

UNIVERSITE D'AVIGNON ET DES PAYS DU VAUCLUSE

INSTITUT UNIVERSITAIRE DE TECHNOLOGIE

Département STatistiques et Informatique Décisionnelle (STID)



ANNEXES

Création d'un site de traitement des interventions informatiques et de rapports statistiques sur ces interventions

MAIRIE DE MANOSQUE

COMMUNAUTE DE COMMUNES LUBERON DURANCE VERDON



Maître de stage : Mlle Jany SABATY

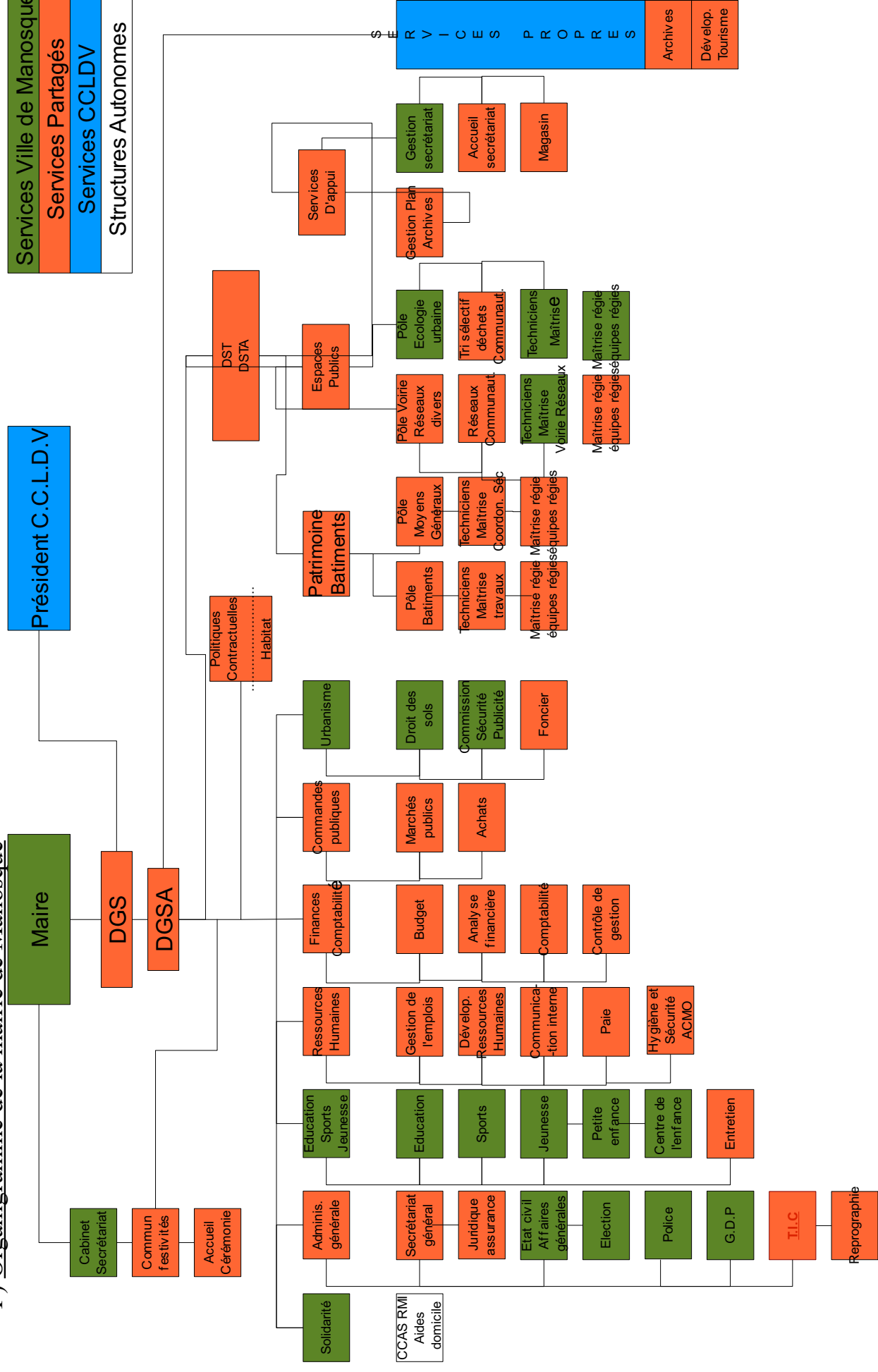
Tuteur : Mr Pierre-Michel BOUSQUET

SOULENQ Bastien
Année universitaire 2009 - 2010

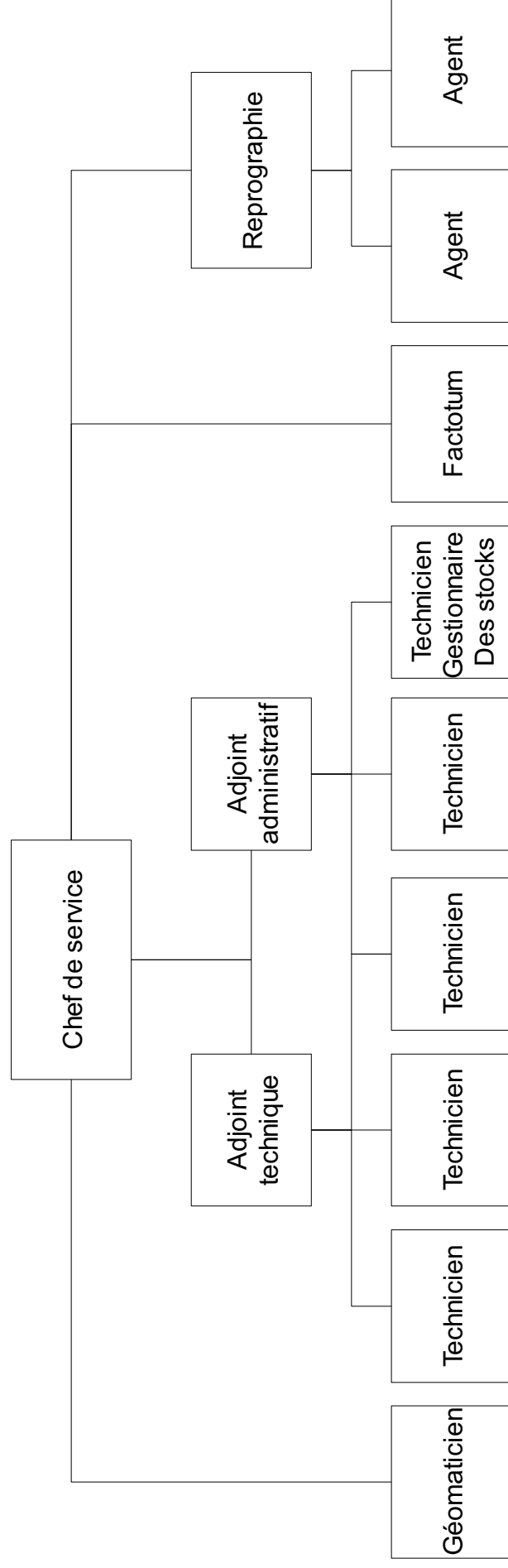
SOMMAIRE

Sommaire	
I. Organigramme de la mairie de Manosque	P02
II. Organigramme du service T.I.C.	P03
III. Explication de objet XMLHttpRequest	P04
IV. Explication du code Php d'insertion des données dans la base Mysql	P07
V. Compte rendu de la réunion du 20/04/2010	P08
VI. Le système de tag	P09
VII. Compte rendu de la réunion du 12/05/2010	P11
VIII. Script R	P12

Services Ville de Manosque
Services Partagés
Services CCILDV
Structures Autonomes



II) Organigramme du service T.I.C



III) Explication de objet XMLHttpRequest

La méthode des objets XMLHttpRequest permet de récupérer des informations sur le serveur sans avoir a recharger la page ! :
voici une explication de son fonctionnement.

Étape 1 : Créer un objet XMLHttpRequest

langage : Javascript

```
100 function go() {  
101     var xhr = getXhr();  
102     xhr.onreadystatechange = function() {  
103         if(xhr.readyState == 4 && xhr.status == 200) {  
104             document.getElementById('utilisateur').innerHTML = xhr.responseText;  
105         }  
106     }  
107     xhr.open("POST", "test.php", true);  
108     xhr.setRequestHeader('Content-Type', 'application/x-www-form-urlencoded');  
109     sel = document.getElementById('service');  
110     util = sel.options[sel.selectedIndex].value;  
111     xhr.send("utilisateur="+util);  
112 }
```

Dessin 1: fonction faisant appel au XMLHttpRequest

Étape 1 :

Fonction qui permet de créer un objet XMLHttpRequest en fonction du navigateur utilisé, cet objet est créé dans la variable xhr Ligne 101

```
function getXhr() {  
    var xhr = null;  
    if(window.XMLHttpRequest)  
        xhr = new XMLHttpRequest();  
    else if(window.ActiveXObject) {  
        try {  
            xhr = new ActiveXObject("Msxml2.XMLHTTP");  
        } catch (e) {  
            xhr = new ActiveXObject("Microsoft.XMLHTTP");  
        }  
    } else {  
        alert("Votre navigateur ne supporte pas les objets XMLHttpRequest...");  
        xhr = false;  
    }  
    return xhr;  
}
```

Dessin 2: fonction 1 création de l'objet

Étape 2 :

Ouvre une connexion avec la page « test.php » par la méthode d'envoi des données POST (caché) Ligne 107

Étape 3 :

On prépare les données que l'on souhaite envoyer à la page php (ici la valeur contenue dans la liste service) Ligne 109-110
La ligne 108 permet l'envoi des données par la méthode POST
On envoie les données (Ligne 111) dans ce cas, on envoie la valeur de la liste déroulante dans une variable « utilisateur »

Test.php

```
1  <?php
2  echo "<select name='util_name' id='util_name' onchange='go2()' ">";
3  echo "<OPTION VALUE='-1'>---</OPTION>";
4  include 'biblio.php';
5  $query = "SELECT Nom FROM utilisateurs WHERE service='";
6  $query .= htmlentities($_POST['utilisateur'], ENT_QUOTES);
7  $query .= "' ORDER BY Nom";
8  $result = mysql_query($query, $connect);
9  while($row = mysql_fetch_row($result)){
10     echo "<option value='". $row[0]."'>". $row[0]. "</option>";
11 }
12 echo "</select>";
13 ?>
```

Dessin 3: Page test.php

Étape 4 :

Lorsque la valeur de « ReadyState » vaut 4, (« ReadyState » étant une fonction sur l'état de la requête demandée, de 0 « requête initialisée » à 4 « résultats prêts à être utilisés ») on peut utiliser le résultat obtenu dans la page appelée, ce résultat apparaît dans la variable « responsetext »
Dans cet exemple, le résultat s'affiche dans une balise contenant une liste vide. Donc lorsque l'utilisateur sélectionne son service, la liste des noms qui est positionnée à côté et qui est vide est automatiquement remplacé par une liste des membres du services sélectionné.

Ce code php affiche une liste déroulante des utilisateurs appartenant au service précédemment sélectionné.

IV) Explication du code Php d'insertion des données dans la base MySQL

```
1 <?PHP
2 include 'biblio.php';
3 ob_start();
4 $query = "INSERT INTO demandes(ID_demande,Nom,service,Denomination,Objet,Etat,Details,Date) ";
5 $query .= "VALUES('','$$_POST['Nom'],'','$$_POST['service'],'','$$_POST['De_nom']";
6 $query .= "','','$$_POST['Objet'],'','$$_POST['Description'],'','$$_POST['o-m-d']).'')";
7 mysql_query($query, $connect);
8 echo 'demande enregistrée';
9 echo "<a href='accueil.php'>retour &grave; la page d'accueil :</a><br><br>";
10     Redirection automatique dans 5 secondes";
11 header('Refresh: 5; url=accueil.php');
12 ob_flush();
13 ?>
```

Explications :

Ligne 2 : inclure la page « biblio.php » qui contient les informations pour se connecter à la base de données.

Lignes 4, 5 et 6 : écrit la requête SQL dans une variable \$query

Ligne 7 : envoi de la requête au serveur.

Lignes 1, 11 et 12 : Permet le rafraichissement automatique de la page

Lignes 8, 9 et 10 : Ce qui est affiché sur la page

V) Compte rendu de la réunion du 20/04/2010

Mardi 27 avril 2010

Bilan

Problème initial : Réaliser des statistiques sur les demandes d'interventions faite au service informatique.

Tâches à effectuer :

- Réaliser un formulaire permettant les demandes d'interventions
- Réaliser un formulaire permettant le traitement des demandes
- réaliser les statistiques sur les interventions.

Première idée :

2 sites : un pour les demandes, un pour les interventions et une base mysql pour stocker les données.

(voir image + code dossier idee1)

Réunion bilan le 20/04 :

Manque de clarté au niveau d'objet de la demande (nouvelles questions ?, regroupement de certaines catégories pour plus de facilité)

Problème au niveau du formulaire de traitement : pas de lien avec l'inventaire déjà existant GLPI, association du nom du technicien à un temps (contesté, à revoir).

Deuxième idée :

- modification structurelle : regroupement des deux sites en un seul avec un système de connexion.
- Amélioration des formulaires de demande et de traitement.
- Modification au niveau de la base de données: récupération des données à « la source » sur le serveur LDAP. Liste des services, des utilisateurs et du matériel.

Logiciels et langages utilisés pour la réalisation du projet :

- langages de programmation : php, javascript, ajax, SQL
- logiciels : phpmyadmin, pour la gestion de la base de données

VI) Le système de tag

Le système de tag utilise la méthode des objet XMLHttpRequest (annexe III)

Pour ajouter un tag, il faut sélectionner son nom dans la liste :

Interventions en cours

Service	Nom	Date	Objet	qui s'en occupe ?	tag	etat
Accueil	nom2	18/05/2010	assistance utilisateur	 	--- ▾	Recu
Urbanisme	nom6	17/05/2010	Reseau		--- ▾	Recu
Accueil	nom2	12/05/2010	PC	  	--- ▾	Recu
Education	nom10	12/05/2010	PC		--- ▾ --- ▾ bastien user2 user3 user4	En cours

Dessin 4: selection du nom

Ce qui par le biais du XMLHttpRequest ajoute une couleur

Interventions en cours

Service	Nom	Date	Objet	qui s'en occupe ?	tag	etat
Accueil	nom2	18/05/2010	assistance utilisateur	 	--- ▾	Recu
Urbanisme	nom6	17/05/2010	Reseau		--- ▾	Recu
Accueil	nom2	12/05/2010	PC	  	--- ▾	Recu
Education	nom10	12/05/2010	PC		--- ▾	En cours

Dessin 5: ajoute une case de couleur

Pour retirer un tag un double clic sur celui ci et après confirmation il est supprimé



Dessin 6: demande de confirmation

Au niveau de la base de données l'ajout ou la suppression d'un tag correspond a l'ajout ou la suppression d'une ligne dans la table `interv_en_cours` ou sont stockés l'id de la demande et le nom du technicien qui l'a taggué.

			id_demande	technicien
☐			3	bastien
☐			4	bastien
☐			5	user2
☐			4	user3
☐			7	user2
☐			8	user3
☐			9	bastien
☐			9	user2
☐			10	user3
☐			30	user3
☐			29	user2
☐			29	bastien
☐			28	bastien
☐			26	bastien
☐			26	user2
☐			26	user3
☐			27	user2
☐			24	bastien
☐			25	user3
☐			23	bastien
☐			22	user2
☐			22	user3
☐			20	bastien
☐			14	user2

*Dessin 7: extrait de la table
`interv_en_cours`*

VII) Compte rendu de la réunion du 12/05/2010

Mercredi 12/05/2010 – Réunion bilan sur le travail accompli

Bilan

Améliorations structurelles à faire :

- ajouter 'autres' aux dénominations
- passer le temps d'une intervention en une liste déroulante par quart d'heure et non plus en texte libre
- lorsqu'une intervention est résolue, envoyer un mail au demandeur s'il a renseigné le champ dans la demande. Le mail doit contenir une réponse générique plus un classement : 'résolu' ou 'sans suite'.
- Nouveau champ dans les demandes : 'mail' à remplir automatiquement avec le contenu de ldap mais laisser la possibilité de le modifier.
- Ajouter une case pour permettre la suppression des interventions avec confirmation javascript.
- Ajouter un état d'intervention : 'Ne concerne pas le service TIC'. Lors de la sélection de cette option, envoi d'un mail générique à la personne qui a formulé la demande si le mail a été renseigné par le demandeur.
- Faire un affichage conditionnel des menus de gestion et du tableau des couleurs en fonction de l'adresse de départ.

Statistique :

La rédaction des rapports automatiques se fera au moyen avec la méthode Sweave sous R puis le passage en pdf par MikTeX (idée initiale). Une solution secondaire serait de générer des rapports odf avec le package odfWeave de R

Mise en réseau local à l'échelle du service informatique la semaine prochaine, afin de tester le site.

Reste à faire en plus des modifications citées ci-dessus :

- La page d'accueil
- la page d'aide
- La partie administration (Ajax sur les modifications profils)
- La page options
- La page statistique
- Modification de sécurité : protéger le site des attaques par injection de code sur les formulaires
- Mise en conformité avec la norme w3c : Valider chaque page sur le site <http://validator.w3.org/> Pour le moment seul index.php est valide w3c.

Connexion avec l'annuaire LDAP : tester si la connexion est possible (données à récupérer : liste des services et de leur personnel ainsi que leurs adresses mails locales s'il y en a une).

VIII) Script R

```
.libPaths("E:/std/Cours/Library")

#####
# Formatage des données

source("data/format.R");

#####
# Importation des données

donnees = read.delim("donnees_f.csv", header=TRUE)

#####
# Statistique
nom_tab = c("Nombre de demandes", "Nombre d'interventions", "Nombre moyen d'intervention par demande",
            "Temps moyen d'intervention par demande", "Temps moyen d'une intervention", "PC", "Imprimantes",
            "Telephones", "Reseau", "assistance utilisateur");

#####
#                               #
#####
#Tableau récapitulatif :
tab_recap_global = matrix(c(dim(donnees)
[1],sum(donnees[,18]),mean(donnees[,18]),mean(donnees[,17]),mean(donnees[,17] /
donnees[,18]),round(length(donnees[,1][donnees[,5] == "PC") / dim(donnees)[1] * 100, 2),round(length(donnees[,1]
[donnees[,5] == "Imprimante"]) / dim(donnees)[1] * 100, 2),round(length(donnees[,1][donnees[,5] == "Telephone"]) /
dim(donnees)[1] * 100, 2),round(length(donnees[,1][donnees[,5] == "Reseau"]) / dim(donnees)[1] * 100, 2) ,
round(length(donnees[,1][donnees[,5] == "assistance utilisateur"]) / dim(donnees)[1] * 100, 2)),1,10)
colnames(tab_recap_global)= nom_tab

#####
#                               #
#####
x=matrix(4,10)
x[,1] = c(length(donnees[,1][donnees[,4] == "Mairie"]),length(donnees[,1][donnees[,4] == "CCLDV"]),
          length(donnees[,1][donnees[,4] == "Autre"]),length(donnees[,1][donnees[,4] == "Service_partage"]))

# Nombre de demandes.
x11()
d1 = pie(table(donnees[,4]),main="Nombre de demandes")

# Nombre d'interventions.
x11()
x[,2] = c(sum(donnees[,18][donnees[,4] == "Mairie"]),sum(donnees[,18][donnees[,4] == "CCLDV"]),sum(donnees[,18]
[donnees[,4] == "Autre"]),sum(donnees[,18][donnees[,4] == "Service_partage"]))
d2 = pie(x[,2],main="Nombre d'interventions", labels = c("Mairie","CCLDV","Autres","Services partagés"))

# Temps d'interventions.
x11()
y = c(sum(donnees[,17][donnees[,4] == "Mairie"]),sum(donnees[,17][donnees[,4] == "CCLDV"]),sum(donnees[,17]
[donnees[,4] == "Autre"]),sum(donnees[,17][donnees[,4] == "Service_partage"]))
d3 = pie(y,main="Temps d'interventions", labels = c("Mairie","CCLDV","Autres","Services partagés"))

# Nombre moyen d'interventions par demande.
x11()
x[,3] = c(mean(donnees[,18][donnees[,4] == "Mairie"]),mean(donnees[,18][donnees[,4] ==
```

```

"CCLDV")),mean(donnees[,18][donnees[,4] == "Autre"]),mean(donnees[,18][donnees[,4] == "Service_partage"]))
d4 = barplot(x[,3], main="Nombre moyen d'intervention par demande", names.arg =
c("Mairie","CCLDV","Autres","Services partagés"))

# Temps moyen d'interventions par demande en minutes.
x11()
x[,4] = c(mean(donnees[,17][donnees[,4] == "Mairie"]),mean(donnees[,17][donnees[,4] ==
"CCLDV"]),mean(donnees[,17][donnees[,4] == "Autre"]),mean(donnees[,17][donnees[,4] == "Service_partage"]))
d5 = barplot(x[,4], main="Temps moyen d'intervention par demande (en minutes)", names.arg =
c("Mairie","CCLDV","Autres","Services partagés"))

# temps moyen d'une intervention.
x11()
x[,5] = c(mean(donnees[,17][donnees[,4] == "Mairie"] / donnees[,18][donnees[,4] == "Mairie"]),mean(donnees[,17]
[donnees[,4] == "CCLDV"] / donnees[,18][donnees[,4] == "CCLDV"]),mean(donnees[,17][donnees[,4] == "Autre"] /
donnees[,18][donnees[,4] == "Autre"]),mean(donnees[,17][donnees[,4] == "Service_partage"] / donnees[,18]
[donnees[,4] == "Service_partage"]))
d6 = barplot(x[,5], main="Temps moyen d'une intervention (en minutes)", names.arg =
c("Mairie","CCLDV","Autres","Services partagés"))

# Objets des demandes
# PC
x11()
x[,6] = c(round(length(donnees[,1][donnees[,4] == "Mairie" & donnees[,5] == "PC"]) / length(donnees[,18]
[donnees[,4] == "Mairie"]), 2) * 100,round(length(donnees[,1][donnees[,4] == "CCLDV" & donnees[,5] == "PC"]) /
length(donnees[,18][donnees[,4] == "CCLDV"]), 2) * 100,round(length(donnees[,1][donnees[,4] == "Autre" &
donnees[,5] == "PC"]) / length(donnees[,18][donnees[,4] == "Autre"]), 2) * 100,round(length(donnees[,1][donnees[,4] ==
"Service_partage" & donnees[,5] == "PC"]) / length(donnees[,18][donnees[,4] == "Service_partage"]), 2) * 100)
d7 = barplot(x[,6], ylim=c(0,100), main="Part des demandes concernant les PC", names.arg =
c("Mairie","CCLDV","Autres","Services partagés"))

# Imprimante
x11()
x[,7] = c(round(length(donnees[,1][donnees[,4] == "Mairie" & donnees[,5] == "Imprimante"]) / length(donnees[,18]
[donnees[,4] == "Mairie"]), 2) * 100,round(length(donnees[,1][donnees[,4] == "CCLDV" & donnees[,5] ==
"Imprimante"]) / length(donnees[,18][donnees[,4] == "CCLDV"]), 2) * 100,round(length(donnees[,1][donnees[,4] ==
"Autre" & donnees[,5] == "Imprimante"]) / length(donnees[,18][donnees[,4] == "Autre"]), 2) *
100,round(length(donnees[,1][donnees[,4] == "Service_partage" & donnees[,5] == "Imprimante"]) /
length(donnees[,18][donnees[,4] == "Service_partage"]), 2) * 100)
d8 = barplot(x[,7], ylim=c(0,100), main="Part des demandes concernant les Imprimantes", names.arg =
c("Mairie","CCLDV","Autres","Services partagés"))

# Telephone
x11()
x[,8] = c(round(length(donnees[,1][donnees[,4] == "Mairie" & donnees[,5] == "Telephone"]) / length(donnees[,18]
[donnees[,4] == "Mairie"]), 2) * 100,round(length(donnees[,1][donnees[,4] == "CCLDV" & donnees[,5] ==
"Telephone"]) / length(donnees[,18][donnees[,4] == "CCLDV"]), 2) * 100,round(length(donnees[,1][donnees[,4] ==
"Autre" & donnees[,5] == "Telephone"]) / length(donnees[,18][donnees[,4] == "Autre"]), 2) *
100,round(length(donnees[,1][donnees[,4] == "Service_partage" & donnees[,5] == "Telephone"]) / length(donnees[,18]
[donnees[,4] == "Service_partage"]), 2) * 100)
d9 = barplot(x[,8], ylim=c(0,100), main="Part des demandes concernant les Téléphones", names.arg =
c("Mairie","CCLDV","Autres","Services partagés"))

# Reseau
x11()
x[,9] = c(round(length(donnees[,1][donnees[,4] == "Mairie" & donnees[,5] == "Reseau"]) / length(donnees[,18]
[donnees[,4] == "Mairie"]), 2) * 100,round(length(donnees[,1][donnees[,4] == "CCLDV" & donnees[,5] ==
"Reseau"]) / length(donnees[,18][donnees[,4] == "CCLDV"]), 2) * 100,round(length(donnees[,1][donnees[,4] ==
"Autre" & donnees[,5] == "Reseau"]) / length(donnees[,18][donnees[,4] == "Autre"]), 2) *
100,round(length(donnees[,1][donnees[,4] == "Service_partage" & donnees[,5] == "Reseau"]) / length(donnees[,18]
[donnees[,4] == "Service_partage"]), 2) * 100)

```

```

[donnees[,4] == "Service_partage")), 2) * 100)
d10 = barplot(x[,9], ylim=c(0,100), main="Part des demandes concernant le réseau", names.arg =
c("Mairie", "CCLDV", "Autres", "Services partagés"))

# Assistance utilisateur
x11()
x[,10] = c(round(length(donnees[,1][donnees[,4] == "Mairie" & donnees[,5] == "assistance utilisateur"]) /
length(donnees[,18][donnees[,4] == "Mairie"]), 2) * 100, round(length(donnees[,1][donnees[,4] == "CCLDV" &
donnees[,5] == "assistance utilisateur"]) / length(donnees[,18][donnees[,4] == "CCLDV"]), 2) *
100, round(length(donnees[,1][donnees[,4] == "Autre" & donnees[,5] == "assistance utilisateur"]) / length(donnees[,18]
[donnees[,4] == "Autre"]), 2) * 100, round(length(donnees[,1][donnees[,4] == "Service_partage" & donnees[,5] ==
"assistance utilisateur"]) / length(donnees[,18][donnees[,4] == "Service_partage"]), 2) * 100)
d11 = barplot(x[,10], ylim=c(0,100), main="Part des demandes concernant les assistances utilisateurs", names.arg =
c("Mairie", "CCLDV", "Autres", "Services partagés"))

### Tableau récapitulatif : ###
tab_recap_denomination = matrix(rbind(tab_recap_global,x),5,10)
colnames(tab_recap_denomination)= nom_tab
rownames(tab_recap_denomination)= c("TOUS", "Mairie", "CCLDV", "Autres", "Services partagés")

#####
#                Par service                #
#####

# Informations générales

glob = matrix(,length(unique(donnees[,2])),4)
rownames(glob) = unique(donnees[,2])
colnames(glob) = c("Nombre de dmd", "Temps global", "% globale en nb", "% global en tps")
for (i in 1:length(unique(donnees[,2]))) {
  glob[i,1] = length(donnees[,5][donnees[,2] == unique(donnees[,2])[i]])
  glob[i,2] = sum(donnees[,17][donnees[,2] == unique(donnees[,2])[i]])
  glob[i,3] = round(length(donnees[,5][donnees[,2] == unique(donnees[,2])[i]]) / dim(donnees)[1] * 100,2)
  glob[i,4] = round(sum(donnees[,17][donnees[,2] == unique(donnees[,2])[i]]) / sum(donnees[,17]) *
100,2)
}

# Objet des demandes

# en nombre
obj = matrix(,length(unique(donnees[,2])),5)
rownames(obj) = unique(donnees[,2])
colnames(obj) = c("PC", "Imp", "Tel", "Res", "Ass")
for (i in 1:length(unique(donnees[,2]))) {
  obj[i,1] = length(donnees[,3][donnees[,2] == unique(donnees[,2])[i]] & (donnees[,5] == "PC"))
  obj[i,2] = length(donnees[,3][donnees[,2] == unique(donnees[,2])[i]] & (donnees[,5] == "Imprimante"))
  obj[i,3] = length(donnees[,3][donnees[,2] == unique(donnees[,2])[i]] & (donnees[,5] == "Telephone"))
  obj[i,4] = length(donnees[,3][donnees[,2] == unique(donnees[,2])[i]] & (donnees[,5] == "Reseau"))
  obj[i,5] = length(donnees[,3][donnees[,2] == unique(donnees[,2])[i]] & (donnees[,5] == "assistance
utilisateur"))
}

# en % par service
obj_pct = matrix(,length(unique(donnees[,2])),5)
rownames(obj_pct) = unique(donnees[,2])
colnames(obj_pct) = c("PC", "Imp", "Tel", "Res", "Ass")
for (i in 1:length(unique(donnees[,2]))) {
  obj_pct[i,1] = round(length(donnees[,3][donnees[,2] == unique(donnees[,2])[i]] & (donnees[,5] ==
"PC")) / sum(obj_pct[i,]) * 100, 2)

```

```

obj_pct[i,2] = round(length(donnees[,3][(donnees[,2]==(unique(donnees[,2])[i])) & (donnees[,5] ==
"Imprimante")) / sum(obj[i,]) * 100 , 2)
obj_pct[i,3] = round(length(donnees[,3][(donnees[,2]==(unique(donnees[,2])[i])) & (donnees[,5] ==
"Telephone")) / sum(obj[i,]) * 100 , 2)
obj_pct[i,4] = round(length(donnees[,3][(donnees[,2]==(unique(donnees[,2])[i])) & (donnees[,5] ==
"Reseau")) / sum(obj[i,]) * 100 , 2)
obj_pct[i,5] = round(length(donnees[,3][(donnees[,2]==(unique(donnees[,2])[i])) & (donnees[,5] ==
"assistance utilisateur")) / sum(obj[i,]) * 100 , 2)
x11()
barplot(obj_pct[i,], ylim=c(0,100),col=c("blue","red","yellow","black","green"), main =
paste("Répartition des demandes du service" ,rownames(obj_pct)[i]))
}

```

```

#####
#           Autres informations           #
#####

```

```

# Nombre de pièces changées :
sum(donnees[,20])
table(donnees[,21])

```

```

#Interventions sous garantie :
sum(donnees[,19])

```

```

#aide sur un logiciel :
length(donnees[donnees[,11] != ""])
table(donnees[,11])

```