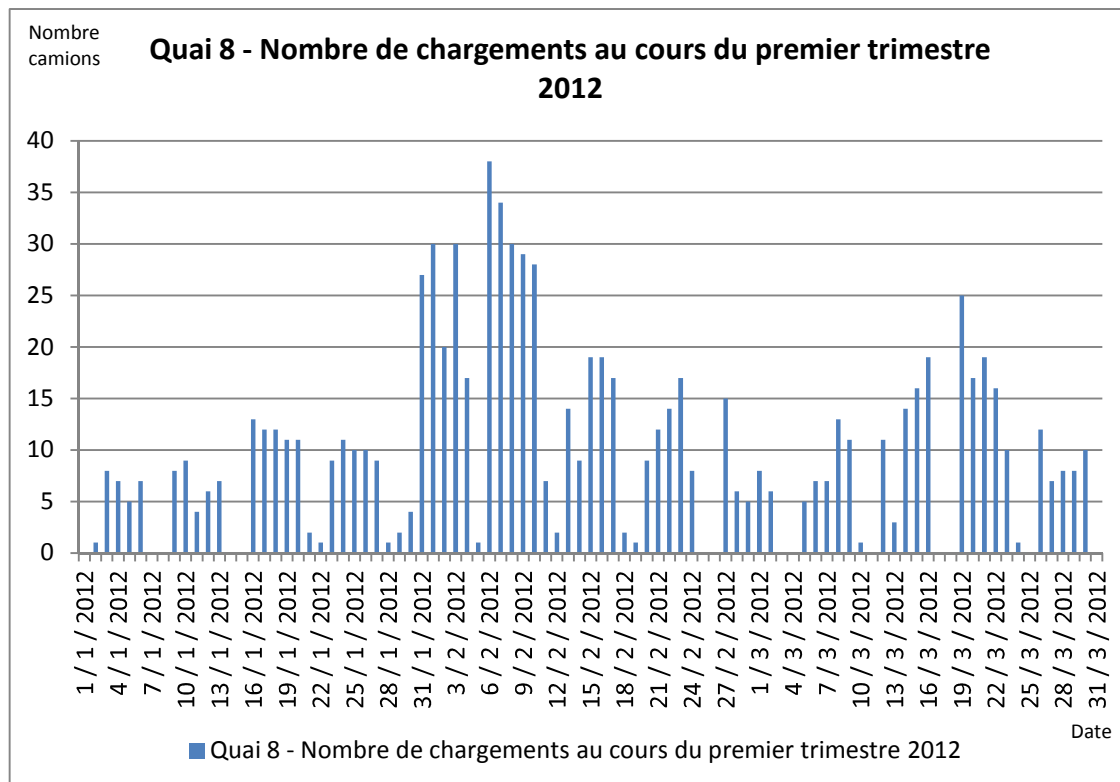
4. Quai 3



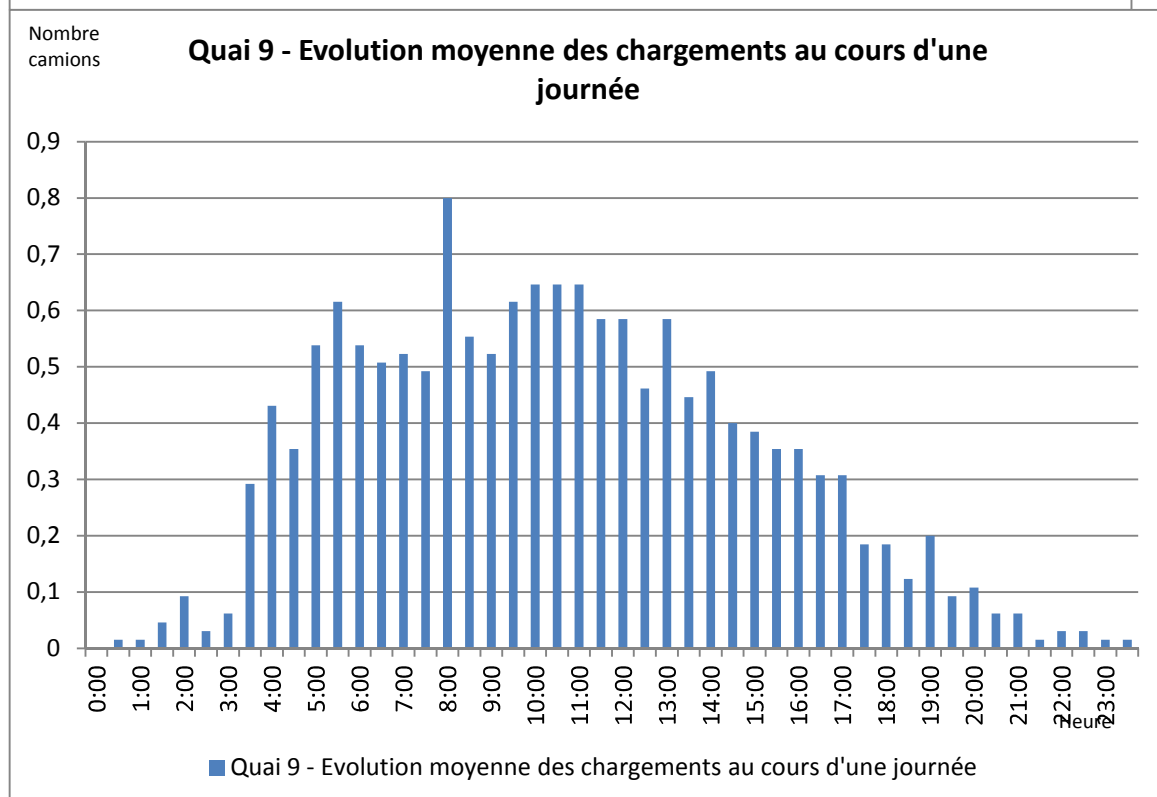
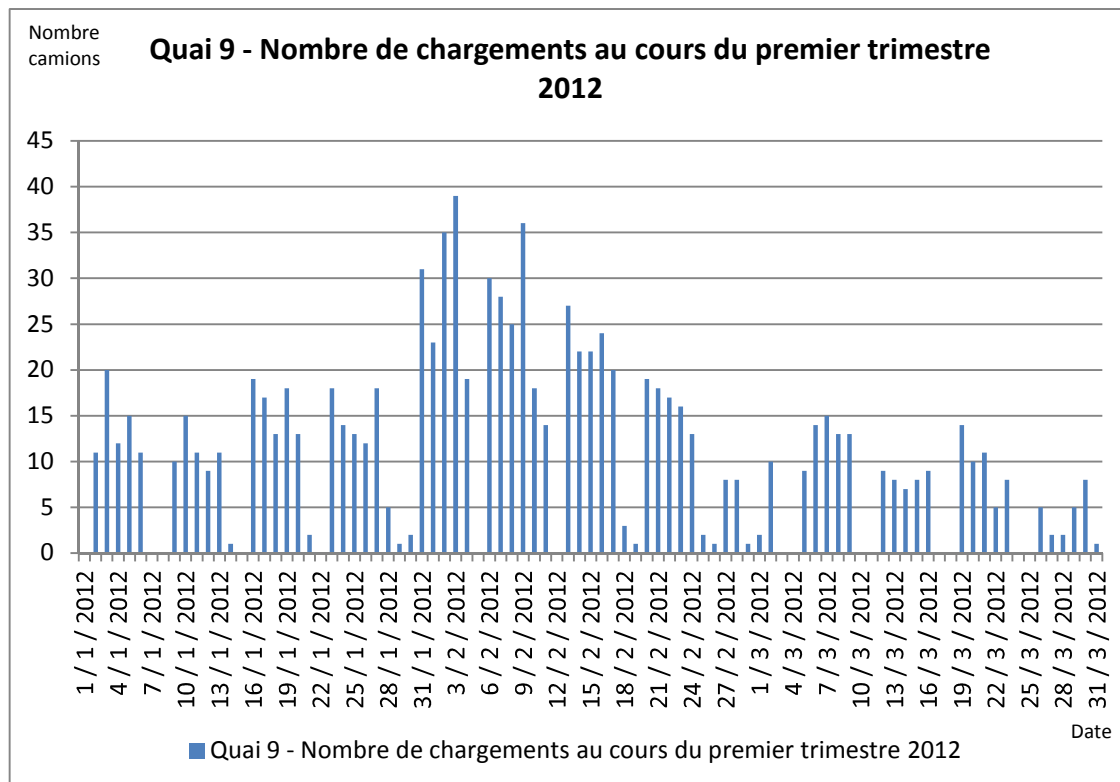
5. Quai 4



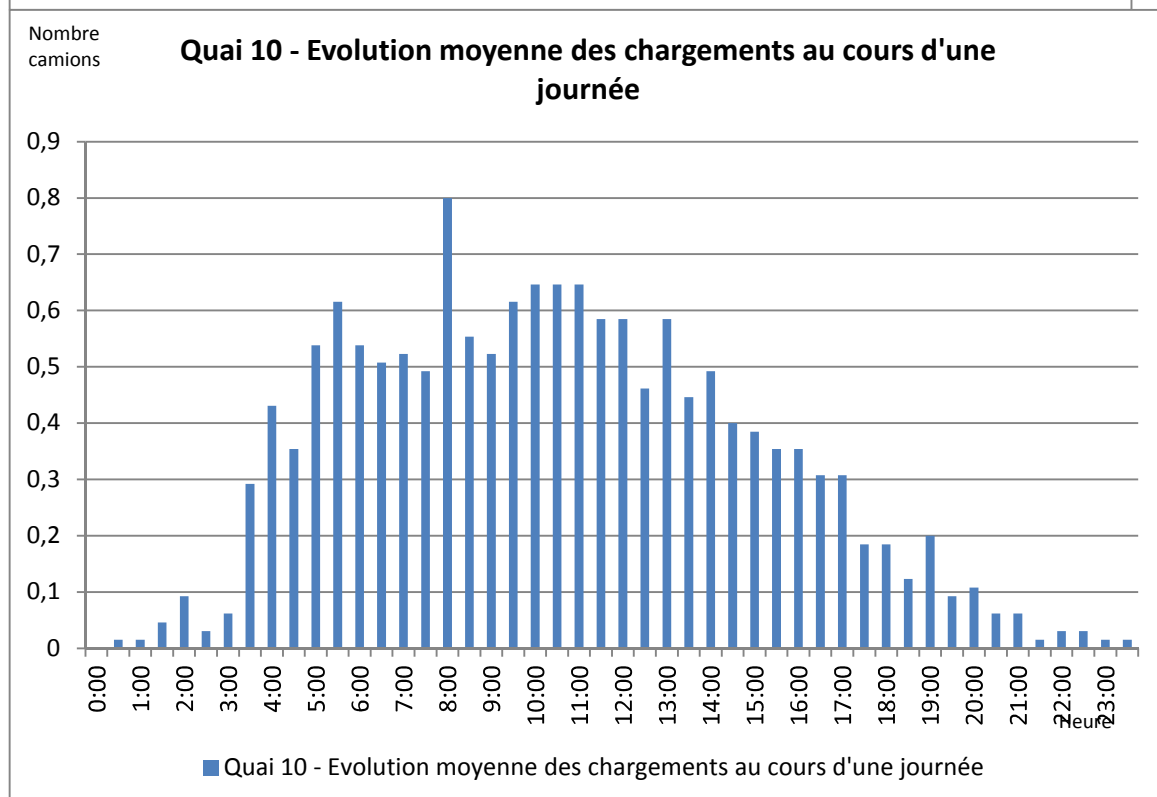
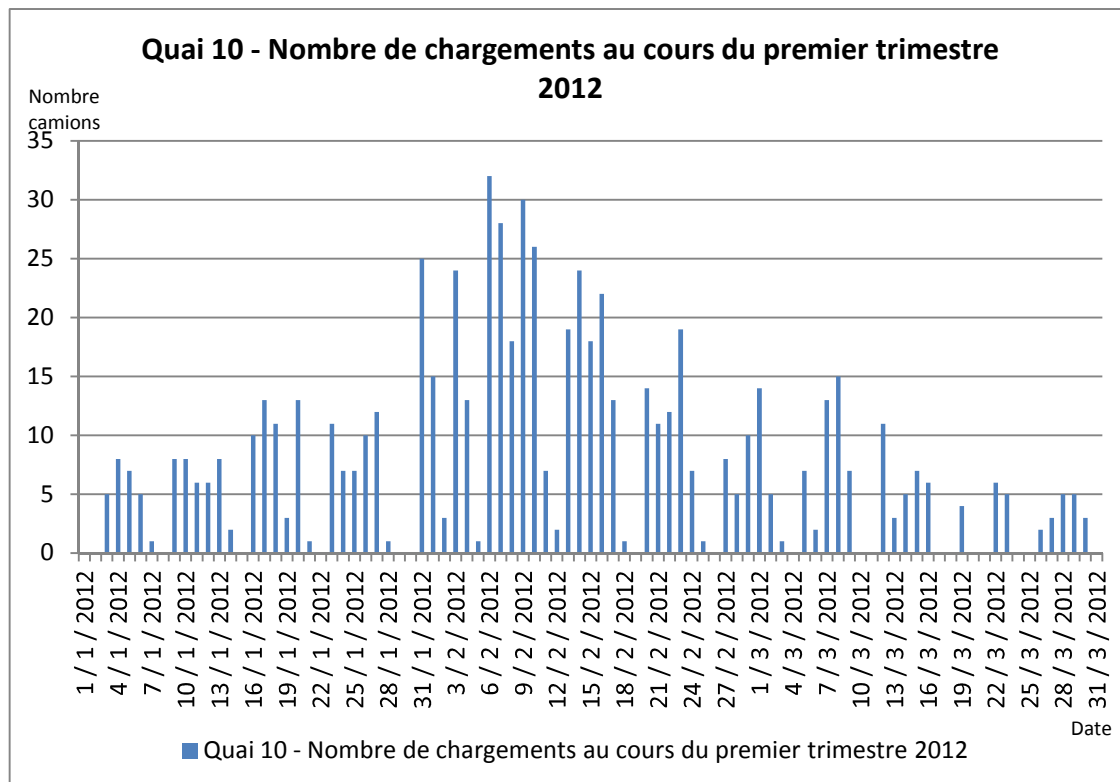
6. Quai 8



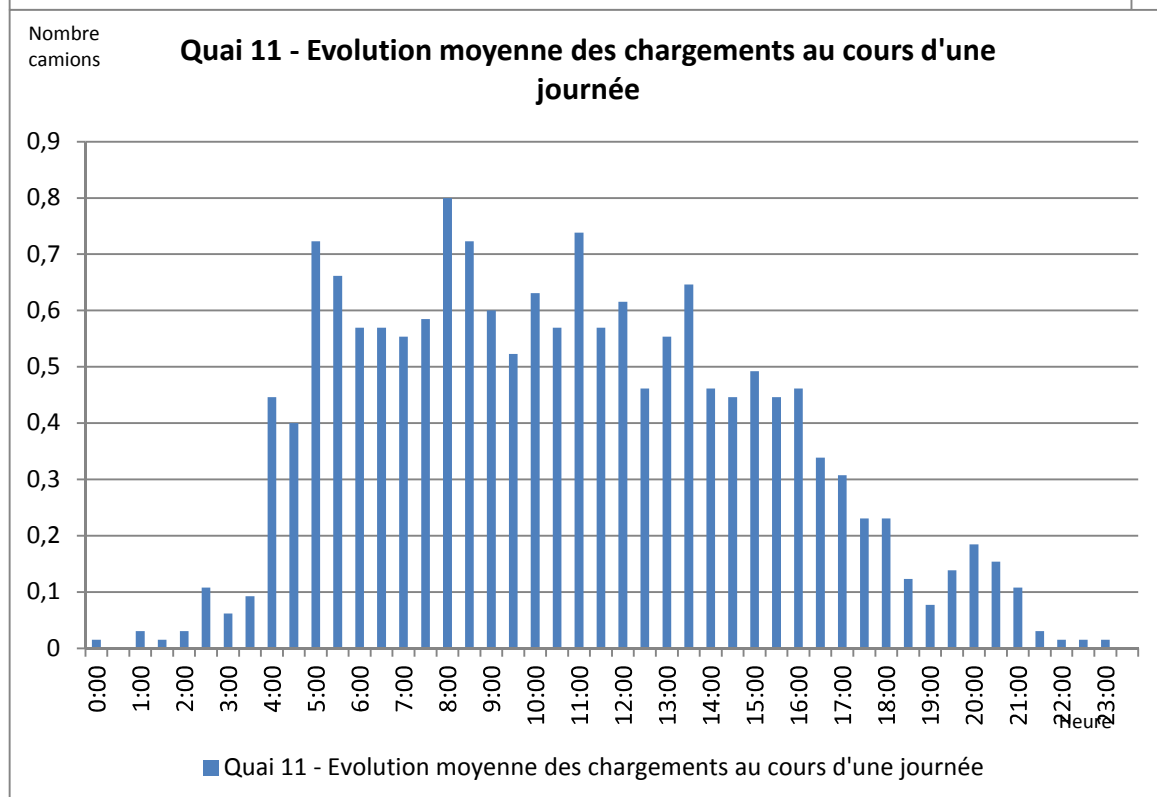
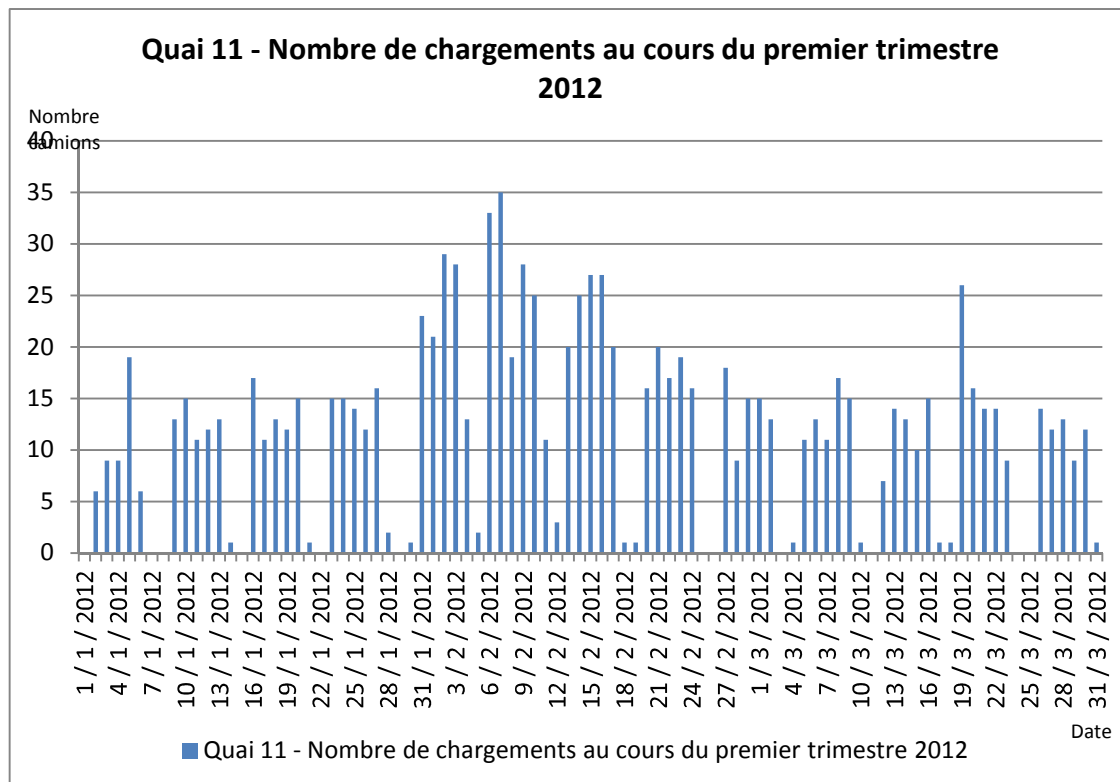
7. Quai 9



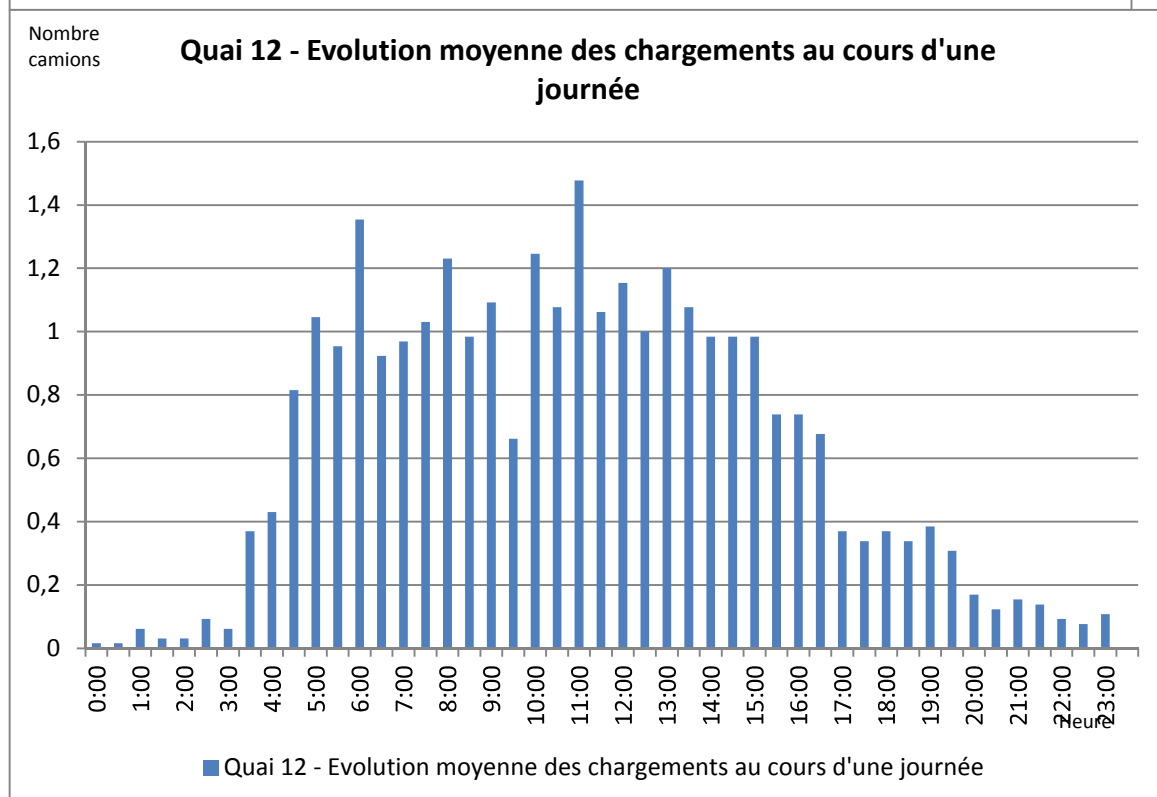
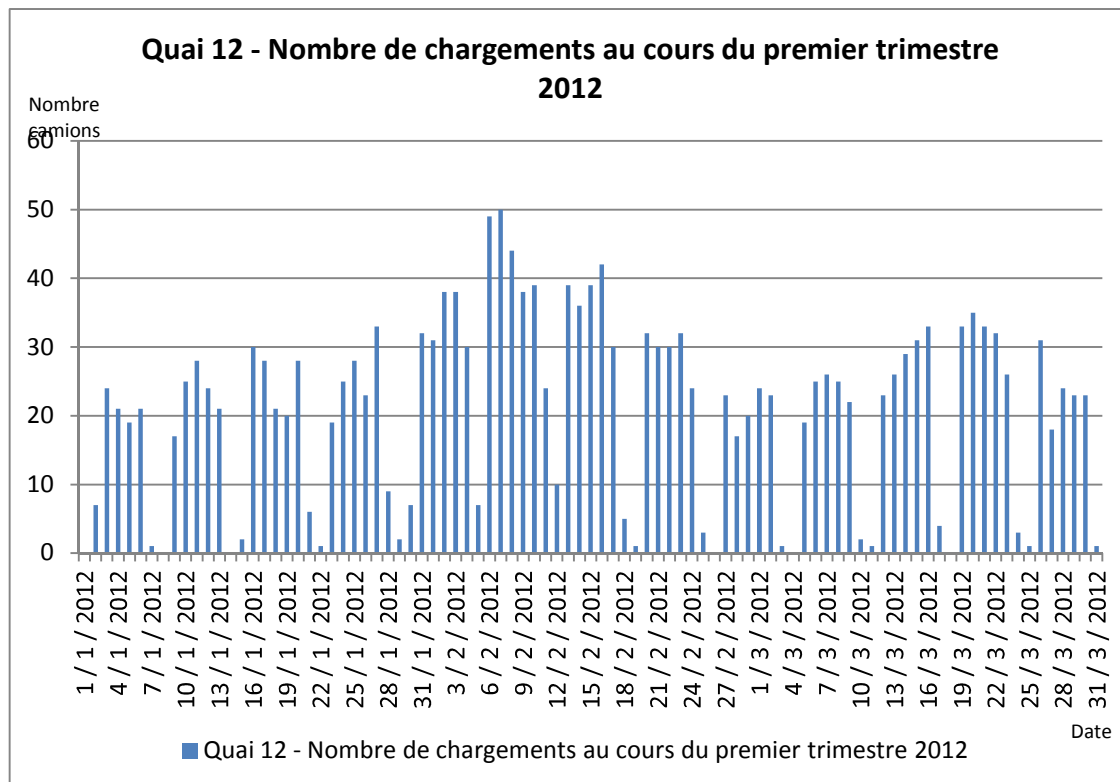
8. Quai 10



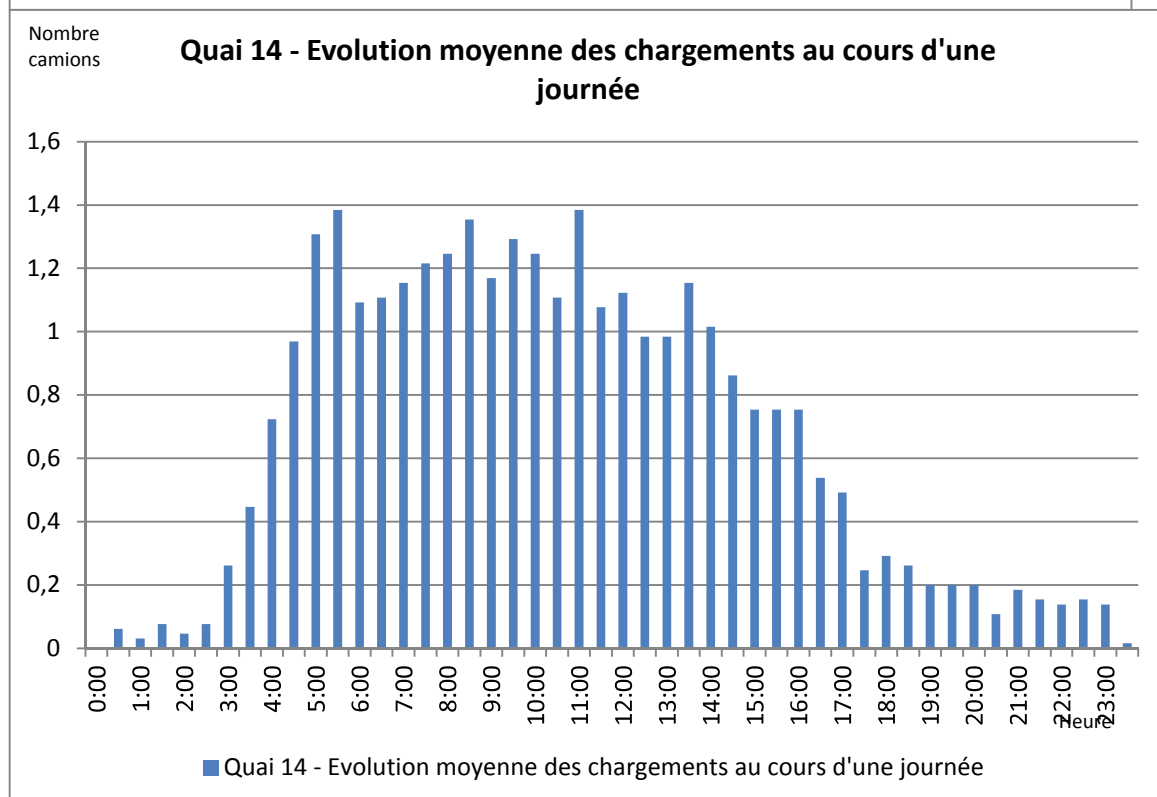
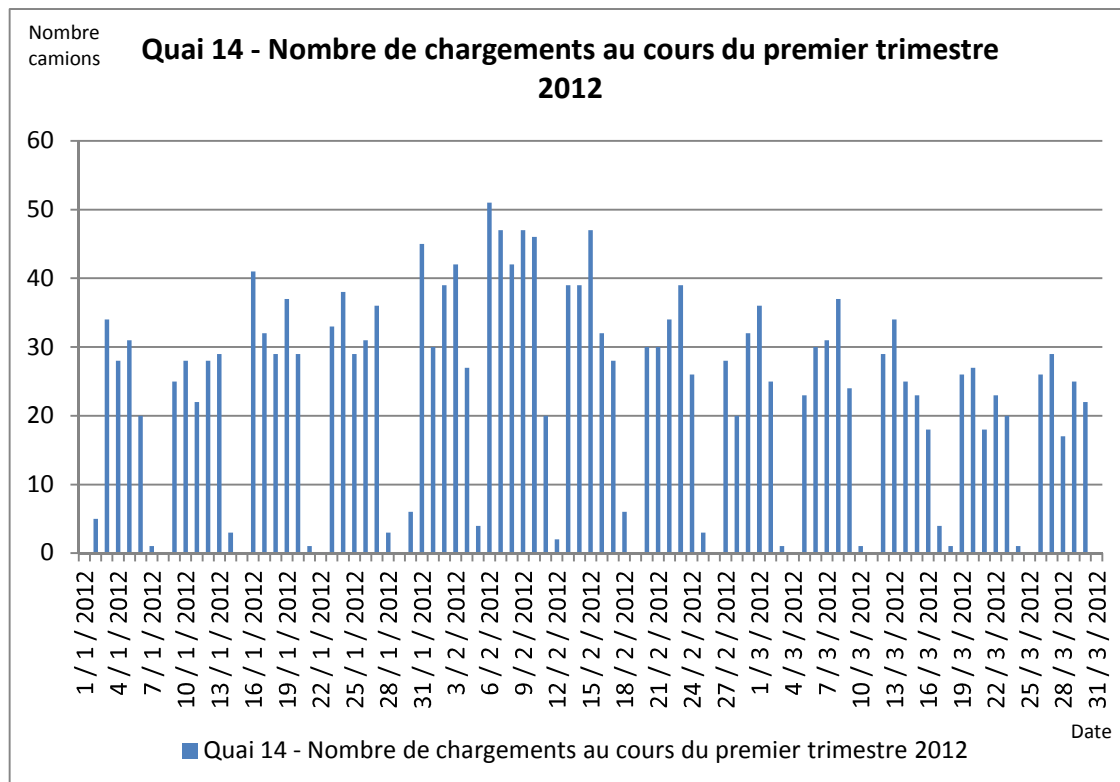
9. Quai 11



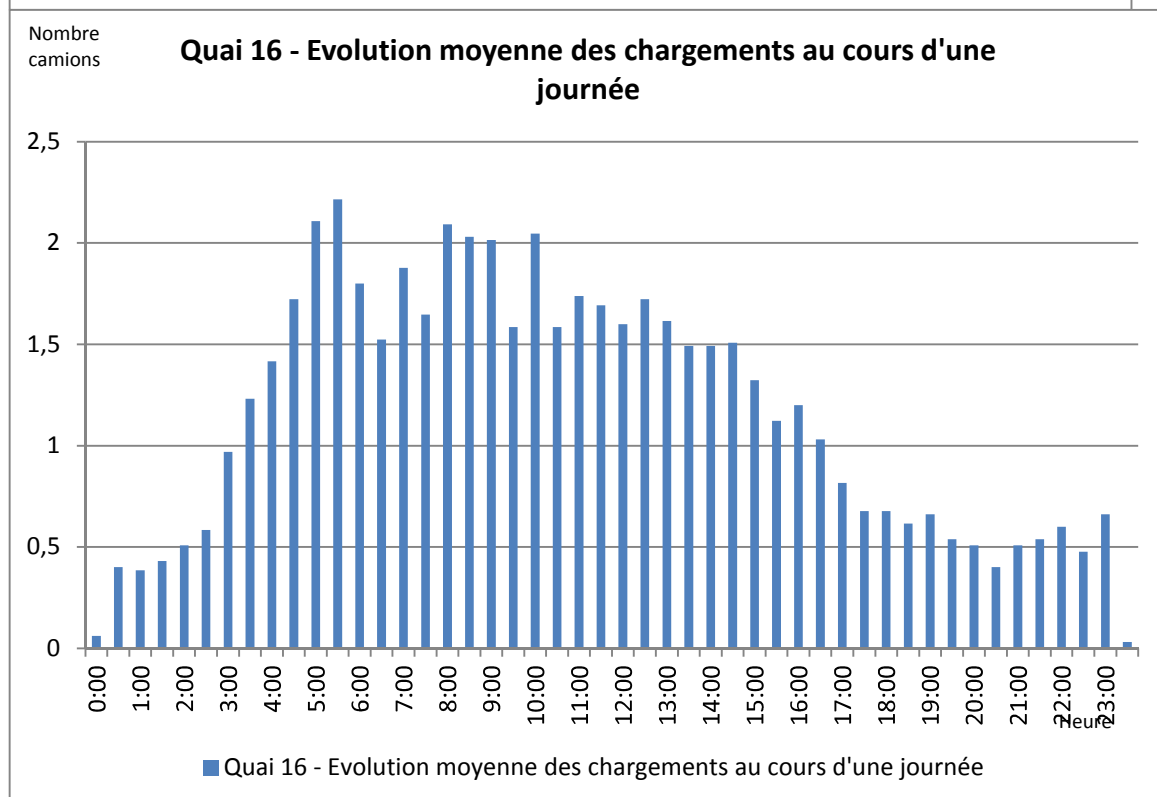
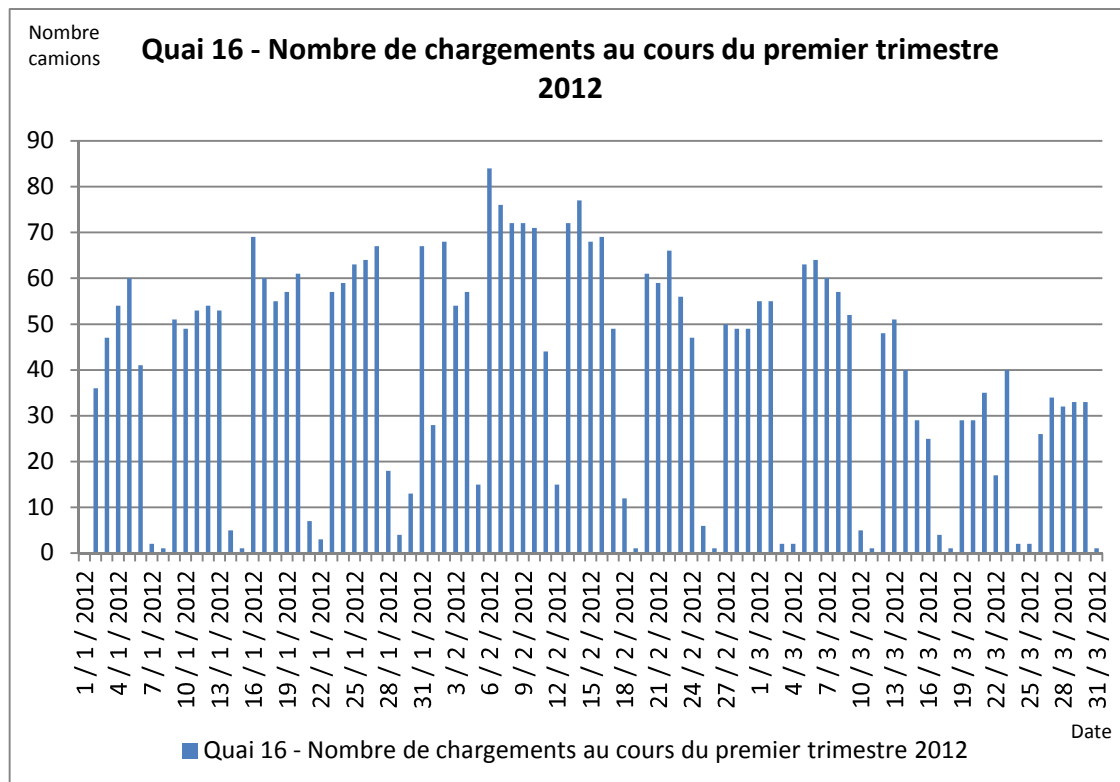
10. Quai 12



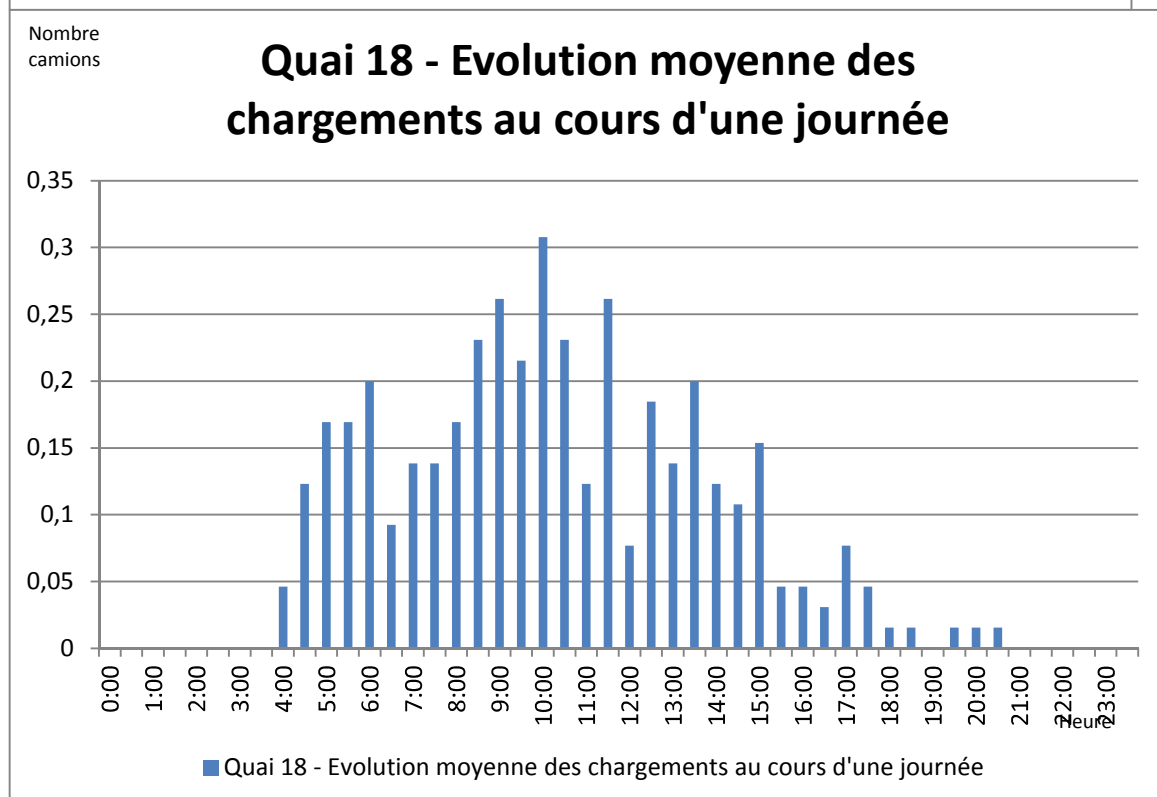
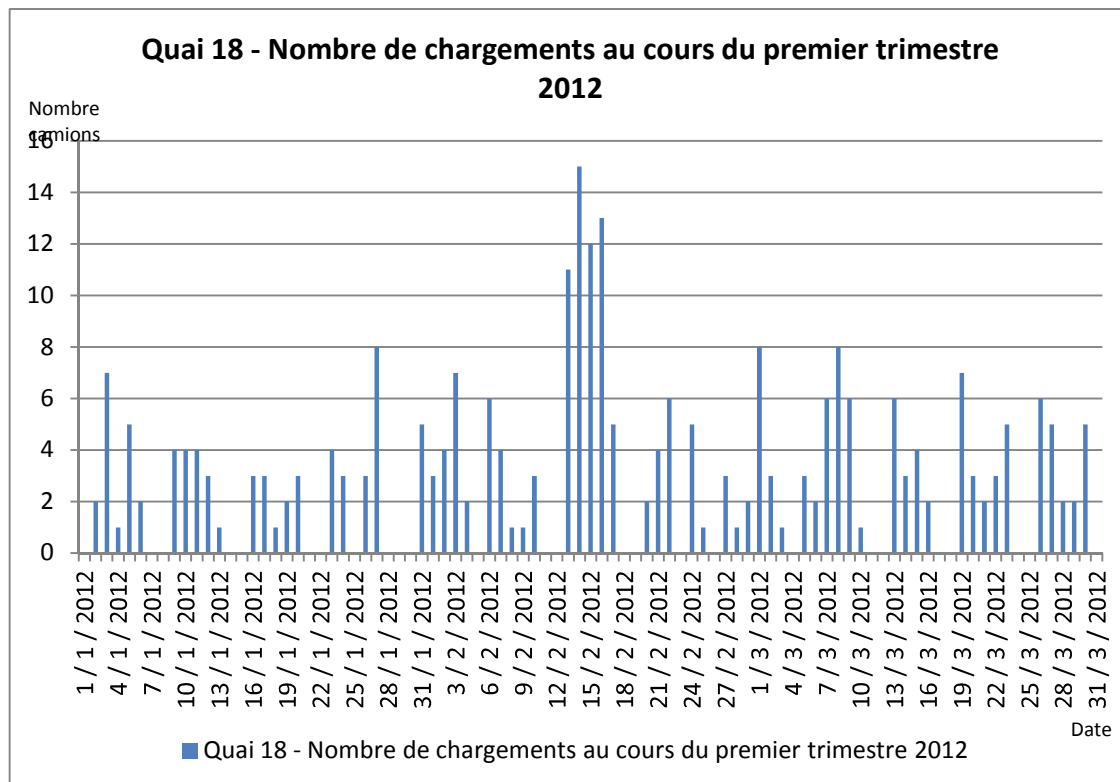
11. Quai 14



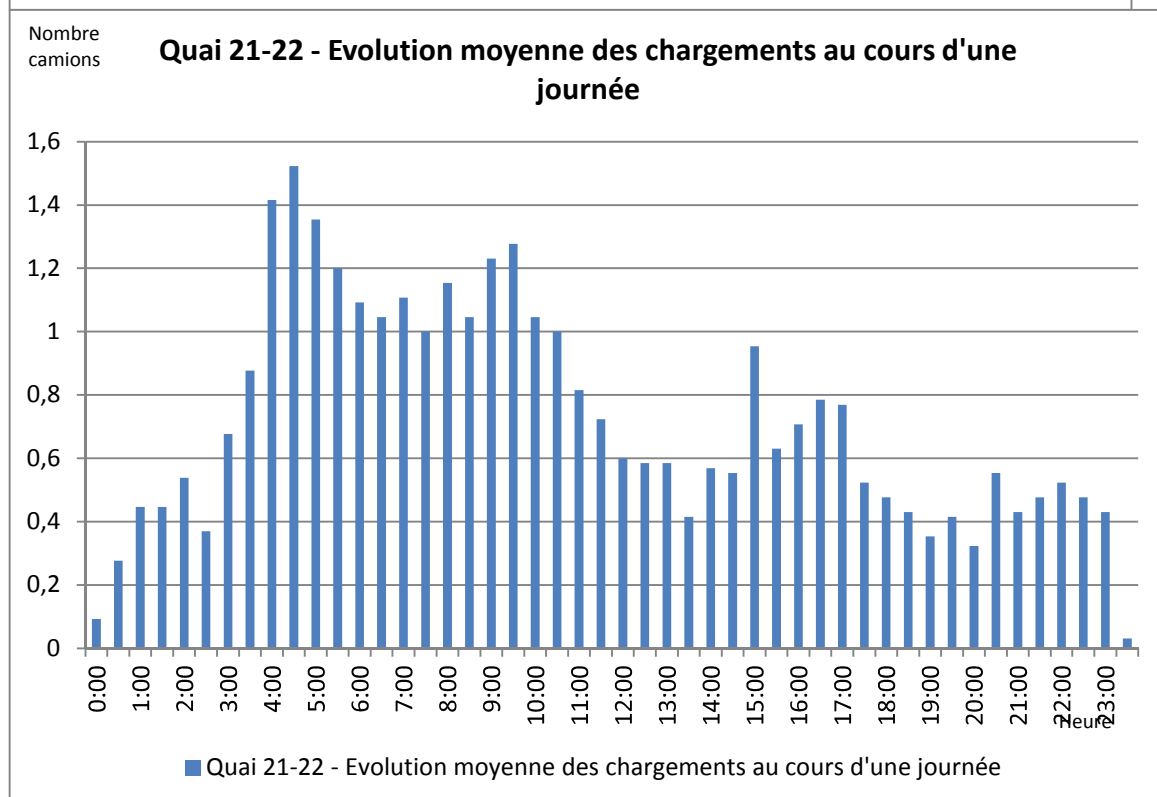
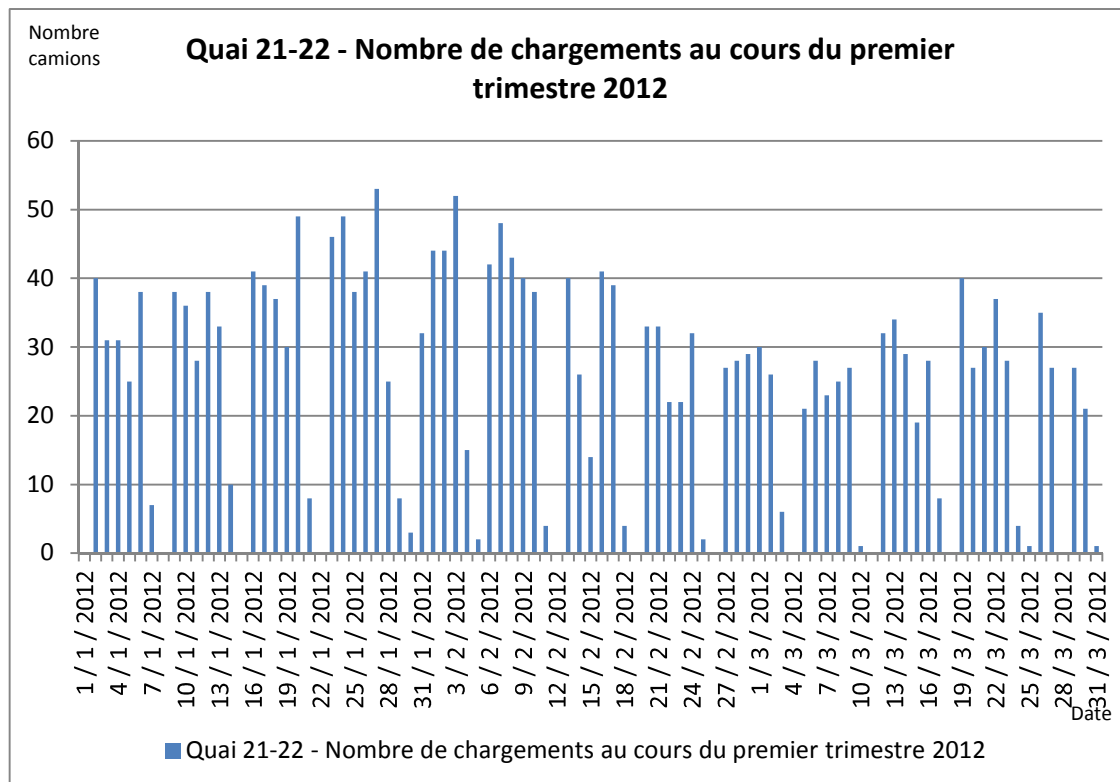
12. Quai 16



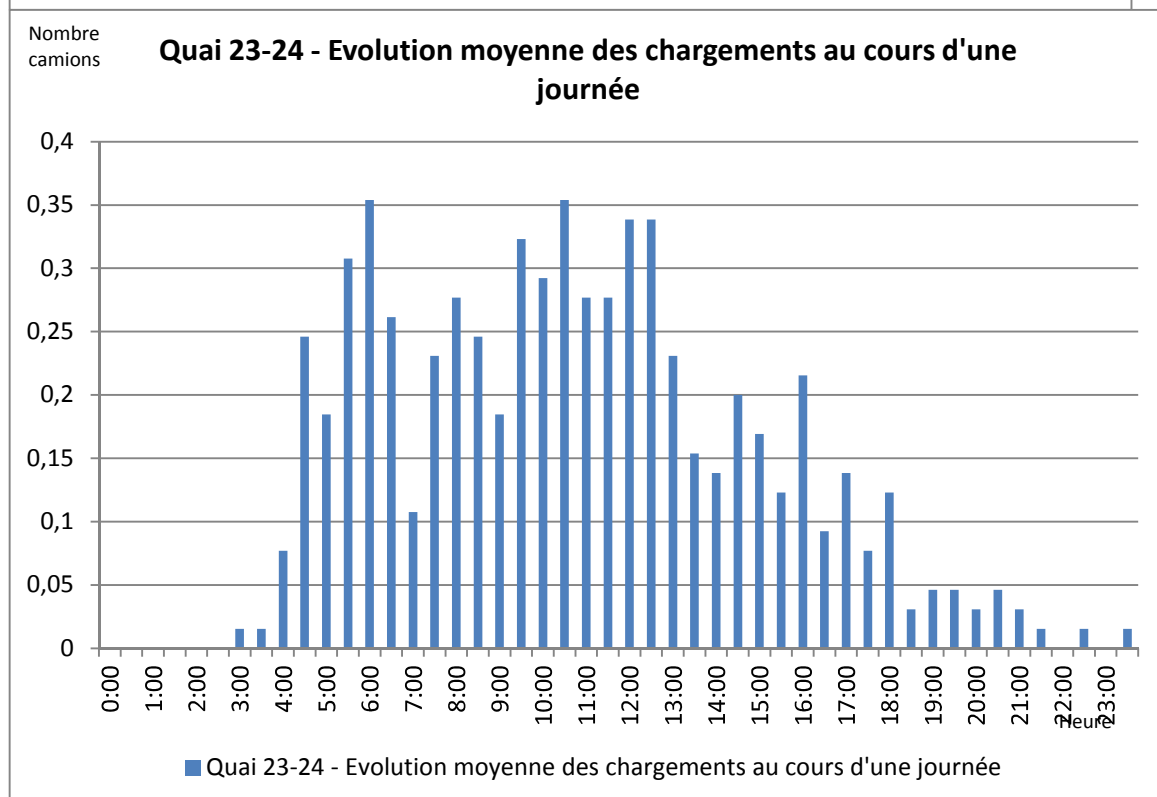
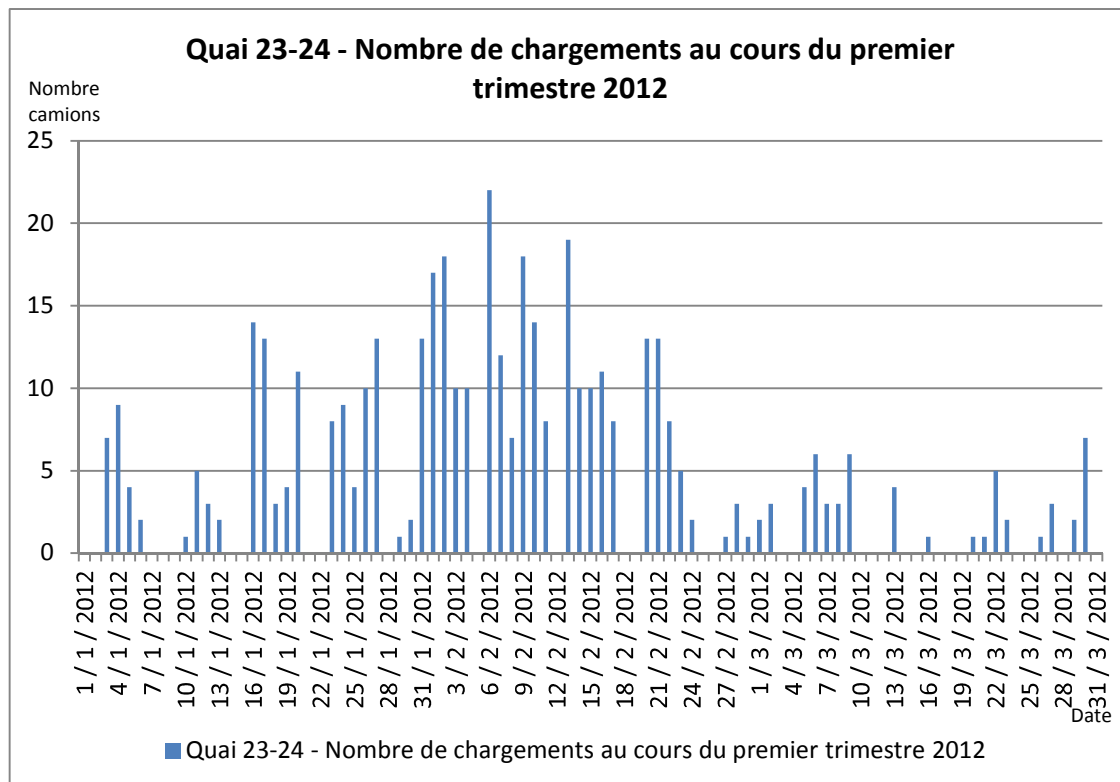
13. Quai 18



14. Quai 21-22



15. Quai 23-24



16. Quai 25-26

