

Données CIRAME, Tmin&Tmax interpolées. Extraction Vaucluse

Avril 2015-04-28 , P. Clastre

Juin 2015 (mise à jour), P. Clastre

Juillet 2015 (mise à jour suite recalcul des données 2008), P .Clastre

Juillet 2015 corrections/validations, M. Guerif

Novembre 2015 Calcul version à 1000m de résolution

Objet : rapport de suivi pour l'extraction des données climatiques du Vaucluse (Tmin et Tmax) à partir des données brutes du Cirame.

Sommaire

Préambule	2
Description des données brutes.....	2
Identification des paramètres de lecture.....	3
Extraction et changement de résolution.....	5
Stockage des données.....	6
Détection du coeff multiplicateur pour la température	7
Coordonnées d'extraction.....	7
Extraction des valeurs du « pixel poste ».....	7
Calcul du coefficient multiplicateur	8
Calculs des données à 1000m de résolution.....	11
Annexe.....	13
Script cmd3 : clip vaucluse et changement de résolution.....	13
Script extract_multiple_value.sh : enchainement extraction de pixel	15
Script extract_pixel_values.sh : extraction de la valeur d'un pixel	16
Script concat_extracted_data.sh : concaténation de toutes les années pour un lieu et une variable	18
Script R : représentation graphique de la comparaison aux données observées.....	19

Préambule

Description des données brutes

Le Cirame gère un réseau de stations météo automatiques implantées dans les principales zones de production agricole de la région Provence-Alpes-Côte d'Azur. Les données collectées quotidiennement sur ce réseau sont validées puis archivées dans une base de données.

Pour des besoins spécifiques, ces données sont intégrées dans un modèle de spatialisation (logiciel LISDQS implémentant la méthode D. Joly, Univ de Bourgogne¹) qui permet de reconstituer des données climatiques à échelle fine.

Les données ont reçues sur disque externe USB. La description de ces données a été faite par le CIRAME, dans un fichier à la racine des données. Ce fichier « A_LIRE_EN_PREMIER_DONNEES_BSQ_CIRAME_.txt » est copié dans le répertoire **P:/arcjob/cirame**, dans lequel on trouvera aussi tous les fichiers nécessaires pour automatiser la procédure de traitement, dont le script, et les fichiers header et prj.

Pour résumé, les données sont organisées dans 26 répertoires. On trouve 13 répertoires Tn [tmin](2001 à 2013) et 13 répertoires Tx [tmax] (2001 à 2013). Chaque répertoire contient une image par jour estimée par le modèle et un fichier .prm qui contient les informations nécessaires à la lecture et à l'utilisation des données; exemple « 14_130501.prm »:

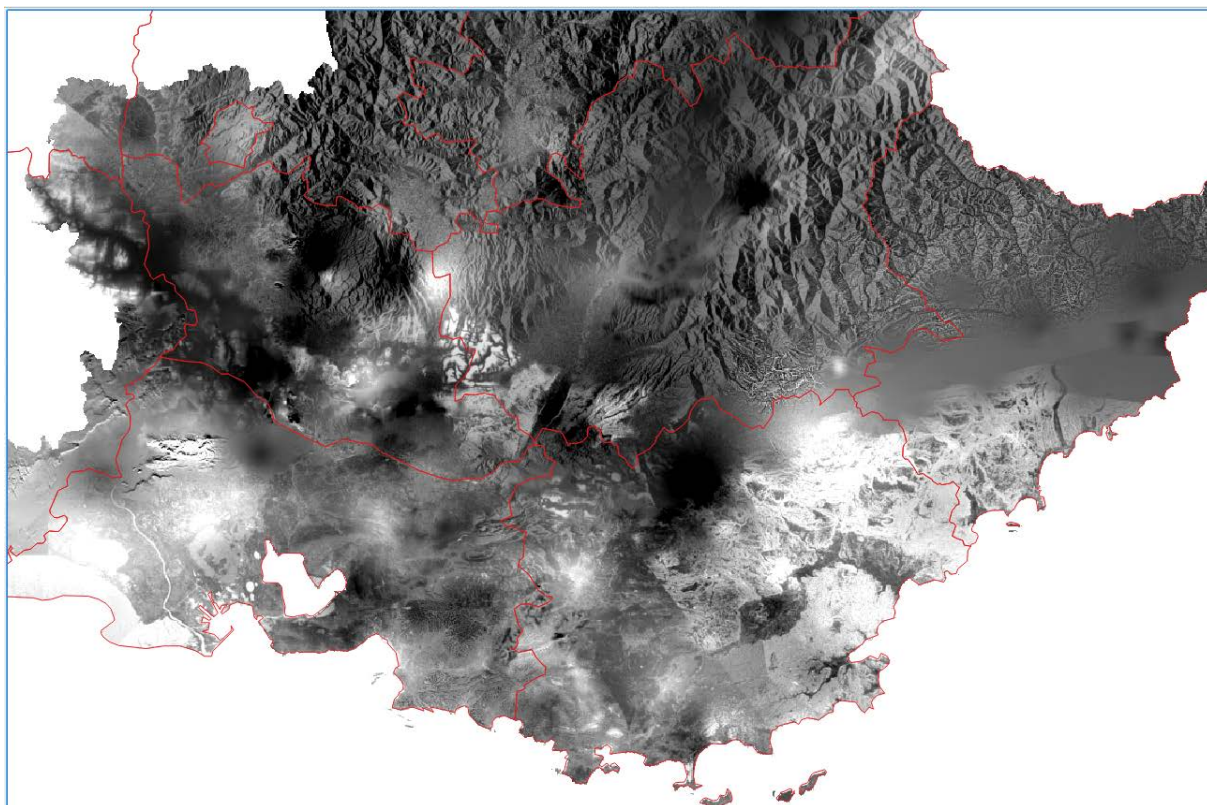
```
4929 5891
2
50 30000
```

L' image est un fichier binaire, contenant 4929 lignes de 5891 colonnes, couvrant toutes la zone PACA. Les valeurs de température sont codées dans des entiers signés de 2 octets ; la maille choisie pour l'interpolation est un carré, de 50m de côté ;le code de donnée manquante est 30 000.

Les valeurs de température sont restituées en $1/10^{\text{ième}}$ ou $1/100^{\text{ième}}$ de degré Celsius (en fonction de la dynamique des valeurs de l'image), sans que l'information ne soit précisée dans le fichier .prm ; elle sera donc à découvrir !

Nota

Pour l'année 2010 en Tmin, on a deux séries: une série de fichiers nommée **12_XXXXXX.bsq** et une autre nommée **Tn_XXXXXX.bsq** (XXXXXX désigne la date). J'ai sollicité Yvan Sivadon, il m'a répondu que les données TN étaient à ignorer car calculées pour un autre contexte.



Identification des paramètres de lecture

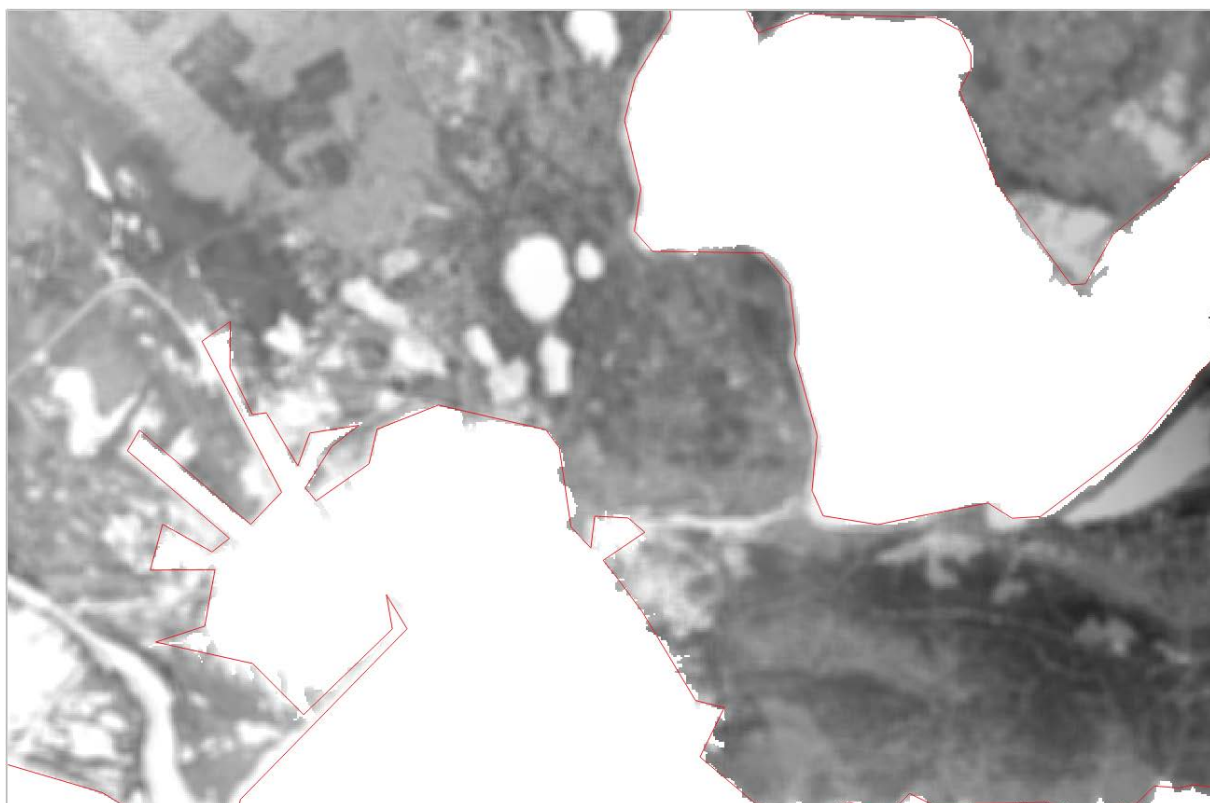
On utilise Qgis pour valider les paramètres de lecture des images. Le principe est de générer un fichier header qui va contenir les informations nécessaires pour permettre l'importation des images.

Ce fichier sera le même pour toutes les images. Il est nommé modèle.hdr dans le répertoire cité au début, et contient :

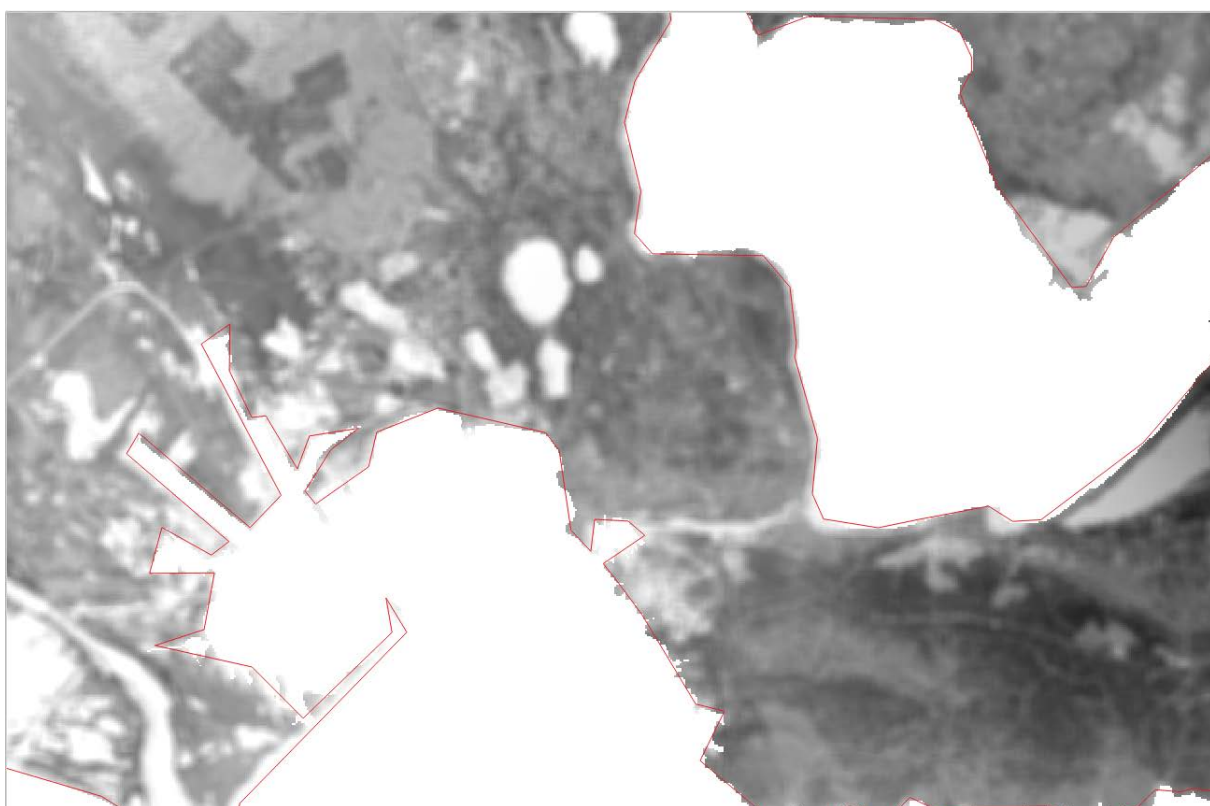
```
ncols 5891  
nrows 4929  
cellsize 50  
xllcorner 736200  
yllcorner 3083150  
nodata_value 30000  
nbits 16  
pixeltype signedint  
byteorder lsbfirst
```

Les coordonnées du coin inférieur gauche sont obtenues par calcul : on retranche à la coordonnée Y du coin supérieur gauche fournie avec les données, la hauteur de l'image. Soit 4929 lignes de 50m.

Avec ce type de données raster, il est impossible de valider avec précision le géoréférencement. J'ai fait le test en donnant un couple de valeur pour le centre du pixel en bas à gauche. Les deux images suivantes, illustrent l'imperceptibilité du décalage engendré.



Zoom sud, coordonnées coin inférieur gauche au centre : $x=736225$, $y=3083175$



Zoom sud, coordonnées coin inférieur gauche : $x=736200$, $y=3083150$

Le bilan est que ce décalage est difficilement perceptible et n'engendrera qu'une erreur minime à la résolution retenue (200m), s'il existe effectivement. Le Cirame n'a pas pu préciser si les données de géoréférencement correspondaient au centre du pixel ou bien au coin.

Pour valider la lecture correcte des données source (lecture d'éléments binaires sur 2 octets pour un entier signé, en big endian), j'ai comparé les résultats avec d'autres outils : ENVI, Matlab et od livré avec l'installation d'osgeo4w.

Matlab

```
%chargement de l'image
x=multibandread('P:\arcjob\cirame\12_010101.bsq',[4929 5891 1],'int16',0,'bsq','ieee-le');
% affichage en éliminant les valeurs < -300 et > 300
imagesc(x,[-300 300])
% affichage valeur minimale dans l'image
min(x(:))
```

Octal Dump

```
bash ; od -t d2 12_010101.bsq
# retourne les valeurs lues dans l'image. Lancée depuis la fenêtre OSgeo4w shell.
# peut être utilisé pour filtrer les valeurs en utilisant un « | grep » à la suite.
```

Extraction et changement de résolution

Pour que le géoréférencement soit complet, il faut préciser le référentiel cartographique. Ceci se fait en ajoutant un fichier prj, qui désigne la projection Lambert III carto.

Notre objectif est d'obtenir des images de températures limitées à la zone vaucluse, et éventuellement à plusieurs résolutions.

La méthode de traitement consiste à lire le fichier source, et en utilisant une routine [gdalwarp](#), faire simultanément, l'extraction de la zone Vaucluse, le changement de résolution, et la sauvegarde en geotiff (plus facile à lire).

Un script, nommé cmd3, a été écrit pour faire cela. Il parcourt les répertoires du disque externe, et pour chacun, liste les fichiers bsq, génère un hdr et un prj avec le même nom que l'image en cours de traitement, et appelle la fonction gdalwarp qui va générer le fichier de sortie dans un répertoire de la machine. Le script est donné en annexe.

Pour l'utiliser, il faut disposer d'un unix avec les commandes standards. Etant exécuté sur une machine windows, l'utilisation de la console «osgeo4w shell» est possible. Il suffit de lancer la commande bash au prompt, pour bénéficier d'un environnement unix proposant les outils systèmes utilisés dans les scripts développés.

Le script cmd3 peut être utilisé avec divers arguments :

- /p/arcjob/cirame/cmd3 -c s : pour extraire en conservant les extractions déjà existantes et la résolution d'origine

- /p/arcjob/cirame/cmd3 -c s Tn_2001 : pour extraire en conservant les extractions déjà existantes et la résolution d'origine sur un seul répertoire
- /p/arcjob/cirame/cmd3 -d 200 : pour extraire en régénérant l'ensemble des données et en dégradant la résolution à 200 m.

En sortie, on obtient les données extraites dans le répertoire désigné par une variable interne du script (\$odir), ainsi qu'un fichier de rapport d'erreurs constatées. En effet, j'ai constaté que parfois l'extraction se passait mal et que le fichier tiff généré était illisible. Ce rapport, s'il existe, contiendra la liste des images extraites mais illisibles. (Voir la variable \$rap_error dans le script)

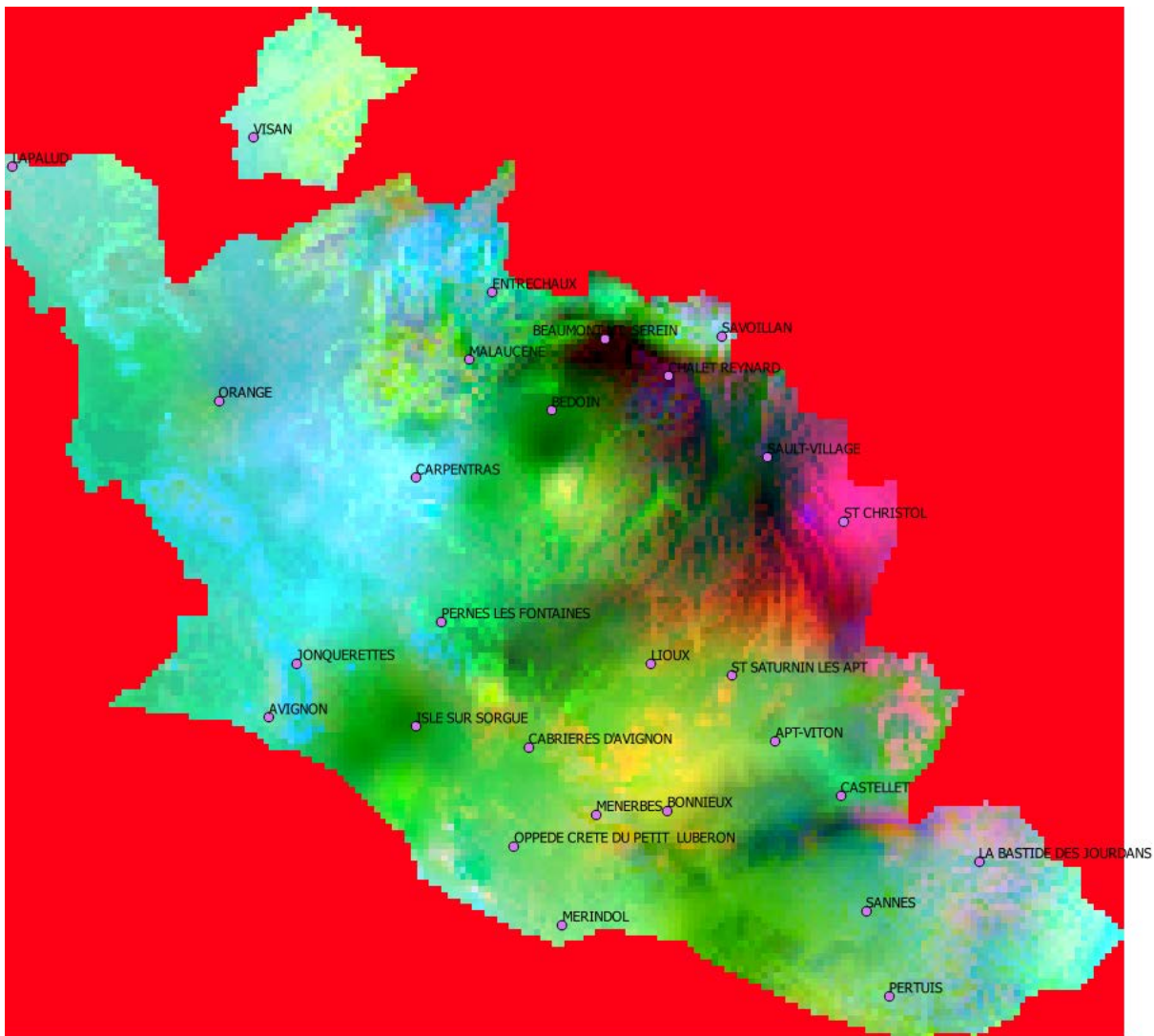


Figure 1: Positions des postes Cirame sur l'image VRT 2001 en composition colorée sur les trois premières dates. Le fond rouge correspond aux NODATA.

Stockage des données

Le résultat des extractions est stocké dans N:\External

Folders\Vigimed\Climat\Données_maillées\cirame\Vaucluse_50m pour les données dans la résolution d'origine, et dans Vaucluse_200m pour les données à la résolution dégradée de 200m.

Les données sources (500 Go) sont toujours sur le disque externe de pclastre.

Détection du coeff multiplicateur pour la température

Lors de la mise en œuvre du logiciel LISDQS par le Cirame, un coefficient multiplicateur est appliqué aux données générées. L'intérêt est de permettre de diminuer le stockage (passage en entier codé sur deux octets), sans perdre trop de précision. Il semble que certaines images utilisent un coefficient à 10 et d'autres à 100. Mais nous n'avons pas l'information. Nous savons que l'ensemble des données journalières de la variable Tn ou Tx d'une année étant traité en une passe, le coefficient est valable pour toutes les images d'une année.

Le principe est de récupérer la valeur d'un pixel « connu » (dans notre cas, le pixel du poste d'Avignon) et d'extraire la valeur stockée dans l'image à cet endroit. En la comparant à la valeur stockée pour le poste météo et la date correspondantes, on pourra déterminer la valeur du coefficient. Les chapitres suivants détaillent donc le processus pour parvenir à déterminer ces coefficients.

Ultérieurement, cette approche de récupération d'une valeur de l'image à une coordonnée précise a été généralisée, et on a obtenu, pour tous les postes Cirame du Vaucluse, les valeurs stockées dans l'image aux coordonnées des postes. Ceci permet de comparer aux valeurs archivées pour les postes et se faire une idée sur la qualité de l'interpolation.

Coordonnées d'extraction

Génération d'une couche de points contenant la position de l'ensemble des postes vaucluse du Cirame. Cette couche est stockée dans la base de données emmah, dans le schéma postes_climatiques, et a pour nom « postes_vaucluse_du_cirame_l3c ». Elle contient en particulier deux colonnes avec les coordonnées en lambert 3 carto. Une version csv de la table attributaire permet d'automatiser l'interrogation de l'ensemble des images pour chaque poste. La couche initiale a été créée à partir d'un fichier de données (Tout sur les stations météorologiques_vaucluse.csv) transmis par Martine.

Extraction des valeurs du « pixel poste »

Le script correspondant est extract_multiple_value.sh. Il va donc lire le fichier de poste et pour chaque couple de coordonnées, lancer un script d'extraction dans l'image (extract_pixel_value.sh) puis concaténer les valeurs obtenues pour toute l'année (les traitements se font année par année). Ce script peut fonctionner pour un site particulier ou pour tous. Ceci se règle en éditant la variable « onlyone ».

Le script extract_pixel_value.sh est donné en annexe. Il utilise les fonctions gdal pour réaliser un VRT contenant toutes les images de l'année, et l'extraction au moyen de la fonction gdallocationinfo. Ce script génère un fichier dans chaque répertoire de donnée (T<n|x>_yyyy_avignon.txt ou yyyy est l'année). Ce fichier ascii précise pour chaque date (365 dates /an = 365 lignes) la valeur de la température générée par le Cirame au point « Avignon » (coordonnées Lambert III Carto : x=803994, y=3182722, précisées en début du fichier extract_pixel_value.sh)

Un autre script, P:\arcjob\cirame\concat_extracted_data.sh, permet alors de fusionner les données journalières extraites. On obtient deux fichiers :

extracted_<xxx>_<yyy>_from_image_Tn_<resol>_<ville>.csv et
extracted_<xxx>_<yyy>_from_image_Tx_<resol>_<ville>.csv).

<xxx> et <yyy> sont les coordonnées du poste sans décimale

<resol> désigne la résolution des images générée à l'étape précédente

<ville> est le nom de la commune associé au poste climatique

Calcul du coefficient multiplicateur

L'obtention du coefficient s'obtient par comparaison aux données sources (observations au poste d'Avignon aérodrome), fournies par Martine. Elles sont stockées dans poste_avignon_INRA_STATION_2001-2013_Tn.csv et poste_avignon_INRA_STATION_2001-2013_Tx.csv.

Le script R (script_compare_image_poste.R) permet de procéder au calcul du coefficient multiplicateur. Le contenu de ce script est donné en annexe. Le coefficient est estimé par approximation (à 10 ou 100) du coefficient du modèle de régression appliqué à l'ensemble des données journalières d'une année.

La série de graphiques suivante illustre les calculs des coefficients de régression pour les différents cas (croisement année et variable). Nous avons vérifié que le changement de résolution n'affectait pas les calculs.

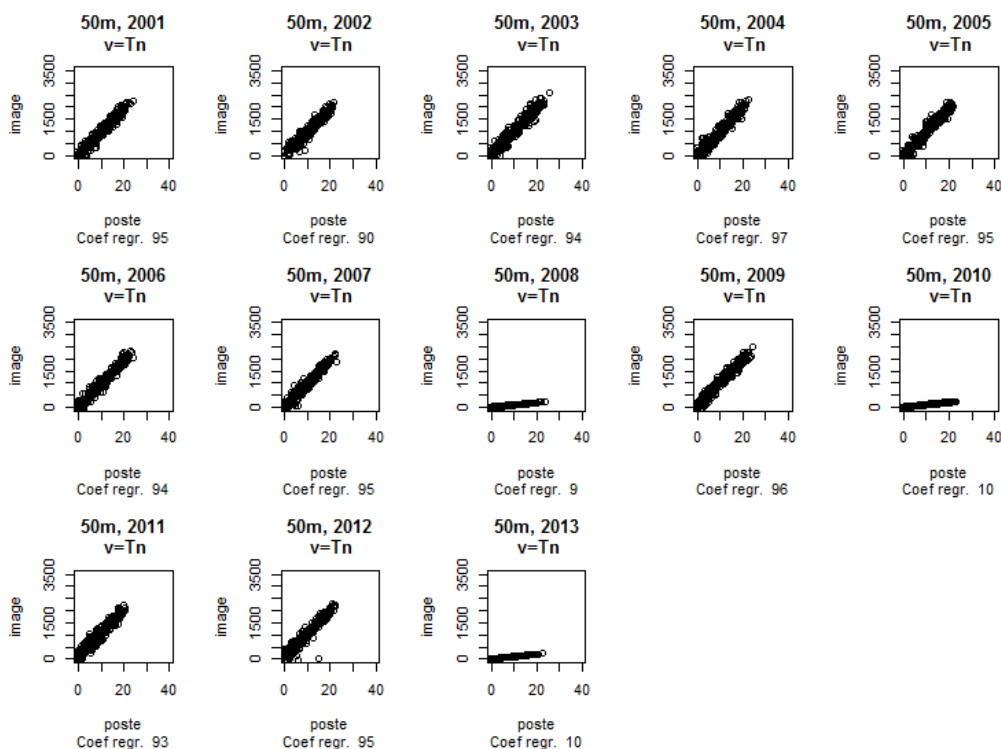


Figure 2 : Régressions par années des valeurs de l'image en fonction de la valeur poste, pour Tmin, résolution 50m

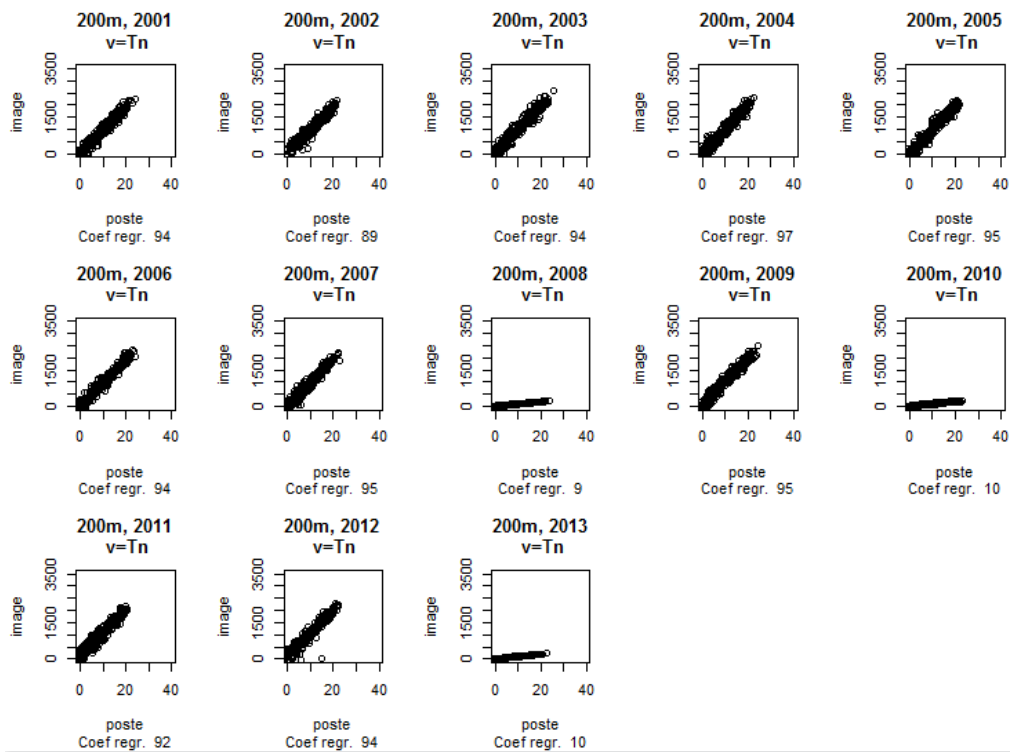


Figure 3 : Régressions par années des valeurs de l'image en fonction de la valeur poste, pour Tmin, résolution 200m

On vérifie que les coefficients des droites sont très peu variables en fonction de la résolution, marquant seulement l'écart attendu entre la température interpolée sur une maille de taille variable et la valeur « vraie » en 1 point (ce qui est rassurant quand aux fonctions utilisées pour faire le ré-échantillonnage).

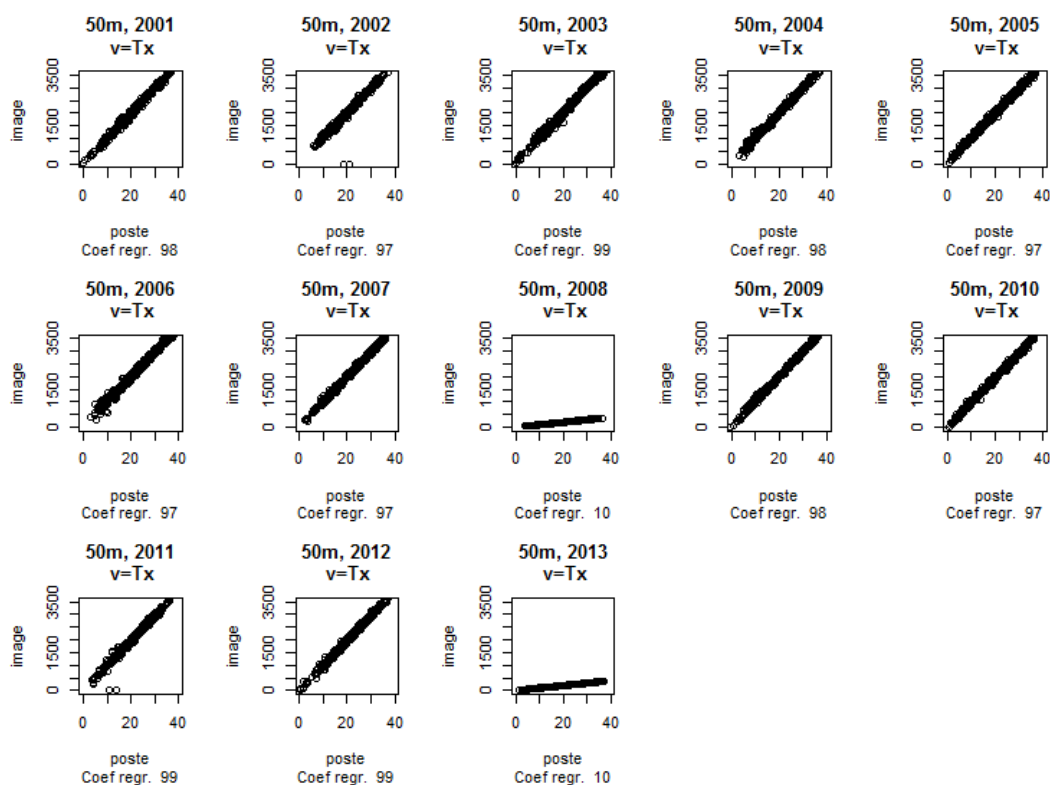


Figure 5 : Régressions par années des valeurs de l'image en fonction de la valeur poste, pour Tmax, résolution 50m

Nota

Les données 2008 ont été recalculées par le Cirame début Juillet 2015 car on obtenait des graphes (X,Y) très dispersés, ainsi qu'un coefficient de régression de l'ordre de 69. Après avoir mis en place ces séries recalculées, les graphes ont été régénérés et permettent d'obtenir des coefficients corrects.

En récapitulant les valeurs des coefficients des droites de régression pour chaque année et chaque variable (Tab. 1), on peut donc déterminer les valeurs du coefficient multiplicateur pour chaque année

An	Tn		Tx	
	coeff cor	CM	coeff cor	CM
2001	95	100	98	100
2002	90	100	97	100
2003	94	100	99	100
2004	97	100	98	100
2005	95	100	97	100
2006	94	100	97	100
2007	95	100	97	100
2008	9	10	10	10

2009	96	100	98	100
2010	10	10	97	100
2011	93	100	99	100
2012	95	100	99	100
2013	10	10	10	10

Tableau 1. Coefficients de régression linéaire (coeff cor) entre valeurs interpolées et valeurs au poste d'Avignon et coefficients multiplicateurs (CM) déduits (résolution 50m).

On constate qu'il vaut très généralement 100, sauf pour 2008, 2010, 2013 pour Tn et 2008 et 2013 pour Tx où il vaut 10

Calculs des données à 1000m de résolution

Les informations suivantes décrivent le processus pour générer les données à 1000m de résolution. Elles contiennent à la fois des commandes et du texte explicatif. Elles sont produites dans ce document à des fins de traçabilité.

1. génération des images à la résolution souhaitée

```
mkdir gridded_data/cirame/Vaucluse_500m
sur pgeom_sig, cd gridded_data/cirame/Vaucluse_50m
Edition script cmd3_pour_1000
Exécution sur pgeom_sig /home/pclastre/nas1-avi/arcjob/cirame/cmd3_pour_1000 -d 1000
```

2. création couche vecteur des mailles à 1000m

```
charger une des images à 1000m dans qgis
Puis "Create a fishnet with the Vector grid tool in the Vector->Research Tools menu.":
vaucluse_vecteur_1000m.shp en exploitant la couche chargée pour les bounding box et la résolution
+ Augmenter de 1000m le minY
+ Diminuer de 1000m le maxX
Sinon, cela rajoute une ligne et une colonne dans le shape
```

Puis "Assign raster values to the polygons with the "Add grid values to shapes" tool in the Processing Toolbox (Simplified interface, GeoAlgorithms, Raster-vector, Raster->vector operations)

Sélectionner les dernières ligne et colonne de mailles car elle est en trop et les supprimer (manip dans l'éditeur d'attribut après sélection, passage en mode édition et clic sur la poubelle)
Avec la table d'attribut, sélectionner les mailles utiles avec l'expression "uneimagedeg" !=0 . et vérifier que l'ensemble des mailles utiles sont bien sélectionnées (le cas échéant modifier l'expression)
Sauvegarder les mailles ainsi sélectionnées dans une nouvelle couche qui sera donc la liste des mailles utiles à la résolution 1000m
Résultat définitif dans la couche mailles_1000m_vaucluse.shp

On obtient la liste des mailles "1000m" par la table d'attribut (ouvrir le fichier dbf avec excel, virer toutes les colonnes sauf la première (ID) et enregistrer le csv: mailles_1000m_vaucluse.csv
Supprimer la première ligne

3. construction des vrt

```
Exécution sur pgeom_sig
cd gridded_data/cirame/Vaucluse_1000m
source /home/pclastre/nas1-avi/arcjob/cirame/build_vrt.sh -d
```

4. génération des données température sous forme de fichier ascii, par maille

Dans Rstudio, source le script script_traitement_image_gdal.R après édition du nom du fichier liste des mailles et modification de la résolution. A cette occasion, j'ai observé un décalage entre les numéros des mailles dans le shape et l'indice à utiliser pour aller chercher la bonne valeur dans les images. D'où l'apparition d'une variable npos. En fait cela vient du fait que la première maille à pour indice 0 !

5. Publication et métadonnées

Intégration de la couche shape dans la base postgis emmah (schéma cirame), avec le SRID 2154 (Lambert 93)
Intégration de la couche dans le geoserver GeOpen4S pour publication WMS, WFS.
Création de la fiche de métadonnées pour la nouvelle couche à 1000m.
Mise à jour du projet lizmap pour publication cartographique, accessible [ici](#).
Mise à jour du présent document et copie de sa version pdf
dans /home/donnees/source_ar/0_InterfaceWeb/gn_plateau_ressources

Annexe

Script cmd3 : clip vaucluse et changement de résolution

```
# script d'extraction des données climatiques vaucluse à partir des données cirame
# Utilisation de la fonction gdalwarp
# Lancement via le shell installé avec osgeo4w, et après exécution du bash, taper source cmd3
# Lit les fichiers sur le disque externe (/j) et les dépose dans un répertoire (variable $odir)
# La couche masque_vaucluse.* est nécessaire pour faire le clip (sinon, enlever les options
cutline)
# Le fichier modele.hdr a été mis au point en utilisant qgis, par tatonnement
# le fichier modele.prj est celui de la couche masque_vaucluse, renommé
# Les informations détaillées sur les données sources, sont dans le readme.
# Deux versions en une. Si besoin de données à la résolution d'origine, décommenter la
deuxième commande gdalwarp
# VARIABLES
# dirs: liste des répertoires source à traiter
# odir: répertoire de sortie (base de l'arborescence)
# d: un répertoire source à traiter
# file: liste des fichiers bsq présents dans le répertoire $d
# f: un fichier bsq à traiter
# res: résolution de l'image en sortie (permet de passer des 50m originaux à une résolution
dégradée
#
# se mettre dans un bash
# se placer dans le répertoire des données source /j/CIRAME_DONNEES_INTERPOLEES
# puis lancer source /p/arcjob/cirame/cmd3 -c (par exemple)
#

if [ $# -lt 2 ]
then
    echo "Usage $0 {-c|-d} {s|200|300..} <rep>"
    echo "Arguments obligatoires:"
    echo "-c: si le fichier existe déjà, on le conserve"
    echo "ou"
    echo "-d: si le fichier existe déjà, on l'efface"
    echo "s|200|300..: s pour sans dégradation de la résolution, 200 pour
rééchantillonnage à 200m ..."
    echo "Arguments optionnels:"
    echo "<rep>: pour ne traiter qu'un des sous répertoire (rep en l'occurrence)"
    return
fi

if [ "$1" = "-c" ]
then
    mode=conserve
    echo "Mode conservation des données"
else
    mode=delete
    echo "Mode suppression des données déjà existantes"
```

```

fi

if [ "$2" = "s" ]
then
    moderesol=1 # on ne change pas la résolution
    echo "Pas de changement de résolution"
else
    moderesol=2
    res=$2
    echo "Changement de résolution: $res"
fi

if [ $# -eq 3 ]
then
    onerep=1
    dirs=$3
    echo "Traitement sur un répertoire unique: $dirs"
else
    onerep=0
    dirs=`ls`
fi

odir=/d/Nosafe/cirame
rap_error=/p/arcjob/cirame/erreurs_creation_fichiers_tif_${res}.txt

if [ -e $rap_error ]
then
    rm $rap_error
fi

for d in $dirs
do
    echo ""
    echo "Traitement répertoire $d"
    if [ ! -d $odir/$d ]
    then
        mkdir $odir/$d
    fi
    cd /j/CIRAME_DONNEES_INTERPOLEES/$d
    file=$(ls *.bsq)
    for f in $file
    do
        fic="{f%.*}"
        echo -n $fic " "
        if [ ! -e $fic.hdr ]
        then
            cp /p/arcjob/cirame/modele.hdr $fic.hdr
        fi
        if [ ! -e $fic.prj ]
        then
            cp /p/arcjob/cirame/modele.prj $fic.prj
        fi
    fi
done

```



```

if [ -e $odir/$d/$fic.tif -a "$mode" = "conserve" ]
then
    echo "Tif conservé"
else
    if [ -e $odir/$d/$fic.tif ]
    then
        rm $odir/$d/$fic.tif
    fi
    if [ $moderesol -eq 2 ]
    then
        # Avec changement de résolution
        gdalwarp -of Gtiff -tr $res $res -cutline
/p/arcjob/cirame/masque_vaucluse.shp -crop_to_cutline -overwrite $f $odir/$d/$fic.tif
    else
        # Sans changement de résolution
        gdalwarp -of Gtiff -cutline
/p/arcjob/cirame/masque_vaucluse.shp -crop_to_cutline -overwrite $f $odir/$d/$fic.tif
    fi
    # controle qualité
    gdalinfo $odir/$d/$fic.tif > /dev/null
    if [ $? -ne 0 ]
    then
        echo "Erreur sur $d / $fic.tif " >> $rap_error
        echo "gdalwarp -of Gtiff -cutline
/p/arcjob/cirame/masque_vaucluse.shp -crop_to_cutline -overwrite $f $odir/$d/$fic.tif" >>
$rap_error
    fi
fi
done
done

```

Script `extract_multiple_value.sh` : enchainement extraction de pixel

script d'extraction multiple de valeur à une coordonnée précisée dans un fichier
se placer manuellement dans le bon répertoire à traiter
#cd /n/External Folders/Vigimed/Climate/Données_maillées/cirame/Vaucluse_50m

```

fic=/p/arcjob/cirame/postes_vaucluse_du_cirame_v2.csv
debut=1
onlyone=AVIGNON # pour ne faire le calcul que sur un seul poste
#onlyone=tous # pour faire le calcul sur tous les postes
while read record; do
    if [ $debut -eq 0 ]
    then
        poste=$(echo $record | cut -d';' -f3 | sed -e "s/[- ']/_/g")
        if [ "$onlyone" = "tous" -o "$poste" = "$onlyone" ]
        then
            x=$(echo $record | cut -d';' -f9 | sed -e 's/\([0-9][0-9]*\)\.\([0-9][0-9]*\)\([0-9][0-9]*\)/\1/')
            y=$(echo $record | cut -d';' -f10 | sed -e 's/\([0-9][0-9]*\)\.\([0-9][0-9]*\)\([0-9][0-9]*\)/\1/')
            echo "Extraction pour le poste $poste, aux coordonnées: $x, $y"
        fi
    fi
done

```

```

        echo "Extraction ..."
        source /p/arcjob/cirame/extract_pixel_value.sh -d $x $y $poste
        echo "Concaténation ..."
        source /p/arcjob/cirame/concat_extracted_data.sh $x $y $poste
    fi
else
    debut=0
fi
done < $fic

```

Script `extract_pixel_values.sh` : extraction de la valeur d'un pixel

```

# traitement pour détecter le coefficient multiplicatif sur les images en
# températures

# Utilisation:
# se placer dans le répertoire racine de tous les répertoires annuels
# cd /d/Nosafe/cirame
# se lance depuis un bash sous shell OSGEO4W par la commande:
# source /p/arcjob/cirame/extract_pixel_value.sh
#
# le script fait la liste de ces répertoires et fait un cd dans chacun
# pour chaque répertoire,
# fait la liste des fichiers
# génère un fichier raster virtuel pour traiter le paquet d'un coup
# lance l'extraction pour la coordonnée d'avignon, en filtrant la sortie
#

# variable désignant le petit bout de code awk qui permettra de fusionner les deux lignes
# après l'extraction de la valeur d'un pixel
# à savoir, la date , et la valeur
script_awk=/p/arcjob/cirame/fusion_deux_lignes.awk

# controle des arguments et affichage du usage
if [ $# -ne 4 ]
then
    echo "Script d'extraction d'une valeur raster dans une série de VRT images"
    echo "Script adapté aux données du cirame"
    echo
    "=====
    echo "Usage: $0 {-c|-d} x y label"
    echo "Option -c: ne fait pas le recalcul si le fichier de sortie existe déjà"
    echo "Option -d: Force l'effacement du fichier de sortie s'il existe déjà"
    echo "      Il faut préciser obligatoirement l'un de ces deux choix"
    echo " x: position en X de la valeur à extraire"
    echo " y: position en Y de la valeur à extraire"
    echo " Aucun controle n'est effectué sur les valeurs données en argument"
    echo " label: nom du point d'extraction"
    echo "=====
    echo "Attention, tous les argument sont obligatoires"
    exit
fi
if [ "$1" = "-c" ]

```

```

then
    mode=conserve
else
    mode=delete
fi
x=$2
y=$3
label=$4
#echo "Coordonnées d'extraction: $x $y; Place: $label"

# coordonnées à extraire
#x=803994
#y=3182722

dirs=$(ls)

for d in $dirs
do
    cd $d
    echo "Traitement répertoire $d"

# création liste des fichiers
if [ ! -e liste.txt ]
then
    ls *.tif > liste.txt
fi
# création d'un VTR dans le répertoire 2001
if [ ! -e $d.vrt ]
then
    gdalbuildvrt -separate -input_file_list liste.txt $d.vrt
fi
echo "Vrt ok"

file=${d}_${label}_${x}_${y}.txt
if [ -e $file -a "$mode" = "conserve" ]
then
    echo "Extract conservé"
else
    if [ -e $file ]
    then
        rm $file
    fi
    # extraction d'une valeur à une coordonnée précise:
#    gdallocationinfo -l_srs epsg:27573 -valonly $d.vrt $x $y > ${d}_${label}_${x}_${y}.txt
#    gdallocationinfo -l_srs epsg:27573 $d.vrt $x $y | egrep -v 'Report|Band|Location:' |
sed -e 's/.*[T1][xn0-9]_\([0-9][0-9]*\)\.tif.*\1/' | sed -e 's/Value:/' | awk -f $script_awk >
${d}_${label}_${x}_${y}.txt
    echo "Extract généré"
fi
cd ..
done

```

Script concat_extracted_data.sh : concaténation de toutes les années pour un lieu et une variable

```
# contactation des fichiers issus du processus d'extraction des valeurs de pixel en avignon
# doit être exécuté depuis le répertoire au dessus de toutes les années
# cd /n/External Folders/Vigimed/Climate/Données_maillées/cirame/Vaucluse_50m
# ou cd /n/External Folders/Vigimed/Climate/Données_maillées/cirame/Vaucluse_200m
#
# se lance depuis un bash sous shell OSGEO4W par la commande
# source /p/arcjob/cirame/concat_extracted_data.sh
#
# récupération de la résolution dans le nom du répertoire courant
resol=$(pwd | sed -e 's/.*Vaucluse_\([0-9][0-9]*\)m\/\1/')

# controle des arguments et affichage du usage
if [ $# -ne 3 ]
then
    echo "Script de concaténation après extraction"
    echo "Script adapté aux données du cirame"
    echo
    "=====
    echo "Usage: $0 x y label"
    echo " x: position en X de la valeur à extraire"
    echo " y: position en Y de la valeur à extraire"
    echo " Aucun controle n'est effectué sur les valeurs données en argument"
    echo " label: nom du point d'extraction"
    echo "=====
    echo "Attention, tous les argument sont obligatoires"
    exit
fi
x=$1
y=$2
label=$3
#echo "Coordonnées d'extraction: $x $y; Place: $label"

#echo "Résolution: $resol, Coordonnées extraite: $x $y"

# on controle qu'on a pu lire la résolution et donc qu'on est dans un des deux répertoires
possibles
if [[ $resol != "50" && $resol != "200" ]]
then
    echo "Impossible de poursuivre, resolution detectee = ;;${resol};;"
    exit
fi

dirsTn=$(ls -d Tn*)
dirsTx=$(ls -d Tx*)

destTn=/p/arcjob/cirame/extracted_${x}_${y}_from_image_Tn_${resol}m_${label}.csv
destTx=/p/arcjob/cirame/extracted_${x}_${y}_from_image_Tx_${resol}m_${label}.csv

if [ -e $destTn ]
```

```

then
  rm -f $destTn
else
  touch $destTn
fi
if [ -e $destTx ]
then
  rm -f $destTx
else
  touch $destTx
fi

for d in $dirsTn
do
wc=$(wc -l $d/$d_${label}_${x}_${y}.txt | awk '{ print $1}')
if [ $wc -ne 365 -a $wc -ne 366 ]
then
  echo "ERREUR sur $d/$d_${label}_${x}_${y}.txt: pas la bonne taille $wc"
  return
fi
#echo "traitement donnees de $d"
cat $d/$d_${label}_${x}_${y}.txt >> $destTn
#cat $d/$d_avignon.txt >> $destTn

done

for d in $dirsTx
do
wc=$(wc -l $d/$d_${label}_${x}_${y}.txt | awk '{ print $1}')
if [ $wc -ne 365 -a $wc -ne 366 ]
then
  echo "ERREUR sur $d/$d_${label}_${x}_${y}.txt: pas la bonne taille $wc"
  return
fi

#echo "traitement donnees de $d"
cat $d/$d_${label}_${x}_${y}.txt >> $destTx
#cat $d/$d_avignon.txt >> $destTx
done

echo "Concaténation terminée."

```

Script R : représentation graphique de la comparaison aux données observées

```

# pcl, Juin 2015, R studio sur pclim142
#
# traitement données: préciser la variable (Tn ou Tx) et la résolution (50 ou 200)
variable<-'Tx'
resolution<-50

# création des noms des fichiers attendus

```

```

#ficposte<-paste('poste_avignon_INRA_STATION_2001-2013_',variable,'.csv',sep='')
ficposte <- 'poste_aeroport_01a13.csv'
ficimage<-
paste('extracted_806014_3182211_from_image_',variable,'_',toString(resolution),'m_AVIGN
ON.csv',sep='')
#
mptn<-read.table(ficposte,sep=";",header=TRUE)
# changement du nom de la colonne des données pour uniformiser la suite
#c4<-names(mptn)[4]
names(mptn)[names(mptn)==variable]<- "poste"

# chargement extractions depuis les images
itn <- read.table(ficimage,sep=";",header=FALSE)
# on renomme les colonnes dans le dataframe de l'image
if (length(names(itn)) > 1) {
  names(itn)[names(itn)=='V1']<-'date_image'
  names(itn)[names(itn)=='V2']<- 'image'
}
else {
  names(itn)[names(itn)=='V1']<- 'image'
}
# et on fusionne toutes les données dans le même dataframe
mptn<-cbind(mptn,itn)

# affichage des graphe croisés, par année, avec calcul du coefficient de regression
ans<-sort(unique(mptn$AN))
par(mfrow=c(3,5))
plot(mptn$poste[mptn$AN==ans[1]],
mptn$image[mptn$AN==ans[1]],xlab='poste',ylab='image')
reg<-lm(mptn$image[mptn$AN==ans[1]] ~mptn$poste[mptn$AN==ans[1]])
title(paste(toString(resolution),'m, ',toString(ans[1]),'\n v=',variable,sep=""), sub=paste('Coef
regr. ',toString(round(coef(reg)[2])))

for(i in 2:length(ans)) {
  plot(mptn$poste[mptn$AN==ans[i]],
mptn$image[mptn$AN==ans[i]],xlab='poste',ylab='image')
  reg<-lm(mptn$image[mptn$AN==ans[i]] ~mptn$poste[mptn$AN==ans[i]])
  title(paste(toString(resolution),'m, ',toString(ans[i]),'\n v=',variable,sep=""), sub=paste('Coef
regr. ',toString(round(coef(reg)[2])))
}

```