Validation de méthodes de correction de biais multivariées et applications à des processus environnementaux

Lingeshwari RAMLUGON

Rapport de stage dans le cadre du Diplôme Universitaire de Technologie en Statistique et Informatique Décisionnelle à l'Institut Universitaire de Technologie d'Avignon.

27 avril 2020 au 01 juillet 2020

Institut Universitaire de Technologie d'Avignon
Campus Jean-Henri Fabre337 chemin des MeinajariesBP 61207, 84911 Avignon

Remerciements

Je tiens à remercier Denis Allard, Directeur de Recherche à INRAE et mon maître de stage, qui m'a donné l'opportunité de faire mon stage dans l'unité BioSP, malgré la crise sanitaire. Je remercie aussi Emily Walker, Ingénieure de recherche à l'unité BioSP pour ses conseils et Iñaki Garcia de Cortazar Atauri, Ingénieur de recherche à l'unité AgroClim, de m'avoir guidé lorsque j'en avais besoin.

Je remercie également Florian Garrel, mon camarade de classe avec qui j'ai réalisé ce stage, pour sa coopération et son investissement. Toutes les missions du stage ont été réalisées en binôme.

Mon stage à INRAE m'a permis d'élargir mes connaissances et d'améliorer mes compétences en programmation R. J'ai pu avoir un encadrement excellent, malgré que le stage a été grandement réalisé en télétravail.

Résumé

Mon stage de deuxième année de DUT STID a été effectué dans l'unité BioSP à INRAE à Agroparc, Avignon. J'ai travaillé en binôme avec mon camarade de classe, Florian Garrel, et nous étions sous la tutelle de Denis Allard et Emily Walker. Toutes les missions ont été réalisées en commun avec F. Garrel, il n'y a pas eu de travail distinct durant notre collaboration, sauf pour la rédaction du rapport. Notre stage s'insère dans le cadre du projet COMPROMISE, qui est mené en collaboration avec le Laboratoire des Sciences du Climat et de l'Environnement (LSCE, CNRS/CEA). Le but du projet consiste à comparer les réponses des modèles environnementaux pour différentes méthodes de correction multivariés (MCM) des séries climatiques et les comparer à la méthode univariée (MCU), CDF-t. Il s'agit de calculer un certain nombre d'indicateurs agroclimatiques à partir de séries climatiques issues du modèle de climat IPSL-CM5. Ce modèle présente des biais, qui sont corrigés en comparant les sorties calculées sur des périodes passées aux valeurs qui ont été mesurées sur ces mêmes périodes. A ce jour, chaque variable climatique est corrigée à tour de rôle, indépendamment les unes des autres, site par site et jour par jour (correction univariée). Cette façon de faire a tendance à ne pas respecter la structure de dépendance des variables, car en réalité, les variables climatiques sont liées entre elles, dans le temps ou dans l'espace. Le fait d'ignorer cette dépendance peut entraîner des conséquences importantes lorsque plusieurs variables climatiques sont implémentées dans les modèles de projection climatiques. C'est pour cette raison que les statisticiens développent actuellement des méthodes de correction multivariées, qui ont pour but de respecter les dépendances inter-variables, spatiales, temporelles ou spatio-temporelles. Nous disposons de variables climatiques (5 variables: Température, Précipitations, Humidité, Vent, Rayonnement) pour 3 régions (Bretagne, Ile-de-France et Provence). Le stage se déroule en deux parties. Dans la première, nous calculons les indicateurs agroclimatiques (3 indicateurs : l'évapotranspiration, le bilan hydrique et la phénologie) pour 3 cultures (blé, maïs, vigne). Dans la deuxième partie, nous calculons l'indice forêt-météo pour chaque région. Le stage comporte donc deux phases distinctes et pour chaque phase, il y a deux parties: (i) préparer les données climatiques pour qu'elles puissent calculer les indicateurs; (ii) écrire des scripts qui permettent de visualiser et comparer ces indicateurs à travers les différentes méthodes de corrections.

Sommaire

Présentation d'INRAE **4**

Introduction **5**

Partie I: Indicateurs Agroclimatiques **7**

A. Préparation des données 7

1) Les répertoires et fichiers de données 7

2) Formatage des données 11

3) Modification des scripts R 13

4) Résultats des calculs: Les indicateurs agroclimatiques 16

B. Création de fonctions 19

1) Boxplot - Indicateur: Phénologie 19

2) Boxplot - Indicateur: Bilan hydrique, Evapotranspiration 20

3) Cartographie - Indicateur: Bilan hydrique, Evapotranspiration 21

4) Cartographie - Indicateur: Phénologie 22

C. Visualisation des indicateurs agroclimatiques 22

1) Cartographies 23

2) Observations 41

Partie II. Indicateur de risque de feux de forêt **44**

A. Création des données 44

B. Création d'une fonction 46

C. Visualisation de l'indice forêt-météo 46

1) Cartographies 46

2) Observations 53

Bilan **54**

Annexe **55**

Sources **116**

Présentation d'INRAE

L'Institut National de Recherche en Agronomique, a été fondée en 1946 avec pour but d'améliorer les techniques d'agricultures en France. Depuis le 1er janvier 2020, INRA a fusionné avec IRSTEA (institut national de recherche en sciences et technologies pour l'environnement et l'agriculture) pour devenir INRAE (institut de recherche en agronomie, alimentation et environnement). INRAE est un organisme national de recherche publique qui est sous la tutelle du ministère de l'Enseignement supérieur et de la Recherche et du Ministère de l'agriculture, de l'agroalimentaire et de la forêt. Selon son décret fondateur (n° 2019-1046 du 10 octobre 2019), l'institut :

- « 1° Produit, publie et diffuse les connaissances scientifiques résultant de ses travaux de recherche et d'expertise;
- 2° Organise, en l'absence de dispositions ou clauses contraires, l'accès libre aux données scientifiques et aux publications;
- 3° Concourt à l'élaboration de la politique et de la stratégie nationale et européenne de recherche;
- 4° Apporte son concours à l'enseignement supérieur et à la formation à la recherche et par la recherche dans ses domaines de compétence;
- 5° Établit et met en œuvre des partenariats avec les établissements d'enseignement supérieur et de recherche;
- 6° Contribue, par la valorisation de ses compétences, de ses savoir-faire et des résultats de la recherche, à la conception d'innovations technologiques et sociales;
- 7° Contribue au développement de la capacité d'expertise scientifique et technologique, conduit des expertises et contribue aux activités de normalisation, en appui aux politiques publiques, aux fondations et associations reconnues d'utilité publique, pour répondre aux enjeux du développement durable;
- 8° Participe aux débats qui portent sur la place de la science et de la technologie dans la société. »

L'institut compte 18 centres de recherche en France métropolitaine et 14 départements scientifique. INRAE Provence-Alpes-Côte d'Azur se place au 4ème rang national parmi les 18 centres INRAE avec un budget consolidé de 71 M€, dont 14 M€ de ressources propres. Il rassemble 1300 agents, dont 710 agents permanents localisés sur 13 sites. J'effectue mon stage dans l'unité BioSP, sur le site d'Avignon. L'unité BioSP est sous la tutelle du département MIA (mathématiques et informatique appliquées) et du département SPE (Santé des plantes et environnement). L'unité BioSP existe sur le centre INRAE d'Avignon depuis 1975 (rattachement au département de Biométrie en 1978). A sa création, son activité de recherche s'est d'abord concentrée sur les statistiques spatiales (géostatistique et géométrie stochastique) avec des intérêts appliqués pour l'environnement physique (sol, climat, etc). Depuis les années 2000, l'activité scientifique de BioSP s'est diversifiée, notamment par ses recrutements, avec (i) une part croissante de questions thématiques en écologie, épidémiologie et biologie des populations et (ii) l'analyse, l'utilisation et la simulation de systèmes dynamiques. La diversité des disciplines présentes à BioSP aujourd'hui permet d'aborder des questions ayant toutes trait à la modélisation des phénomènes spatiaux avec des approches variées, souvent complémentaires. Pour institutionnaliser cette orientation, l'unité a pris le nom « Biostatistique et Processus Spatiaux » en 2006.

Introduction

L'Institut National de la Recherche en Alimentation, Agronomie et Environnement (INRAE) cherche à évaluer l'impact du changement climatique sur l'agriculture dans les années à venir et développe des mesures d'adaptation de l'agriculture et de la forêt au changement climatique. Pour cela, en 2014, INRA a lancé le programme ACCAF (Adaptation de l'agriculture et de la forêt au changement climatique). ACCAF finance des projets avec des partenaires, dont le projet COMPROMISE, qui est une collaboration avec le Laboratoire des Sciences du Climat et de l'Environnement (LSCE).

Le changement climatique affecte la croissance des cultures, le rendement, ainsi que les récoltes. Cet impact est évalué à l'aide de modèles de processus (modèles de bilan hydrique, modèles de croissance de plantes, indicateurs de sécheresse,etc.), en utilisant les variables climatiques (jusqu'à 5 variables : Température, Précipitations, Humidité, Vent, Rayonnement) issues des modèles de climat produits par les climatologues.

Les modèles de climat sont connus pour avoir des biais qui doivent être corrigés. La correction est faite en comparant les variables climatiques calculées par les modèles climatiques à la valeur qui a été observée (dans le passé). Le but de cette correction est d'assurer que les statistiques des variables climatiques simulées (telles que la moyenne et l'écart-type) correspondent aux statistiques des variables observées. Jusqu'à présent, la correction des biais est faite par des méthodes de correction univariées, qui corrigent les variables climatiques sans préserver les structures de dépendances inter-variables, spatiales, temporelles ou spatio-temporelles. La méthode de correction univariée qui est la plus utilisée dans la recherche environnementale, est la méthode CDF-t. Le fait de ne pas préserver les structures de dépendances, peut être conséquent si un grand nombre de variables climatiques sont implémentées dans les modèles climatiques. Si c'est le cas, les modèles de climat peuvent produire des variables climatiques contenant trop de biais. Par la suite, ces variables climatiques, qui sont utilisées pour évaluer l'impact du changement climatique sur les processus agricoles, produisent des résultats biaisés.

Afin de réduire les biais des modèles climatiques, les statisticiens développent des méthodes de correction de biais multivariées (MCM). Le projet COMPROMISE consiste à comparer les indicateurs climatiques calculés par des variables climatiques auxquelles ont été appliquées différentes méthodes de correction.

Dans ce rapport, nous considérons, 1 méthode de correction univariée: Cumulative Distribution Function transfer, CDF-t et 2 méthodes de correction multivariées: Rank Resampling for Distributions and Dependences, R2D2 (Vrac, 2018) et Dynamical Optimal Transport Correction, dOTC (Robin et al., 2019).

A chaque méthode de correction multivariée, s'appliquent 3 protocoles qui permettent d'ajuster les variables simulées aux variables observées, en préservant les interdépendances des variables. Les protocoles sont: 5D, spatial et full.

Le protocole 5D, aussi appelé inter-variable, a pour but de corriger les corrélations inter-variables dans chaque cellule de la grille[(1)].

Le protocole spatial, a été conçu pour ajuster les corrélations spatiales des modèles climatiques pour chaque variable physique considérée indépendamment des autres.

Le protocole full est fait pour corriger simultanément les corrélations spatiales et inter-variables des simulations.

Nous considérons deux périodes de temps, 1985 à 2014 (passé) et 2036 à 2065 (futur). Pour la période passée, nous disposons de variables climatiques calculées selon les différentes méthodes de correction et protocoles, et les variables climatiques issues des données de réanalyse "Système d'Analyse Fournissant des Renseignements Atmosphériques à la Neige" (SAFRAN), que nous considérons comme notre référence. Pour la période future, nous disposons uniquement des variables climatiques calculées selon les différentes méthodes de correction et protocoles.

Pour chaque variable climatique, nous avons donc celle calculée par:

- 1CDF-t (MCU),
- SpatialR2D2 (MCM R2D2 avec protocol spatial),
- SpatialdOTC (MCM dOTC avec protocol spatial),
- IntervarR2D2 (MCM R2D2 avec protocol 5D),
- IntervardOTC (MCM dOTC avec protocol 5D),
- SpatIntervarR2D2 (MCM R2D2 avec protocol full),
- SpatIntervardOTC (MCM dOTC avec protocol full), et celle de SAFRAN.

Les variables climatiques que nous avons sont celles qui ont été observées et calculées sur 3 régions de France: Bretagne, Ile-de-France et Provence. [(1)]Chaque région est représentée par un quadrillage (grille) de plusieurs pixels (sites) et le nombre de sites varie d'une région à une autre. Nous avons pour chaque région, chaque correction et chaque période, dans un pixel, 5 variables climatiques.

Pour comparer les méthodes de corrections, nous évaluons leurs impacts sur les indicateurs climatiques.

Dans la première partie dédiée aux indicateurs agroclimatiques, nous calculons la phénologie, le bilan hydrique et l'évapotranspiration. Pour cela, 3 cultures ont été choisies: blé, maïs et vigne. Dans la deuxième partie, des feux de forêts, nous calculons l'Indice Forêt Météo.

Pour récapituler, nous avons des répertoires avec:

- Les données de réanalyse SAFRAN d'une grille de 8 km par 8 km récoltées entre 1985 et 2014,
- Les données corrigées avec les méthodes de correction de biais CDF-t, R2D2 et dOTC pour les périodes 1985-2014 et 2036-2065, selon les 3 protocoles : 5D, spatial, full,

ainsi que des scripts de calculs qui permettent de calculer les indicateurs agroclimatiques (phénologie, bilan hydrique, évapotranspiration). Il s'agit maintenant de préparer les données, de calculer des indicateurs et de les visualiser.

Partie I: Indicateurs Agroclimatiques

Nous calculons les stades phénologiques, le bilan hydrique et l'évapotranspiration pour chaque culture dans chaque région.

Chaque culture a une variété, le maïs et le blé ont les variétés précoce et tardive; la vigne a les variétés chardonnay, syrah et cabernet-sauvignon. Le rendement des cultures dépend de la qualité du sol (sol sec, humide, saturé, profond, peu profond).

Nous avons à notre disposition des scripts de calcul écrits par l'unité AgroClim d'INRAE, et les variables climatiques corrigées pour chaque région et chaque période, calculées par le LSCE. Les scripts de calcul permettent de calculer les indicateurs pour les différentes variétés et différents types de sol (jusqu'à 3 types de sols). Nous considérons cependant dans notre étude seulement la variété précoce (pour le maïs et le blé) et la variété chardonnay (pour la vigne), et le sol de type 1.

La première étape consiste à comprendre la structure des données climatiques ainsi que les scripts de calcul.

A. Préparation des données

Chaque région est représentée par une grille. Pour la Bretagne, par exemple, nous avons une grille de dimension 21 lignes par 20 colonnes, soit 420 pixels. Dans RStudio, la grille est visualisée comme un dataframe ou une matrice. Parmi ces sites, il y en a certains qui ne contiennent pas de valeurs climatiques (pixels avec NaN).

N.B: Les répertoires sont détaillés dans cette phase, car le nom de chaque répertoire contient des informations qui sont nécessaires pour faire tourner les scripts de calcul.

1) Les répertoires et fichiers de données

Nous avons un répertoire initial qui contient 4 dossier: Code, MBC_Bretagne, MBC_Provence, MBC_IdF (renommé en MBC_Ile_de_France).

Dans le répertoire Code, nous avons plusieurs scripts de calcul. Dans le cadre de ce projet, les scripts utilisés et leurs descriptions:

Script R	Apellation	Description
calcul_ET0.r	calcul_ET0	calcul de l'évapotranspiration
Calcul_Pheno_Compromise.r	Calcul Pheno	calcul de la phénologie, appelle les fichiers PHENO_CURVE_ MODEL ou PHENO_ LINEAR_ MODEL selon la culture sélectionnée
Main_Compromise.r	MAIN	le script principal qui déclenche le calcul des indicateurs avec la fonction source()
param_Compromise_ble.r param_Compromise_mais.r param_Compromise_vigne.r	param	permet de définir les répertoires des données d'entrées et de sorties, les périodes, les corrections, le type de modèle phénologique à étudier et le nombre de sites.
pheno_curve_model_ 4stades_vigne.r pheno_curve_model_ 4stades_vigneSW.r	PHENO_CURVE_MODEL	calcul des stades phénologiques pour la vigne
pheno_linear_model_4stades.r pheno_linear_model_5stades.r pheno_linear_model_6stades.r	PHENO_LINEAR_MODEL	calcul des stades phénologiques, le modèle dépend de la culture sélectionnée
soil_water_balance.r	SOIL_WATER_BALANCE	déclenche le calcul du bilan hydrique pour chaque sol (dans notre cas, un sol seulement)
SWBressuyage.r	SWB	calcul du bilan hydrique (en fonction de la réserve en eau du sol)

Dans le répertoire Code, nous trouvons aussi un autre dossier fich_param qui contient les

paramètres (seuil de température qui affecte la plante s'il fait trop chaud/froid , pluviométrie en cas de forte pluie, etc.) pour chaque type de culture (précoce, tardive, chardonnay, syrah, etc.) et chaque indicateur climatique.

Dans les répertoires MBC_Bretagne, MBC_Provence, MBC_Ile_de_France, nous avons deux autres dossiers: MBC et SAFRAN dans lesquels il y a les variables climatiques stockées dans les fichiers .RData. Dans les dossiers MBC de chaque région, le fichiers sont nommés suivant un certain format. Pour la correction CDF-t et la région Bretagne nous avons, par exemple: BC_data_day_1dCDFt_1985_2014_2036_2065_Bretagne.RData et dans le dossier SAFRAN, nous avons Ref_data_day_SAFRAN_1985_2014_Bretagne.RData.

Le graphique suivant résume la classification des fichiers:

```
Dossier de travail
 ¦--Code
 ¦   ¦--calcul_ET0.r
 ¦   ¦--Calcul_Pheno_Compromise.r
 ¦   ¦--Main_Compromise.r
 ¦   ¦--param_Compromise_ble.r
 ¦   ¦--param_Compromise_mais.r
 ¦   ¦--param_Compromise_vigne.r
 ¦   ¦--pheno_curve_model_4stades_vigne.r
 ¦   ¦--pheno_curve_model_4stades_vigneSW.r
 ¦   ¦--pheno_linear_model_5stades.r
 ¦   ¦--pheno_linear_model_4stades.r
 ¦   ¦--pheno_linear_model_6stades.r
 ¦   ¦--soil_water_balance.r
 ¦   ¦--SWB.r
 ¦   ¦--SWBressuyage.R
 ¦   °--fich_param
 ¦--MBC_Bretagne
 ¦   °--Bretagne
 ¦       ¦--MBC
 ¦       ¦   ¦--BC_data_day_1dCDFt_1985_2014_2036_2065_Bretagne.RData
 ¦       ¦   ¦--BC_data_day_IntervardOTC_1985_2014_2036_2065_Bretagne.RData
 ¦       ¦   ¦--BC_data_day_IntervarR2D2_1985_2014_2036_2065_Bretagne.RData
 ¦       ¦   ¦--BC_data_day_SpatialdOTC_1985_2014_2036_2065_Bretagne.RData
 ¦       ¦   ¦--BC_data_day_SpatialR2D2_1985_2014_2036_2065_Bretagne.RData
 ¦       ¦   ¦--BC_data_day_SpatIntervardOTC_1985_2014_2036_2065_Bretagne.RData
 ¦       ¦   °--BC_data_day_SpatIntervarR2D2_1985_2014_2036_2065_Bretagne.RData
 ¦       °--SAFRAN
 ¦           °--Ref_data_day_SAFRAN_1985_2014_Bretagne.RData
 ¦--MBC_Ile_de_France
 ¦   °--Ile_de_France
 ¦       ¦--MBC
 ¦       °--SAFRAN
 °--MBC_Provence
     °--Provence
```

```
36          ¦--MBC
37          °--SAFRAN
```

Dans chacun des fichiers .RData, nous trouvons les 5 variables climatiques pour les 2 périodes (sauf SAFRAN où il n'y a que la période passée) et les latitudes et longitudes de la région. Les variables climatique sont déclarées telles que:

- température - tas
- humidité relative - hurs
- précipitations - pr
- vitesse du vent - sfcWind
- rayonnement - rsds

```
load("C:/Users/Ling/Desktop/Stage INRAE/Dossier de travail/MBC_Bretagne/Bret
     agne/MBC/BC_data_day_1dCDFt_1985_2014_2036_2065_Bretagne.RData")
ls()
```

```
##  [1] "hurs_day_1dCDFt_1985_2014_Bretagne"
##  [2] "hurs_day_1dCDFt_2036_2065_Bretagne"
##  [3] "IND_SAFRAN_Bretagne"
##  [4] "LAT_SAFRAN_Bretagne"
##  [5] "LON_SAFRAN_Bretagne"
##  [6] "mat_mailles_SAFRAN_Bretagne"
##  [7] "pr_day_1dCDFt_1985_2014_Bretagne"
##  [8] "pr_day_1dCDFt_2036_2065_Bretagne"
##  [9] "rsds_day_1dCDFt_1985_2014_Bretagne"
## [10] "rsds_day_1dCDFt_2036_2065_Bretagne"
## [11] "sfcWind_day_1dCDFt_1985_2014_Bretagne"
## [12] "sfcWind_day_1dCDFt_2036_2065_Bretagne"
## [13] "tas_day_1dCDFt_1985_2014_Bretagne"
## [14] "tas_day_1dCDFt_2036_2065_Bretagne"
```

Les variables climatiques sont stockées dans un format 3D dans un array. La dimension de l'array est: le nombre de sites de la région par le nombre de jours pour la période.Chaque période a une durée de 30 ans, donc pour chaque variable climatique nous avons 10950 valeurs (365 x 30). Dans les études climatologiques, le 29 février est négligé, et les études sont menées sous l'hypothèse qu'une année contient toujours 365 jours. Le stockage des données, est tel que, pour une variable climatique d'une correction/SAFRAN dans une période, en Bretagne, nous avons un array de dimension 21 x 20 x 10950.

```
dim(LAT_SAFRAN_Bretagne)
```

```
## [1] 21 20
```

```r
dim(LON_SAFRAN_Bretagne)
```

```
## [1] 21 20
```

```r
temp <- tas_day_1dCDFt_1985_2014_Bretagne
dim(temp)
```

```
## [1]    21    20 10950
```

```r
class(temp)
```

```
## [1] "array"
```

Le script principal est le script MAIN. Il charge les variables climatiques (température, précipitations, humidité, vent, rayonnement) avec la fonction load(), et ensuite lance les calculs des indicateurs agroclimatiques en appelant les scripts de calcul avec la fonction source(). L'appel aux autres scripts est fait dans un certain ordre, et dépend des conditions (qui sont vérifiés avec la fonction if()). Nous établissons un organigramme du script MAIN, ainsi qu'un diagramme de dépendances du code (voir annexe).

Une fois que nous avons analysé les dépendances des codes et compris leur fonctionnement, nous sommes aptes à formater les données.

2) Formatage des données

Le script de calcul MAIN, requiert que les données soient dans un format 2D (matrice), où les lignes représentent les sites et les colonnes les jours. De plus, nous sommes informés qu' il faut que les années bissextiles soient pris en compte, et que le script ne peut traiter des sites ayant des valeurs NaN.

Nous devons donc transformer les arrays en matrice et rajouter un jour aux années bissextiles. Nous décidons de dupliquer le 31 décembre, et donc les années bissextiles aurons un 32 décembre au lieu d'un 29 février (dans tous les cas, il y a bien 366 jours, cela ne pose pas problème car les jours sont en jours juliens).

Les années bissextiles pour la période passée (1985 à 2014) sont 1988, 1992, 1996, 2000, 2004, 2008 et 2012. Pour la période future (2036 à 2065) elles sont 2036, 2040, 2044, 2048, 2052, 2056, 2060, 2064.

Nous préparons un script de calcul, prepare_dataclimat.r, qui, en fonction de la région et période choisie, récupère les variables climatiques de chaque correction et SAFRAN depuis les arrays, les mets dans une matrice de dimensions nombre de sites (lignes) par 10950, enlève les sites qui ont des valeurs NaN (donc des lignes) et duplique les colonnes qui contiennent les valeurs du 31 décembre des années bissextiles. Chaque variable climatique est formatée à tour de rôle et est sauvegardée dans un nouveau répertoire, DATACOMPROMISE.

Le répertoire DATACOMPROMISE a été créé pour les données préparées (variables climatiques formatées). Ce nouveau répertoire contient des dossiers pour chaque correction, SAFRAN et chaque période. Par exemple, le dossier qui contient les variables climatiques de la Bretagne pour la correction univariée et la période passée s'appelle BC_data_day_1dCDFt_1985_2014_Bretagne et pour SAFRAN Ref_data_day_SAFRAN_1985_2014_Bretagne. Dans chacun de ces dossiers, nous avons 5 fichiers .RData (1 pour chaque variable climatique).

```
Dossier de travail
 °--MBC_Bretagne
     °--Bretagne
         ¦--DATACOMPROMISE
         ¦   ¦--BC_data_day_1dCDFt_1985_2014_Bretagne
         ¦   ¦   ¦--hurs_day_1dCDFt_1985_2014_Bretagne.RData
         ¦   ¦   ¦--pr_day_1dCDFt_1985_2014_Bretagne.RData
         ¦   ¦   ¦--rsds_day_1dCDFt_1985_2014_Bretagne.RData
         ¦   ¦   ¦--sfcWind_day_1dCDFt_1985_2014_Bretagne.RData
         ¦   ¦   °--tas_day_1dCDFt_1985_2014_Bretagne.RData
         ¦   ¦--BC_data_day_1dCDFt_2036_2065_Bretagne
         ¦   ¦--BC_data_day_IntervardOTC_1985_2014_Bretagne
         ¦   ¦--BC_data_day_IntervardOTC_2036_2065_Bretagne
         ¦   ¦--BC_data_day_IntervarR2D2_1985_2014_Bretagne
         ¦   ¦--BC_data_day_IntervarR2D2_2036_2065_Bretagne
         ¦   ¦--BC_data_day_SpatialdOTC_1985_2014_Bretagne
         ¦   ¦--BC_data_day_SpatialdOTC_2036_2065_Bretagne
         ¦   ¦--BC_data_day_SpatialR2D2_1985_2014_Bretagne
         ¦   ¦--BC_data_day_SpatialR2D2_2036_2065_Bretagne
         ¦   ¦--BC_data_day_SpatIntervardOTC_1985_2014_Bretagne
         ¦   ¦--BC_data_day_SpatIntervardOTC_2036_2065_Bretagne
         ¦   ¦--BC_data_day_SpatIntervarR2D2_1985_2014_Bretagne
         ¦   °--BC_data_day_SpatIntervarR2D2_2036_2065_Bretagne
         ¦--MBC
         °--SAFRAN
```

Dans les fichiers .RData des variables climatiques, il y a la matrice formatée (nombre de sites de la région par nombre de jours sur la période et le tout sans valeurs manquantes).

```
load("C:/Users/Ling/Desktop/Stage INRAE/Dossier de
    travail/MBC_Bretagne/Bretagne/DATACOMPROMISE/BC_data_day_1dCDFt_1985_20
    14_Bretagne/tas_day_1dCDFt_1985_2014_Bretagne.RData")
```

```
dim(DATA_CLIMAT_tmoy)
```

```
## [1]   259 10957
```

```r
sum(is.na(DATA_CLIMAT_tmoy))
```

```
## [1] 0
```

N.B: Nous avons utilisé le nom des fichiers initials pour nommer les répertoires créés. Le nom des répertoires et des fichiers contiennent le plus de détails possibles sur les données qu'ils contiennent. Ceci est important pour ne pas se tromper lors de la sélection des données d'entrées. Nous gardons le même format pour les fichiers de sorties.

Le code va récupérer les données d'entrées (données initiales) dans le répertoire concerné qui contient le nom de la région et la période. Et ensuite va sauvegarder les données de sorties (variables climatiques formatées) dans les répertoires qui ont les mêmes noms de région et période que les données d'entrées.

Dans les coordonnées que nous avons initialement, il y a autant de longitude et latitude qu'il y a de sites. Cependant, nous devons enlever les sites avec les NaN et leurs coordonnées. Il est important d'avoir un fichier avec les coordonnées des sites contenant des variables climatiques seulement, car les scripts de calcul de phénologie associe chaque ligne de la matrice à la ligne correspondante pour lire les latitudes, par exemple, la ligne 2 de la matrice a pour latitude la valeur à la ligne 2 dans le fichier contenant les coordonnées.

Nous préparons un script, prepare_coord_SAFRAN.r. Ce script construit la matrice de dimensions: nombre de sites par nombre de jours (le format exigé par le script MAIN). Il détecte ensuite le numéro de la ligne (site) qui contient des NaN. Et pour ce même numéro de ligne, il enlève les coordonnées. On sauvegarde ensuite ce nouveau fichier de coordonnées dans le répertoire MBC (coord_SAFRAN.txt).

3) Modification des scripts R

Une fois les variables climatiques et coordonnées des sites prêtes, nous mettons à jour les scripts de calculs.

Dans le script MAIN, nous ajoutons:

- Une condition if(), qui permet de déclarer le nombre de sites en fonction de région. Le nombre de sites est inférieur au nombre de sites qu'il y avait initialement, pour chaque région. Cela est lié au fait que les sites n'ayant pas de variables climatiques ont été enlevées de la matrice.

```r
## MAIN DEFINITION DU NOMBRE DE SITES
if (region == "Ile_de_France") {
    nb_sites = 319
}

if (region == "Bretagne") {
    nb_sites = 259
}
```

```
if (region == "Provence") {
    nb_sites = 337
}
```

- Les boucles pour les régions et les cultures.

```
## MAIN SELECTION DE LA REGION
regionliste = c("Provence")  ##'Ile_de_France','Provence','Bretagne'

## SELECTION DE LA CULTURE
cultureliste = c("mais")  ##'vigne','mais','ble'

for (region in regionliste) {
    print(region)

    for (culture in cultureliste) {
        print(culture)

        ## Code MAIN
    }

}
```

Dans les scripts param, nous:

- gardons seulement la variété précoce pour le blé et le maïs et chardonnay pour la vigne,

```
## param
varietyliste = c("precoce")  ##Pour mais et ble
varietyliste = c("chardonnay")  ##Pour vigne
```

- définissons les périodes à tour de rôle (dans un premier temps, nous mettons le passé et lançons les scripts, et ensuite, nous modifions manuellement et relançons pour le futur),

```
## param
periode_interet <- c("1985_2014")  #'2036_2065')
debut_periode <- c(1985)  #2036
fin_periode <- c(2014)  #2065
```

- définissons la liste de sites comme étant le nombre de sites total de la région,

```
## param
siteliste <- c(1:nb_sites)
```

- déclarons soil1 comme étant le type de sol pour tous les sites,

```
## param
soil1 <- c(1:nb_sites)

# soil <- list(soil1,soil2,soil3) #soil2 et soil3 ne seront
# pas utilisés pour notre étude
soil <- list(soil1)
```

- créons 3 répertoires pour les données de sorties (indicateurs agroclimatiques) avec la fonction dir.create().

```
## param
reppheno <- paste(reptravail, "Compromise", "/", region, "/",
    culture, "/", "PHENO", sep = "")
repsoilwat <- paste(reptravail, "Compromise", "/", region, "/",
    culture, "/", "BH", sep = "")
repevapo <- paste(reptravail, "Compromise", "/", region, "/",
    culture, "/", "ET0", sep = "")

dir.create(reppheno, showWarnings = FALSE, recursive = TRUE)
dir.create(repsoilwat, showWarnings = FALSE, recursive = TRUE)
dir.create(repevapo, showWarnings = FALSE, recursive = TRUE)
```

Dans les scripts calcul_ET0, Calcul_Pheno, SWB, nous faisons de sorte que les données de sorties aient les noms des corrections (ou SAFRAN) et les périodes. Nous utilisons la fonction save() et paste().

```
## calcul_ET0
save(DATA_CLIMAT_ET0, file = paste(repevapo, "/", "ET0_", correction,
    "_", per, ".RData", sep = ""))
```

```
## Calcul_Pheno
save(DATA_PHENO, file = paste(reppheno, "/pheno_", plant, "_",
    variety, "_", correction, "_", per, ".RData", sep = ""))
```

```
## SWB
save(WATER_RESERVE, file = paste(repsoilwat, "reserve_", plant,
    "_", variety, "_", correction, "_", periode, "_", colnames(PARAM_SWB)[k +
        2], ".RData", sep = ""))
```

Les données de sorties (indicateurs agroclimatiques calculés) sont enregistrées dans un nouveau répertoire Compromise (dans le répertoire Code). Avec les modifications dans le fichier param, lorsque le code est lancé pour une région et culture, 3 dossiers (ET0 pour l'évapotranspiration, BH pour le bilan hydrique, PHENO pour la phénologie) sont créés.

```
Dossier de travail
 °--Code
     °--Compromise
         ¦--Bretagne
         ¦   ¦--ble
         ¦   ¦   ¦--BH
         ¦   ¦   ¦   ¦--reserve_ble_precoce_1dCDFt_1985_2014_soil1.RData
         ¦   ¦   ¦   °--swc_ble_precoce_1dCDFt_1985_2014_soil1.RData
         ¦   ¦   ¦--ET0
         ¦   ¦   ¦   °--ET0_1dCDFt_1985_2014.RData
         ¦   ¦   °--PHENO
         ¦   ¦       ¦--pheno_ble_precoce_1dCDFt_1985_2014.RData
         ¦   ¦       °--pheno_SWB_ble_precoce_1dCDFt_1985_2014.RData
         ¦   ¦--mais
         ¦   °--vigne
         ¦--Ile_de_France
         °--Provence
```

4) Résultats des calculs: Les indicateurs agroclimatiques

Rappel : Les résultats obtenus sont sur la variété précoce pour le maïs et le blé, et chardonnay pour la vigne.

Une fois les répertoires de sorties indiqués dans les codes, nous lançons les scripts. Après plusieurs heures de calculs, nous observons que:

Le dossier PHENO comprend deux fichiers: pheno et pheno_SWB (pour chaque correction/SAFRAN et période) Les fichiers pheno contiennent les stades de phénologie à étudier.

```
load("C:/Users/Ling/Desktop/Stage INRAE/Dossier de
     travail/Code/Compromise/B
     retagne/ble/PHENO/pheno_ble_precoce_1dCDFt_1985_2014.RData")
```

```
head(DATA_PHENO)
```

```
##      site year  S0  S1  S2  S3  S4  S5
## [1,]    1 1986 274 277 427 484 496 542
## [2,]    2 1986 274 277 427 485 497 543
## [3,]    3 1986 274 277 420 482 494 542
## [4,]    4 1986 274 277 429 485 497 544
## [5,]    5 1986 274 277 429 485 497 544
## [6,]    6 1986 274 277 429 485 497 544
```

Les fichiers pheno_SWB contiennent les stades phénologie pour le calcul du bilan hydrique (ils ne seront pas utilisés pour la visualisation).

```
load("C:/Users/Ling/Desktop/Stage INRAE/Dossier de travail/Code/Compromise/Bretagne/ble/PHENO/pheno_SWB_ble_precoce_1dCDFt_1985_2014.RData")
```

```
head(DATA_PHENO_SWB)
```

```
##      site year  S0  S1  S2  S3  S4
## [1,]    1 1986 274 277 427 484 492
## [2,]    2 1986 274 277 427 485 493
## [3,]    3 1986 274 277 420 482 490
## [4,]    4 1986 274 277 429 485 493
## [5,]    5 1986 274 277 429 485 493
## [6,]    6 1986 274 277 429 485 493
```

La phénologie est un indicateur annuel - nous obtenons pour chaque site, une valeur (pour un stade) dans une année. La valeur correspond au numéro du jour où le stade commence.

Les stades phénologiques sont:

- S0: semis
- S1 : levée
- S2 : élongation
- S3 : floraison
- S4 : maturité
- S5: sénescence

Le dossier ET0 comprend un fichier seulement (pour chaque correction/SAFRAN et période). Les valeurs sont journalières. Nous obtenons 10957 valeurs pour chaque site dans le passé et 10958 valeurs pour chaque site dans le futur. Si nous prenons la Bretagne pour 1985 à 2014, nous avons une matrice de dimension 259 par 10957.

```
load("C:/Users/Ling/Desktop/Stage INRAE/Dossier de
     travail/Code/Compromise/Bretagne/ble/ET0/ET0_1dCDFt_1985_2014.RData")
```

```
dim(DATA_CLIMAT_ET0)
```

```
## [1]   259 10957
```

```
DATA_CLIMAT_ET0[1:5, 1:8]
```

```
##           X1       X2       X3       X4       X5       X6       X7       X8
## [1,] 7.931553 8.946461 9.803749 3.328171 4.781556 6.320897 6.084255 2.432992
## [2,] 8.182024 8.868187 9.752565 3.333603 4.748465 5.962272 5.948964 2.379767
## [3,] 5.341594 8.068528 7.235926 2.612978 3.776577 5.203236 4.707925 1.861480
## [4,] 7.078180 9.391417 8.970063 3.237019 4.639794 6.658122 6.066942 2.317385
## [5,] 7.514172 9.305714 9.144701 3.234736 4.715639 6.472579 6.000691 2.284213
```

Le dossier BH contient deux indicateurs: reserve et swc. Les fichiers reserve correspondent à la réserve en eau du sol (en cm) et les fichiers swc correspondent à la teneur en eau du sol (en pourcentage massique). Comme l'évapotranspiration, les indicateurs du bilan hydrique sont journaliers.

```
load("C:/Users/Ling/Desktop/Stage INRAE/Dossier de travail/Code/Compromise/Bretagne/ble/BH/reserve_ble_precoce_1dCDFt_1985_2014_soil1.RData")
```

```
dim(WATER_RESERVE)
```

```
## [1]   259 10957
```

```
WATER_RESERVE[1:5, 1:8]
```

```
##          [,1]     [,2]     [,3]     [,4]     [,5]     [,6]     [,7]     [,8]
## [1,] 217.4479 211.1854 204.5085 203.6030 200.2559 195.8312 191.5723 196.0404
## [2,] 217.2726 211.0649 204.4775 203.6832 200.3592 196.1857 192.0214 196.9473
## [3,] 219.4986 213.8506 209.0152 208.6132 205.9696 202.3273 199.0318 203.6465
## [4,] 218.2350 211.6611 205.5926 204.6179 201.3700 196.7094 192.4625 196.9385
## [5,] 217.9257 211.4117 205.2334 204.3038 201.0028 196.4720 192.2715 196.9227
```

```
load("C:/Users/Ling/Desktop/Stage INRAE/Dossier de travail/Code/Compromise/Bretagne/ble/BH/swc_ble_precoce_1dCDFt_1985_2014_soil1.RData")
```

```
dim(SOIL_WATER_CONTENT)
```

```
## [1]   259 10957
```

```
SOIL_WATER_CONTENT[1:5, 1:8]
```

```
##          [,1]     [,2]     [,3]     [,4]     [,5]     [,6]     [,7]     [,8]
## [1,] 27.33199 26.88467 26.40775 26.34307 26.10399 25.78795 25.48373 25.80289
## [2,] 27.31947 26.87606 26.40554 26.34880 26.11137 25.81326 25.51581 25.86766
## [3,] 27.47847 27.07505 26.72966 26.70094 26.51211 26.25195 26.01656 26.34618
## [4,] 27.38822 26.91865 26.48518 26.41556 26.18357 25.85067 25.54732 25.86704
## [5,] 27.36612 26.90084 26.45953 26.39313 26.15734 25.83372 25.53368 25.86590
```

B. Création de fonctions

Pour visualiser l'évapotranspiration, le bilan hydrique et les stades phénologiques nous avons créé des fonctions.

N.B. Créer des codes génériques est un travail long, car il nécessite aussi de débugger les codes. Nous avons durant notre stage, fait beaucoup de débogage, que ce soit pour préparer les données avant de calculer les indicateurs ou pour créer des fonction permettant de visualiser ces indicateurs.

Pour faciliter le débogage des fonctions créées, nous avons fait plusieurs fonctions interreliées, au lieu d'une seule longue fonction. Les scripts de fonctions (appelés fichiers source) dans un premier temps: appellent les données, les filtrent, les sauvent dans l'environnement. Ensuite, la fonction finale est exécutée.

Pour chacune des fonction créée, il y a un script où il y a la fonction prête à utiliser, et il y a un fichier source où la fonction est créée.

Remarque: Pour pouvoir manipuler les indicateurs climatiques journaliers (évapotranspiration, réserve d'eau et teneur en eau du sol), il a fallu créer un script qui vectorise les dates de la période, incluant les 32 décembres au lieu des 29 février pour les années bissextiles. Pour la période passée, c'est le script fonction_dated_colnames_1985_2014.r et la période future, le script fonction_dated_colnames_2036_2065.r

1) Boxplot - Indicateur: Phénologie

Fonction: boxplot.indicateur
Script: Boxplot_phenologie.r
Fichier source: fonction_boxplot_phenologie.r

Nous souhaitons afficher des boxplots pour la phénologie. Nous créons une fonction qui permet de choisir: la région, la culture, la période, le stade phénologique, l'année de début, l'année de fin et les corrections.

Selon des arguments de la fonction, le code récupère les données dans les répertoires concernés et affiche les boxplots.

La fonction:
Charge les données de phénologie (fichiers pheno) pour toutes les corrections;
Récupères les lignes dont l'année égale celle de l'argument et la colonne dont le stade correspond à celui de l'argument;
Affiche le boxplot des valeurs de ce stade.

```
boxplot.indicateur(region, culture, periode, indicateur, annee_debut,
    annee_fin, selection_correction)
```

region	le nom de la region dans ce format: “Bretagne”, “Ile_de_France”, “Provence”.
culture	la culture, choix entre “mais”,”ble”,”vigne”
periode	passé ou futur en choisissant entre “1985_2014” ou “2036_2065”.
indicateur	le stade de phénologie qui souhaite être représenté, un stade entre: “S0” pour le semis ; “S1” pour la levée; “S2” pour l’élongation; “S3” pour la floraison; “S4” pour la maturité; “S5” pour la sénescence.
annee_debut	l’année à laquelle on commence l’affichage des boxplots. Doit être comprise dans la limite de la période. Si la culture est blé ou vigne, peut commencer à partir de 1986 pour la passé et 2037 pour le futur seulement.
annee_fin	l’année jusqu’à laquelle on affiche les boxplots. Doit être comprise dans la limite de la période.
selection_correction	un vecteur avec le noms des corrections. Par défaut contient toutes les corrections, c("1dCDFt","IntervardOTC","IntervarR2D2", "SpatialdOTC", "SpatialR2D2", "SpatIntervardOTC", "SpatIntervarR2D2").

2) Boxplot - Indicateur: Bilan hydrique, Evapotranspiration

Fonction: boxplot.indicateur
Script: Boxplot_ET0_BH.r
Fichier source: fonction_boxplot_BH_ET0.r

Cette fonction créée permet d’afficher les boxplots d’une statistique (moyenne, variance, mean, max, etc.) d’un indicateur (évapotranspiration, réserve en eau, teneur en eau du sol) pour un mois, sur une période choisie.

```
boxplot.indicateur(x, region, culture, periode, mois, annee_debut,
    annee_fin, statistique, selection_correction)
```

x	l'indicateur agroclimatique qui doit être représenté. Pour l'évapotranspiration "ET0", pour la teneur en eau du sol "SWC", pour la réserve en eau du sol, "WATER_RESERVE"
region, culture, periode, annee_debut, annee_fin, selection_correction	identiques à la fonction boxplot.indicateur pour la phénologie
mois	le mois pour lequel on calcule l'indicateur, un entier entre 1 et 12.
statistique	la fonction R de ce qui doit être calculé: mean, median, max, min, sd

Par exemple: Pour le blé en Bretagne, nous souhaitons avoir la moyenne d'évapotranspiration au mois de juillet de 1996 à 2000.

La fonction est programmé de la façon suivante:
Elle charge les données selon l'indicateur choisi;
Elle sélectionne les valeurs qui vont du 1er juillet au 31 juillet 1996, 1er juillet au 31 juillet 1997, 1er juillet au 31 juillet 1998, etc.;
Elle les met dans un vecteur;
Elle calcule la statistique sur ce vecteur;
Elle répète cette étape pour toutes les corrections;
Elle affiche la statistique de l'indicateur seulement pour les corrections choisies.

Chaque boxplot résume un nombre d'observations qui est égal au nombre de site.

3) Cartographie - Indicateur: Bilan hydrique, Evapotranspiration

Fonction: cartographie.indicateur
Script: Cartographie_ET0_BH.r
Fichier source: fonction_cartographie_ET0_BH.r

Pour la culture d'une région, cette fonction calcule la statistique sur les valeurs de l'évapotranspiration ou du bilan hydrique d'un mois pendant 30 ans (29 ans pour le bilan hydrique de la vigne).

```
cartographie.indicateur(x, region, culture, periode, mois, statistique)
```

x	l'indicateur agroclimatique qui doit être représenté. Pour l'évapotranspiration "ET0", pour la teneur en eau du sol "SWC", pour la réserve en eau du sol, "WATER_RESERVE".
region, culture, periode, mois, statistique	identiques à la fonction boxplot.indcateur pour l'évapotranspiration et le bilan hydrique.

Exemple: L'évapotranspiration en Provence pour le maïs au mois de juillet de 2036 à 2065.

La fonction exécute les étapes suivantes:
Charge les données selon l'indicateur choisi, pour chaque correction;
Sélectionne les valeurs du mois sur toute la période 1er juillet au 31 juillet 2036, 1er juillet au 31 juillet 2037, 1er juillet au 31 juillet 2037, etc.;
Elle les mets dans un vecteur;
Elle calcule la statistique sur ce vecteur.

Elle affiche la cartographie de la région avec la statistique de l'indicateur.

Si la période passée est indiquée, nous retrouvons 8 cartographies (SAFRAN + 7 corrections), si la période future est indiquée, nous retrouvons 7 cartographies (les 7 corrections seulement).

4) Cartographie - Indicateur: Phénologie

Fonction: cartographie.indicateur
Script: Cartographie_phenologie.r
Fichier source: fonction_cartographie_phenologie.r

Cette fonction exécute les étapes suivantes:
Charge les données de phénologie pour chaque correction;
Sélectionne la colonne qui correspond au stade de l'argument;
Pour chaque site, calcule la statistique sur la période.

Elle affiche suite la cartographie de la région avec la statistique du stade.

```
cartographie.indicateur(region, culture, periode, indicateur,
    statistique)
```

region,culture, periode, indicateur	identiques à la fonction boxplot.indicateur de la phénologie
statistique	la fonction R de ce qui doit être calculé, mean, median, max, min, sd

C. Visualisation des indicateurs agroclimatiques

N.B. Cette partie contient une liste non-exhaustive des graphiques des indicateurs agroclimatiques.

L'indicateur swc (réserve en eau du sol) du bilan hydrique ne montre aucune différence dans les graphiques. Il ne sera pas présenté dans ce rapport.

J'ai choisi de présenter la phénologie, l'évapotranspiration, la réserve en eau du sol. Cela pour le maïs en Bretagne, le blé en Ile-de-France et la vigne en Provence (les régions propices à ces cultures).

1) Cartographies

```
source("fonction_cartographie_ET0_BH.R")
cartographie.indicateur("ET0", "Bretagne", "mais", "1985_2014",
    7, sd)
```

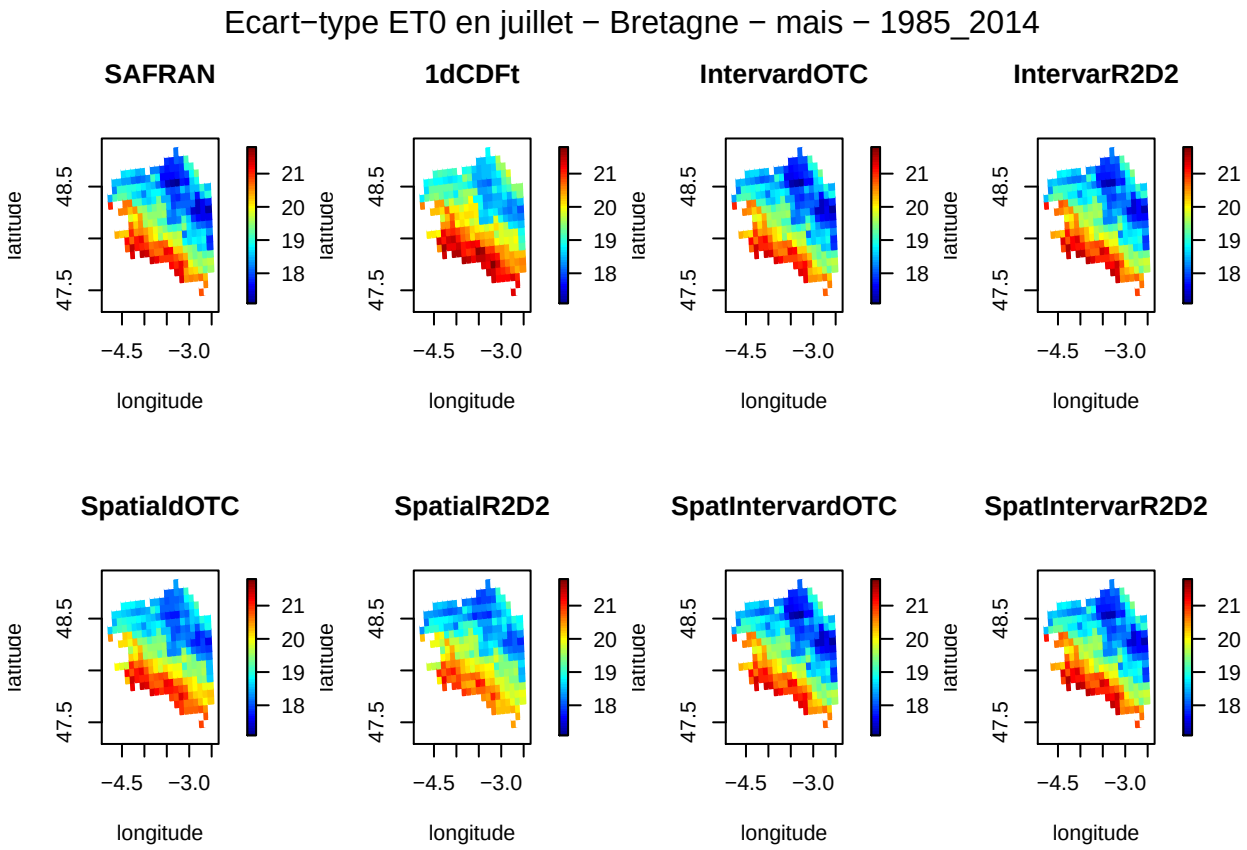


Figure 1.1

```
source("fonction_cartographie_ET0_BH.R")
cartographie.indicateur("ET0", "Bretagne", "mais", "2036_2065",
    7, sd)
```

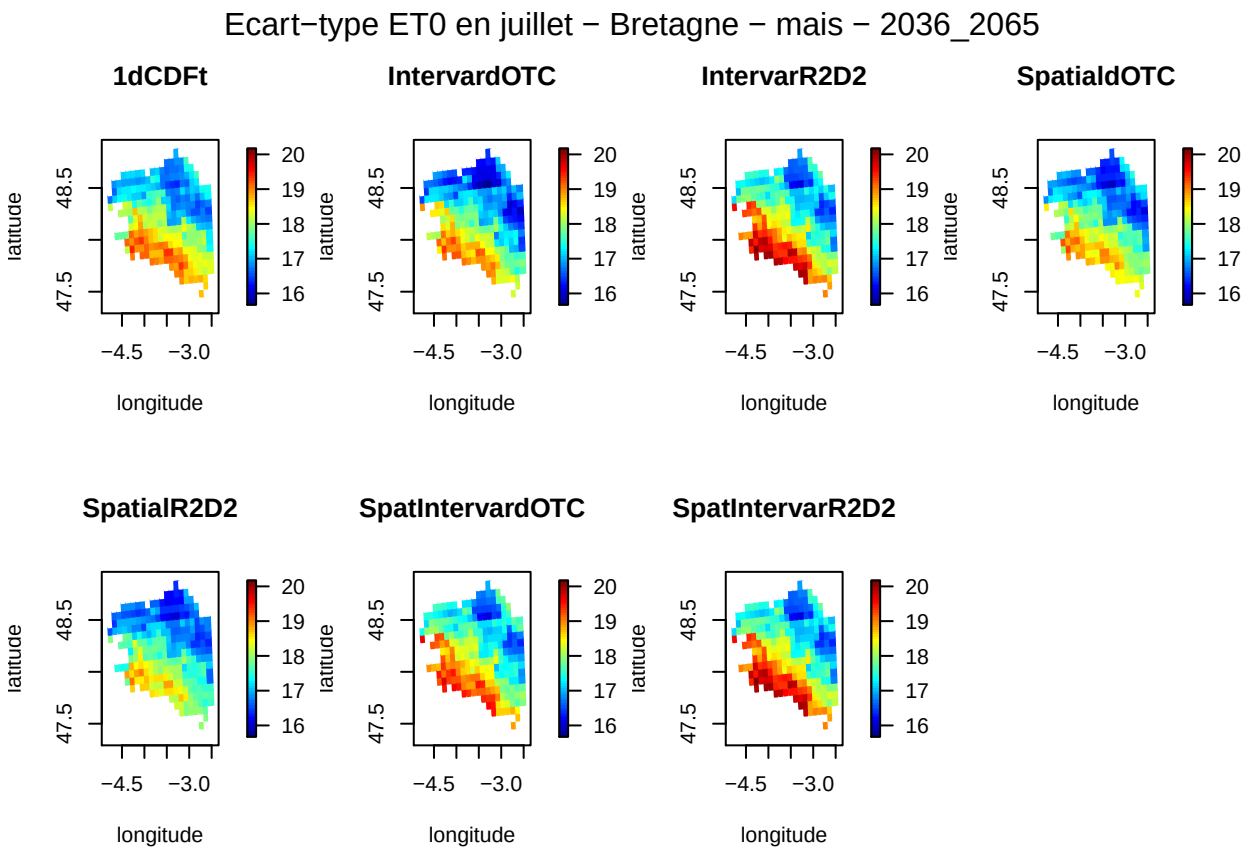


Figure 1.2

```
source("fonction_cartographie_ET0_BH.R")
cartographie.indicateur("ET0", "Ile_de_France", "ble", "1985_2014",
    7, sd)
```

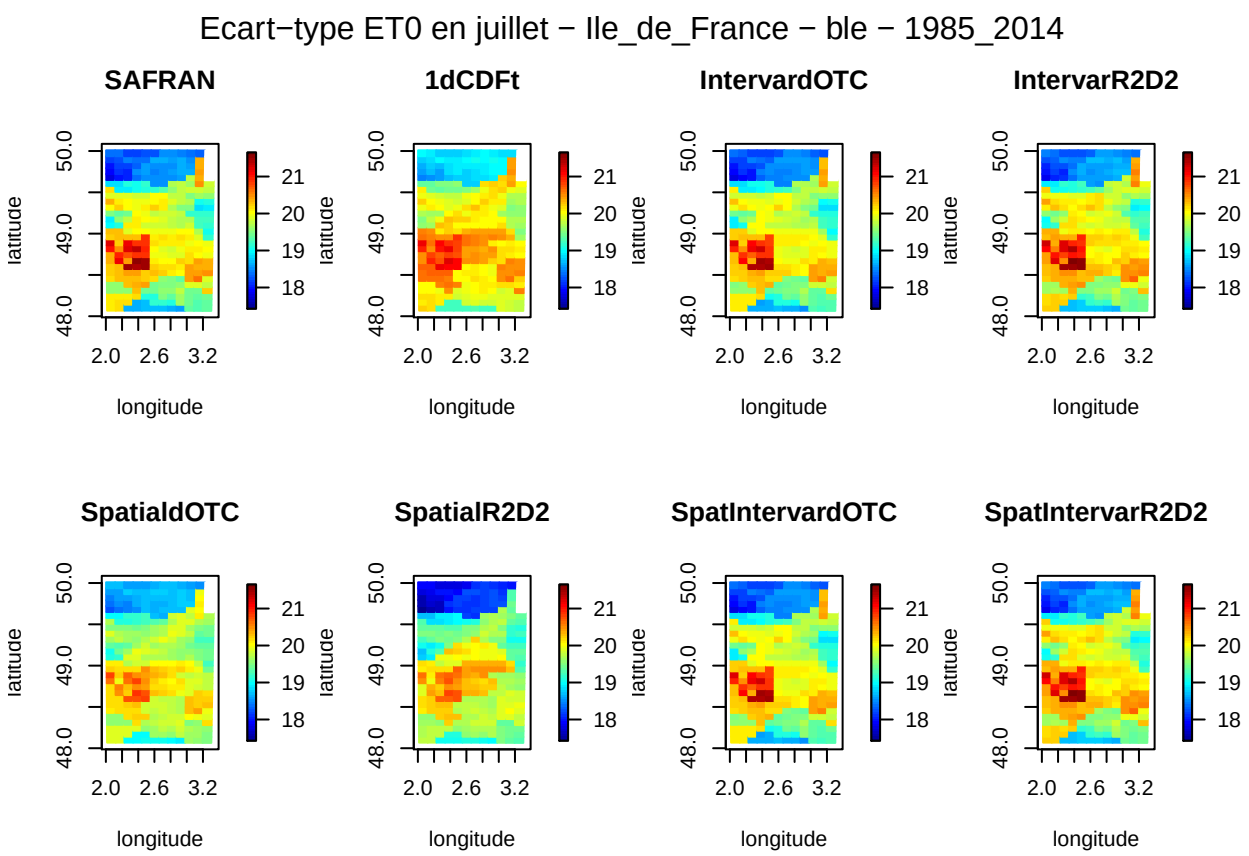


Figure 1.3

```
source("fonction_cartographie_ET0_BH.R")
cartographie.indicateur("ET0", "Ile_de_France", "ble", "2036_2065",
    7, sd)
```

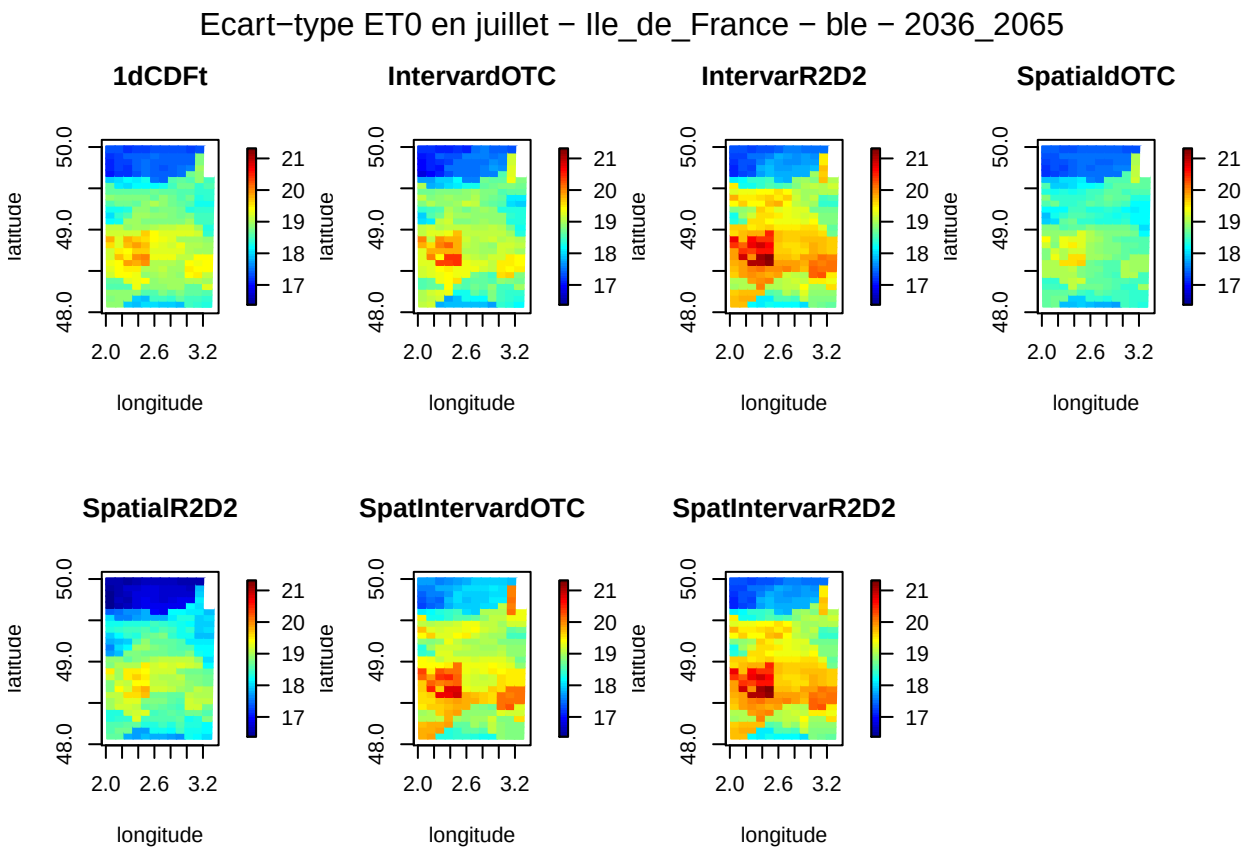


Figure 1.4

```
source("fonction_cartographie_ET0_BH.R")
cartographie.indicateur("ET0", "Provence", "vigne", "1985_2014",
    7, sd)
```

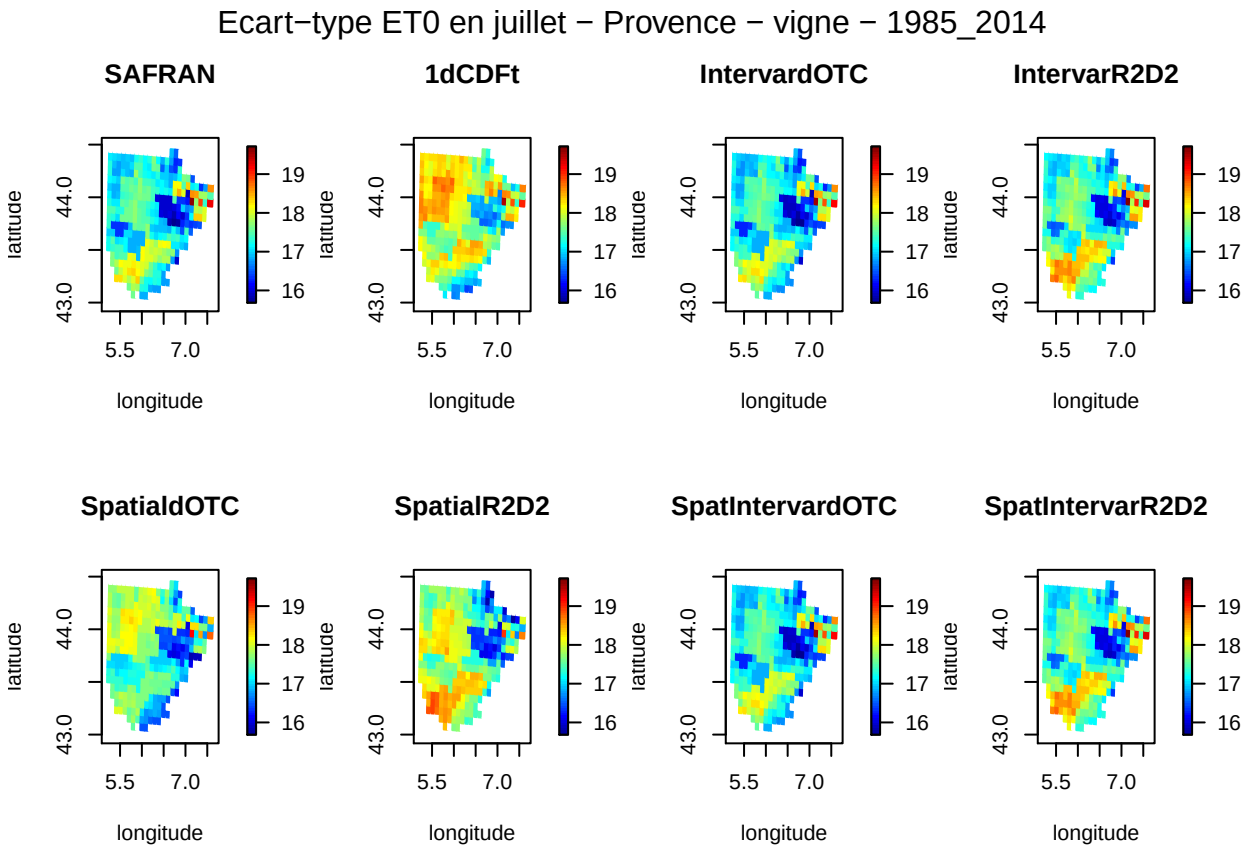


Figure 1.5

```
source("fonction_cartographie_ET0_BH.R")
cartographie.indicateur("ET0", "Provence", "vigne", "2036_2065",
    7, sd)
```

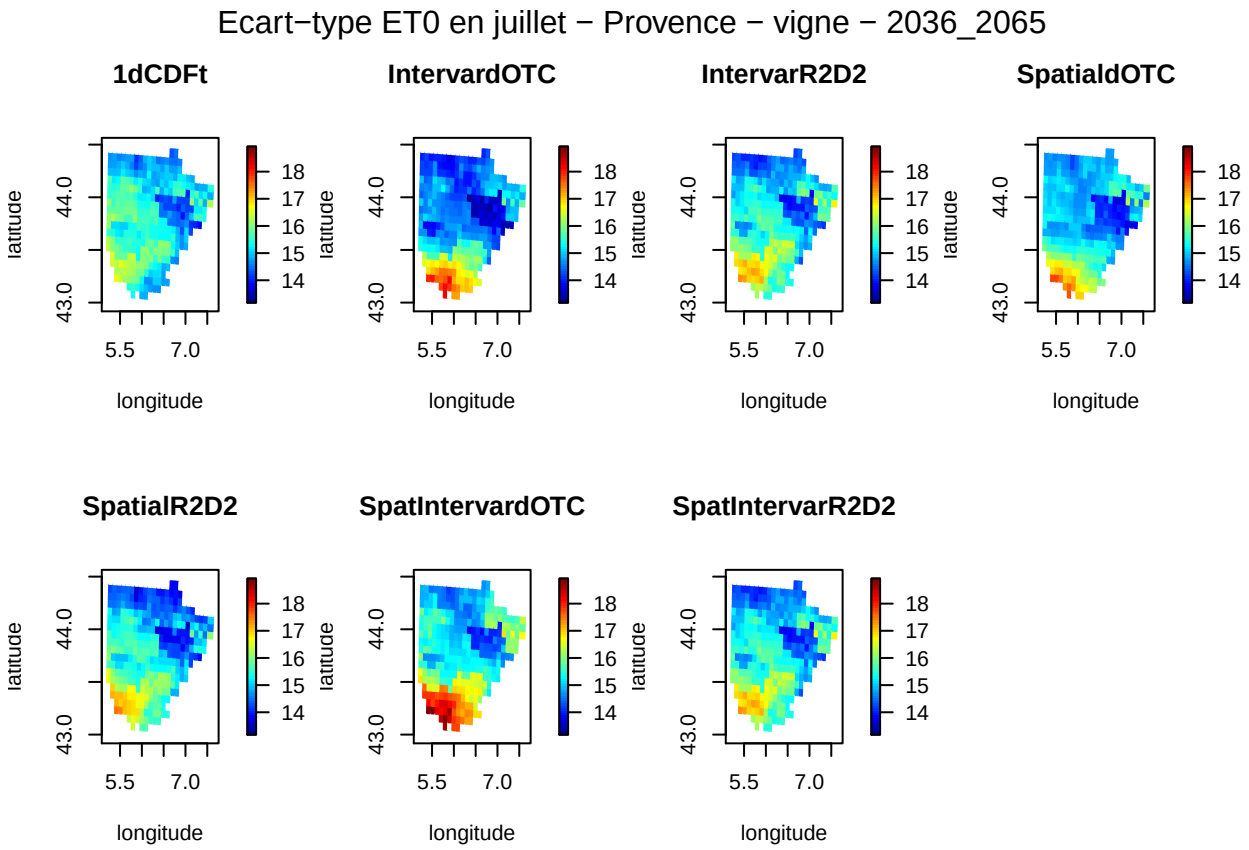


Figure 1.6

```
source("fonction_cartographie_ET0_BH.R")
cartographie.indicateur("WATER_RESERVE", "Bretagne", "mais",
    "1985_2014", 7, mean)
```

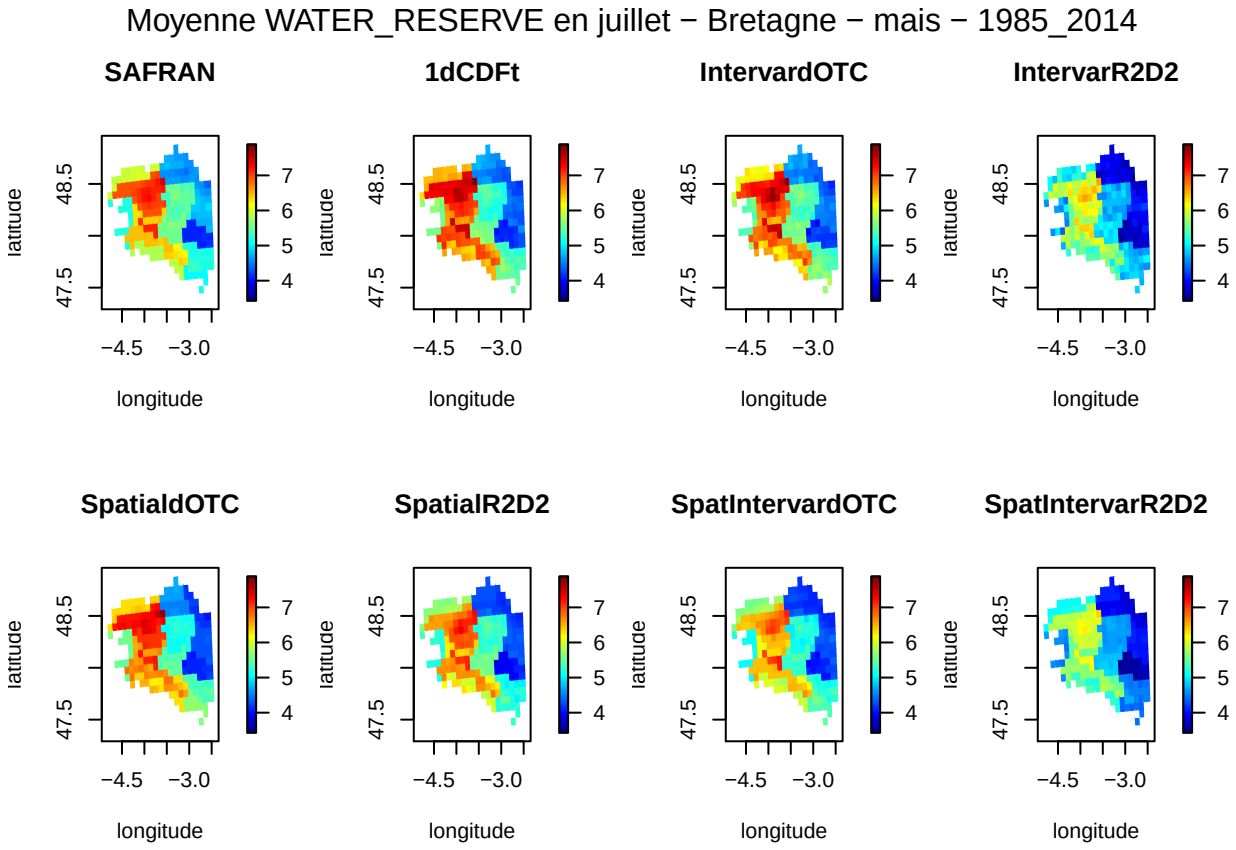


Figure 1.7

```
source("fonction_cartographie_ET0_BH.R")
cartographie.indicateur("WATER_RESERVE", "Bretagne", "mais",
    "2036_2065", 7, mean)
```

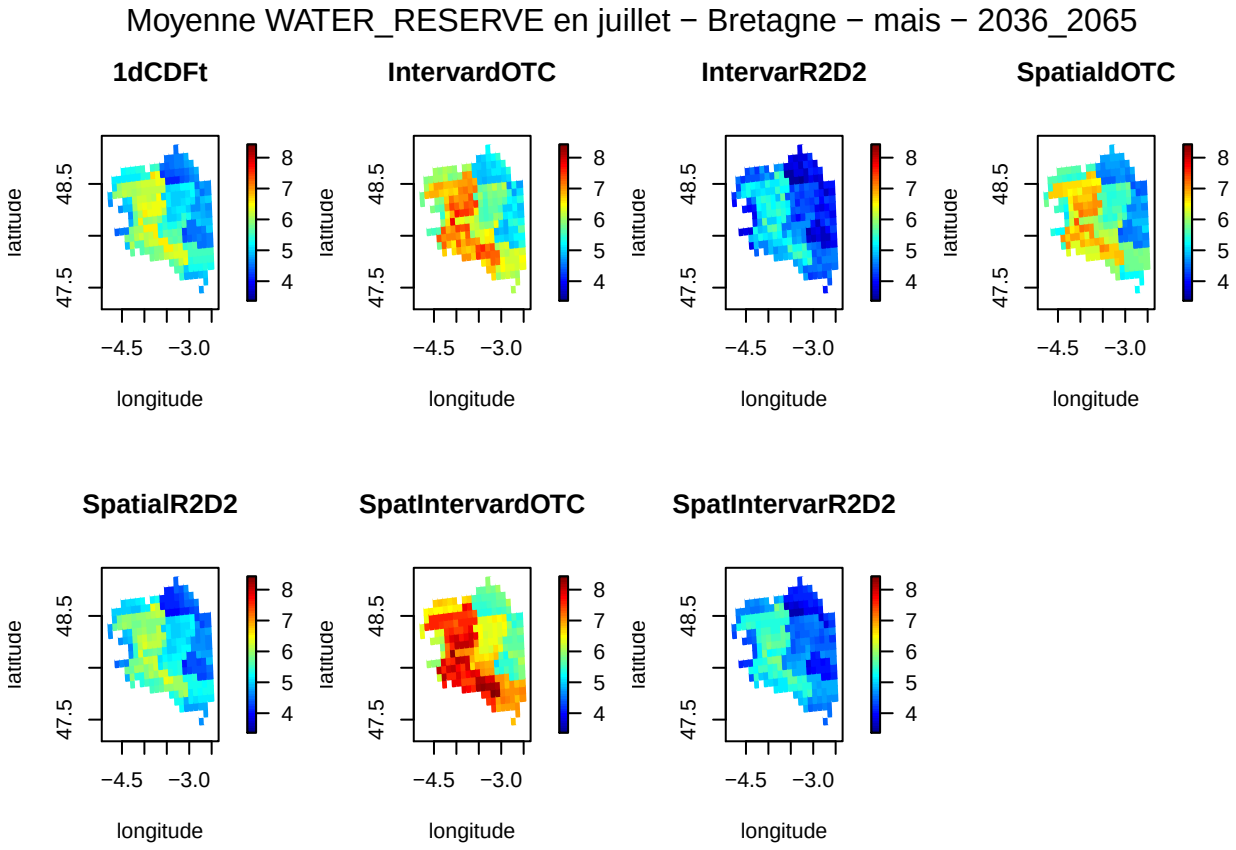


Figure 1.8

```
source("fonction_cartographie_ET0_BH.R")
cartographie.indicateur("WATER_RESERVE", "Ile_de_France", "ble",
    "1985_2014", 7, mean)
```

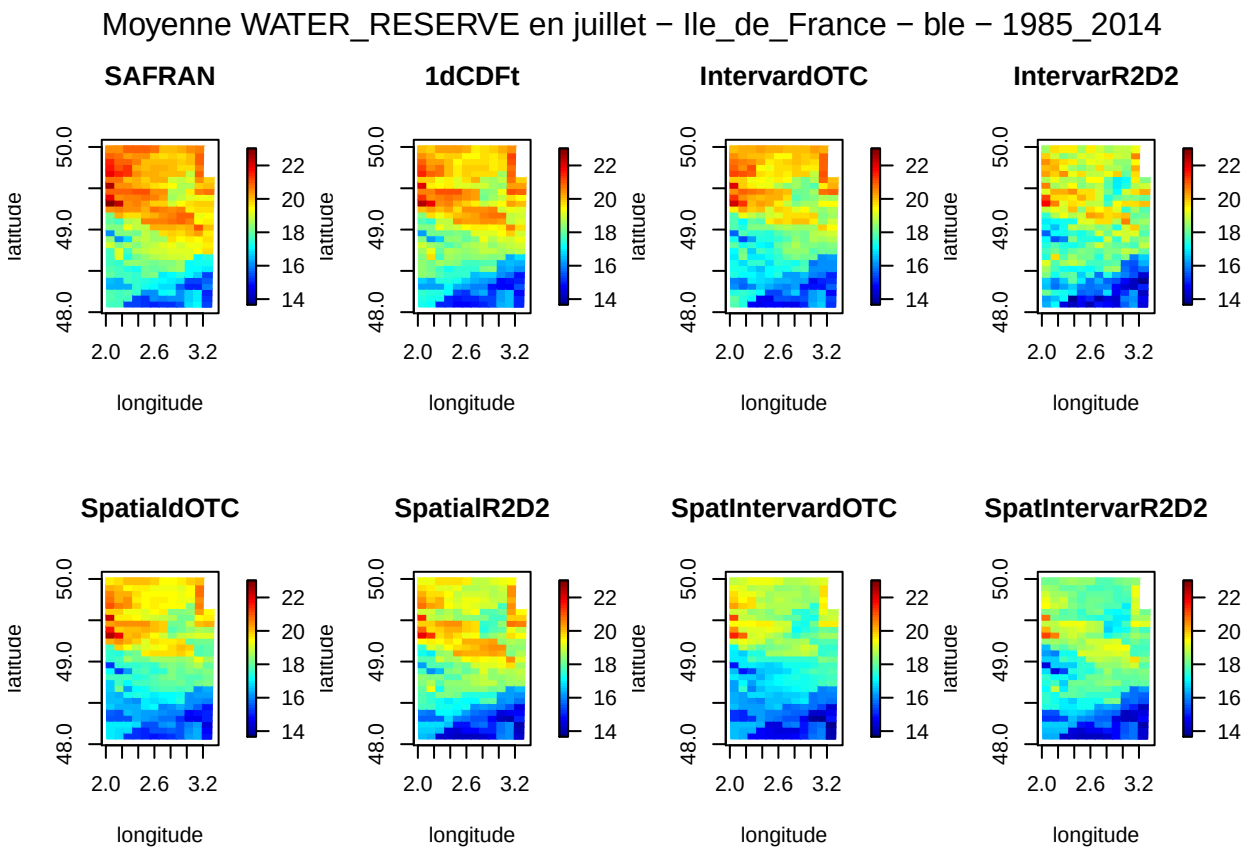


Figure 1.9

```
source("fonction_cartographie_ET0_BH.R")
cartographie.indicateur("WATER_RESERVE", "Ile_de_France", "ble",
    "2036_2065", 7, mean)
```

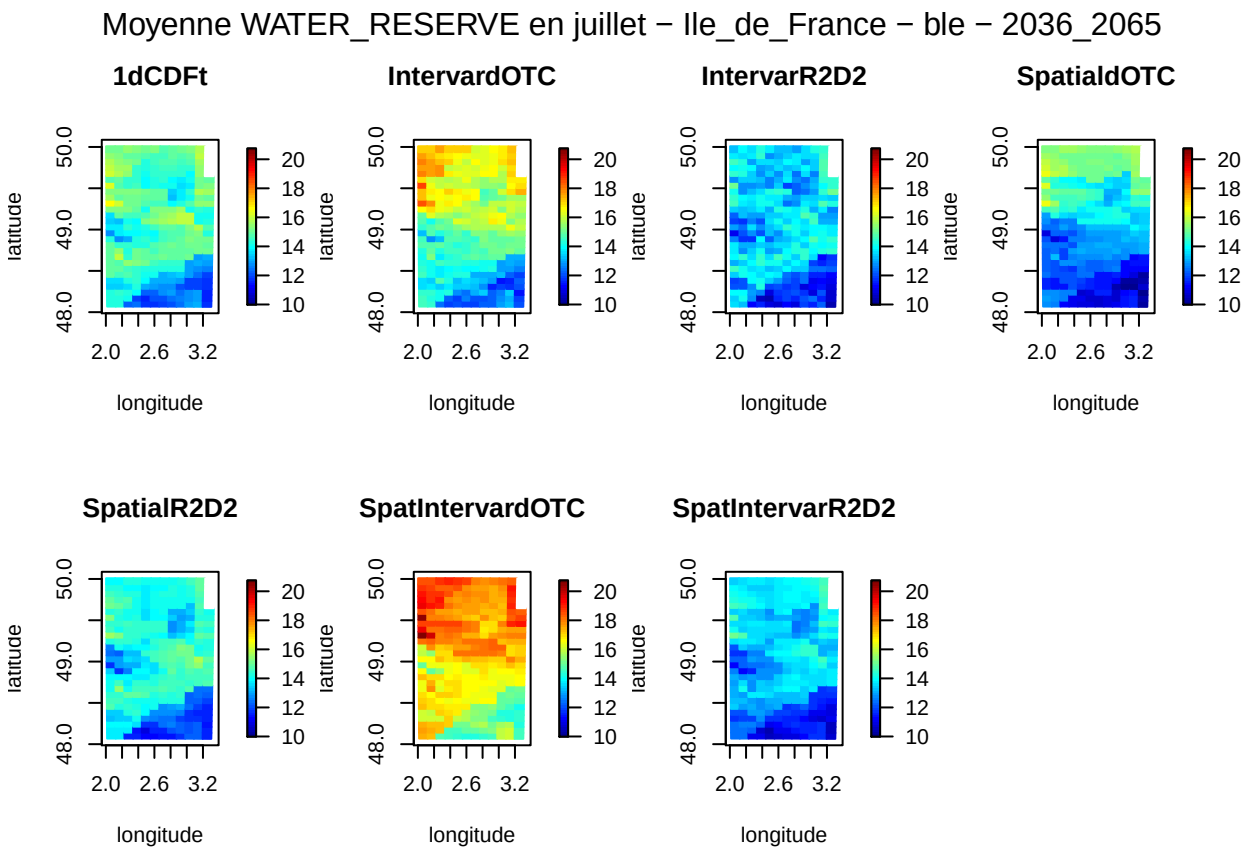


Figure 1.10

```
source("fonction_cartographie_ET0_BH.R")
cartographie.indicateur("WATER_RESERVE", "Provence", "vigne",
    "1985_2014", 7, mean)
```

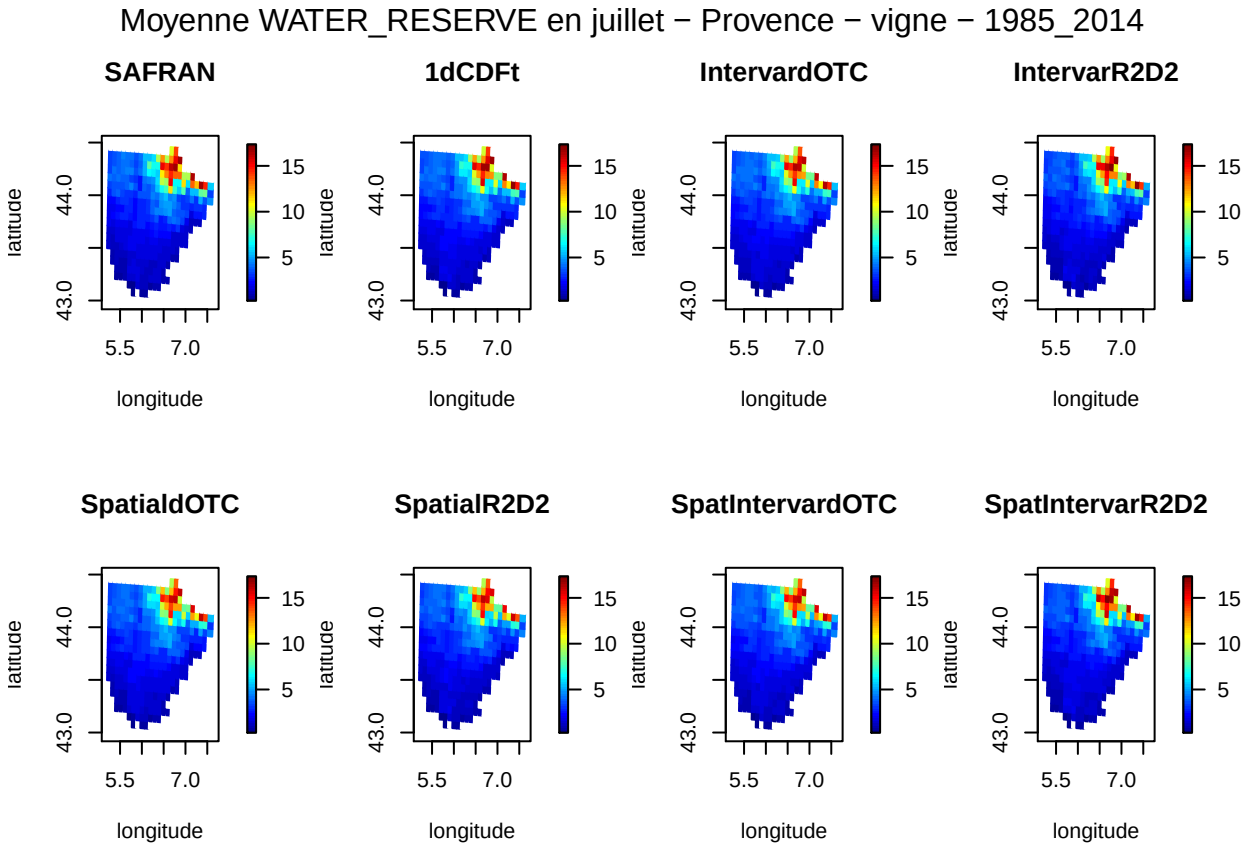


Figure 1.11

```
source("fonction_cartographie_ET0_BH.R")
cartographie.indicateur("WATER_RESERVE", "Provence", "vigne",
    "2036_2065", 7, mean)
```

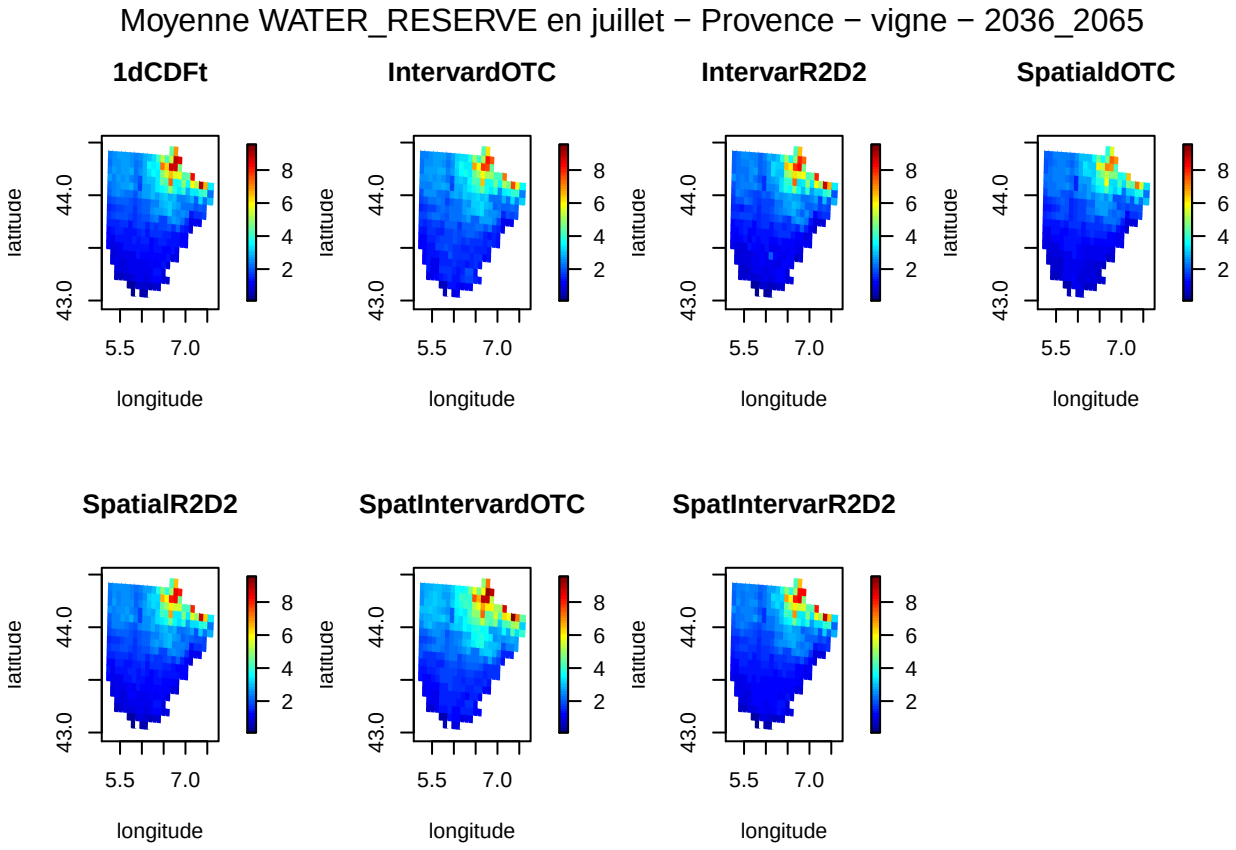


Figure 1.12

```
source("fonction_cartographie_phenologie.R")
cartographie.indicateur("Bretagne", "mais", "1985_2014", "S3",
    max)
```

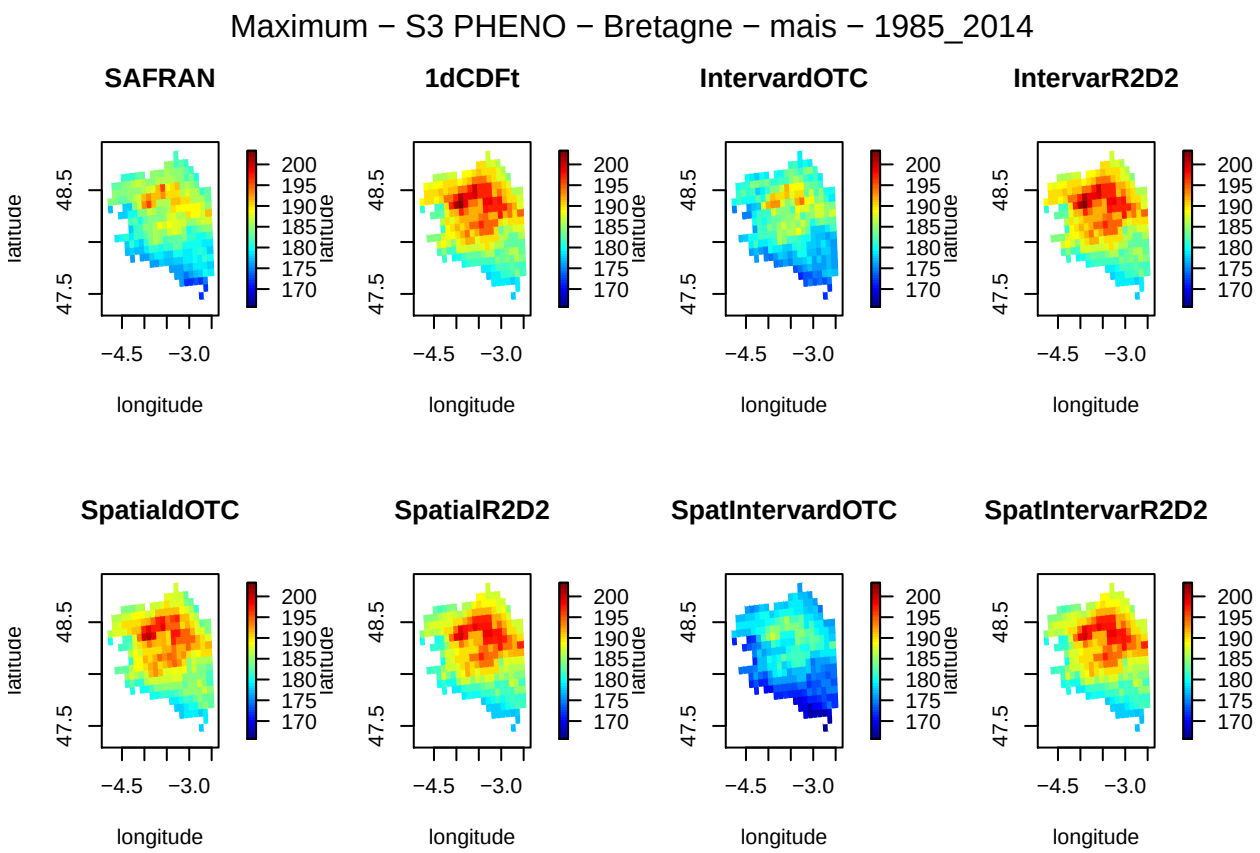


Figure 1.13

```
source("fonction_cartographie_phenologie.R")
cartographie.indicateur("Bretagne", "mais", "2036_2065", "S3",
    max)
```

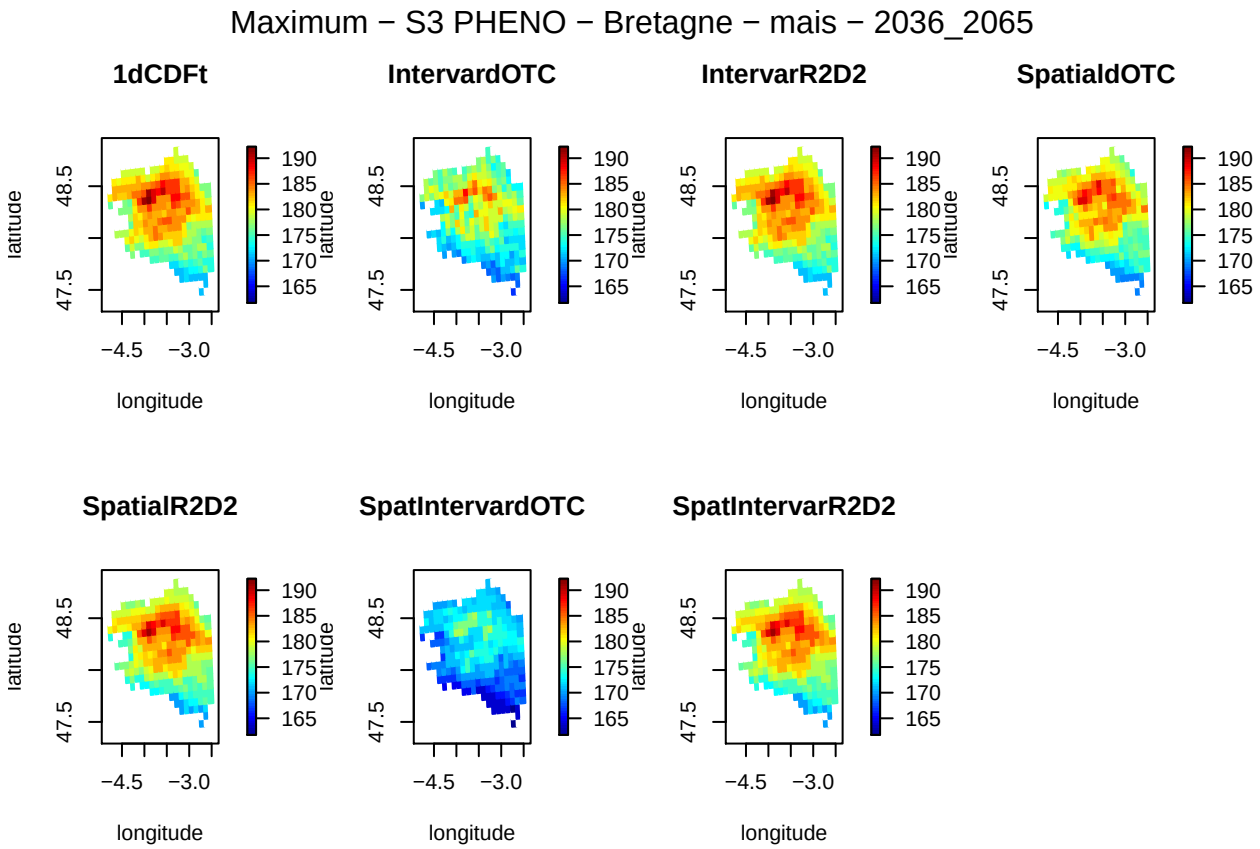


Figure 1.14

```
source("fonction_cartographie_phenologie.R")
cartographie.indicateur("Ile_de_France", "ble", "1985_2014",
    "S3", max)
```

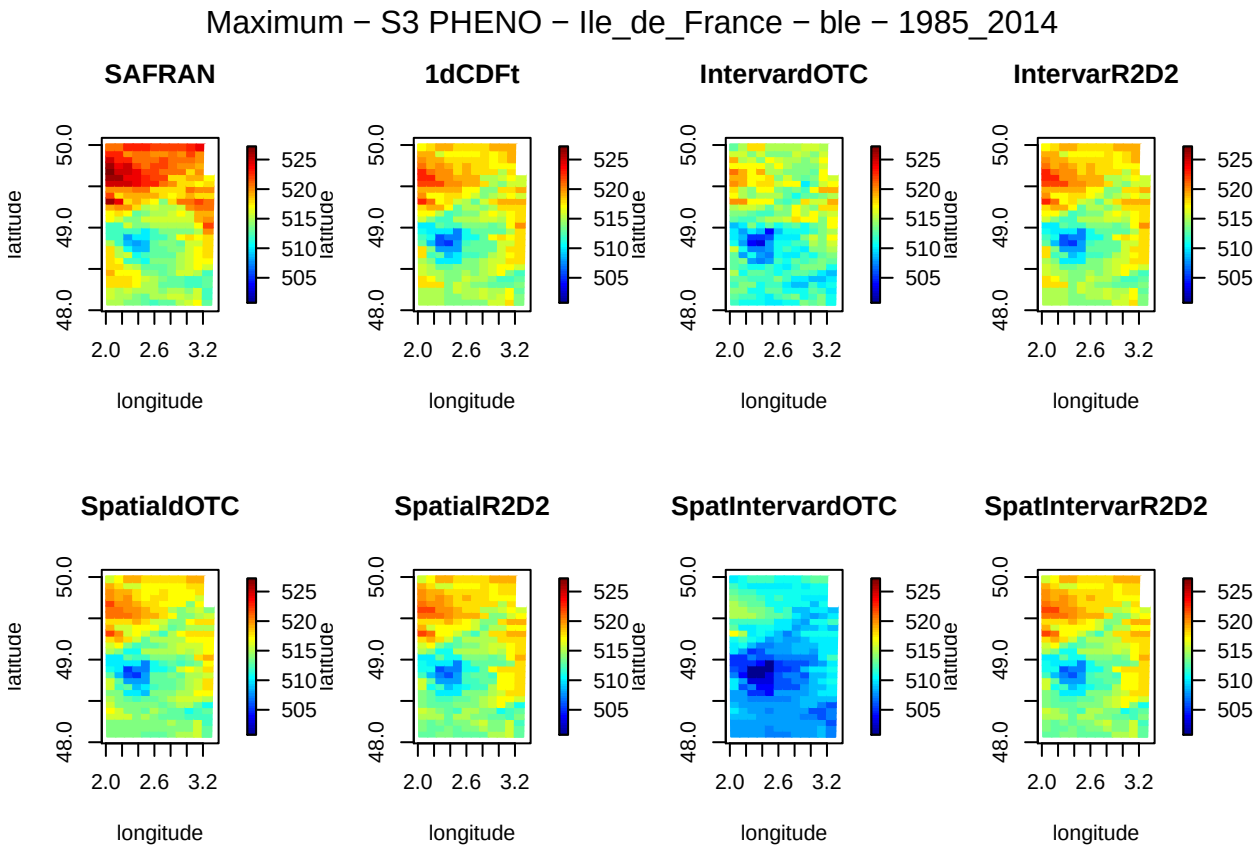


Figure 1.15

```
source("fonction_cartographie_phenologie.R")
cartographie.indicateur("Ile_de_France", "ble", "2036_2065",
    "S3", max)
```

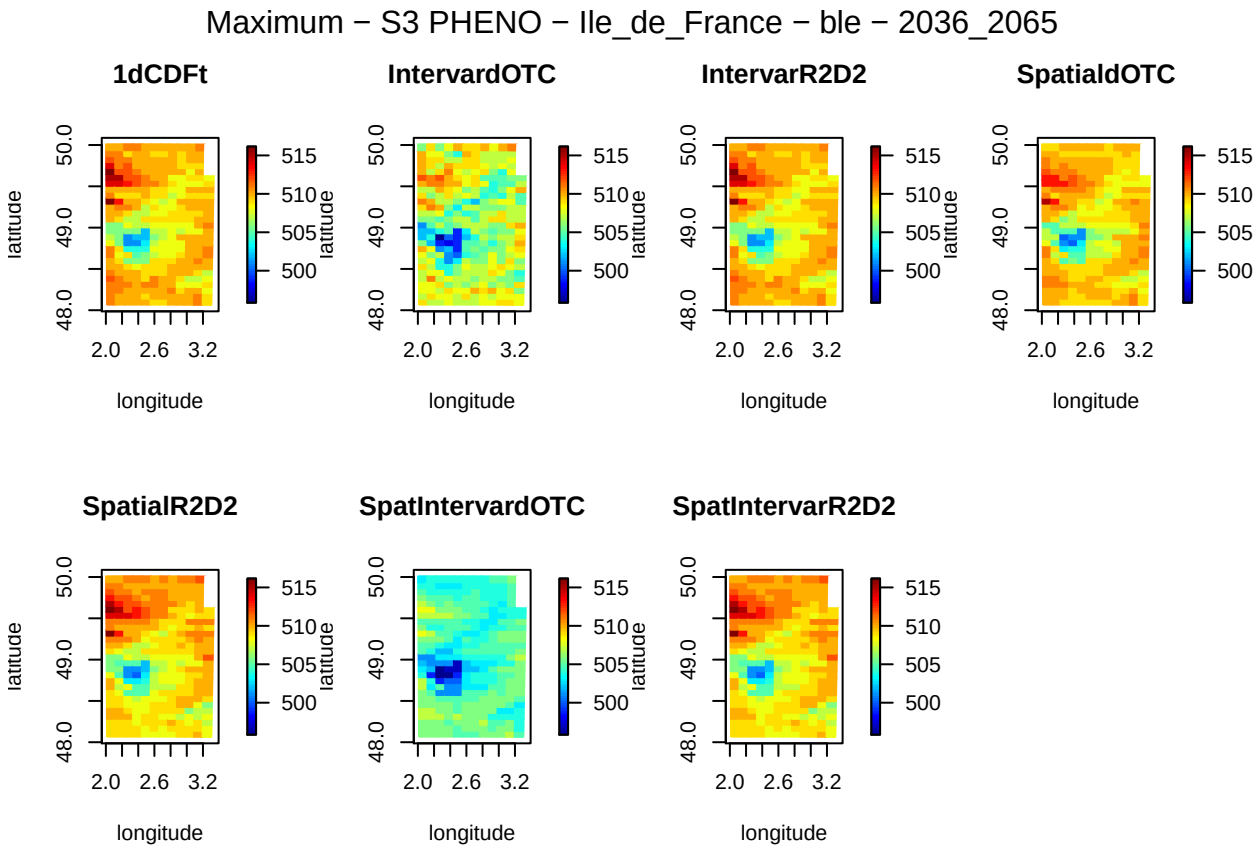


Figure 1.16

```
source("fonction_cartographie_phenologie.R")
cartographie.indicateur("Provence", "vigne", "1985_2014", "S3",
    max)
```

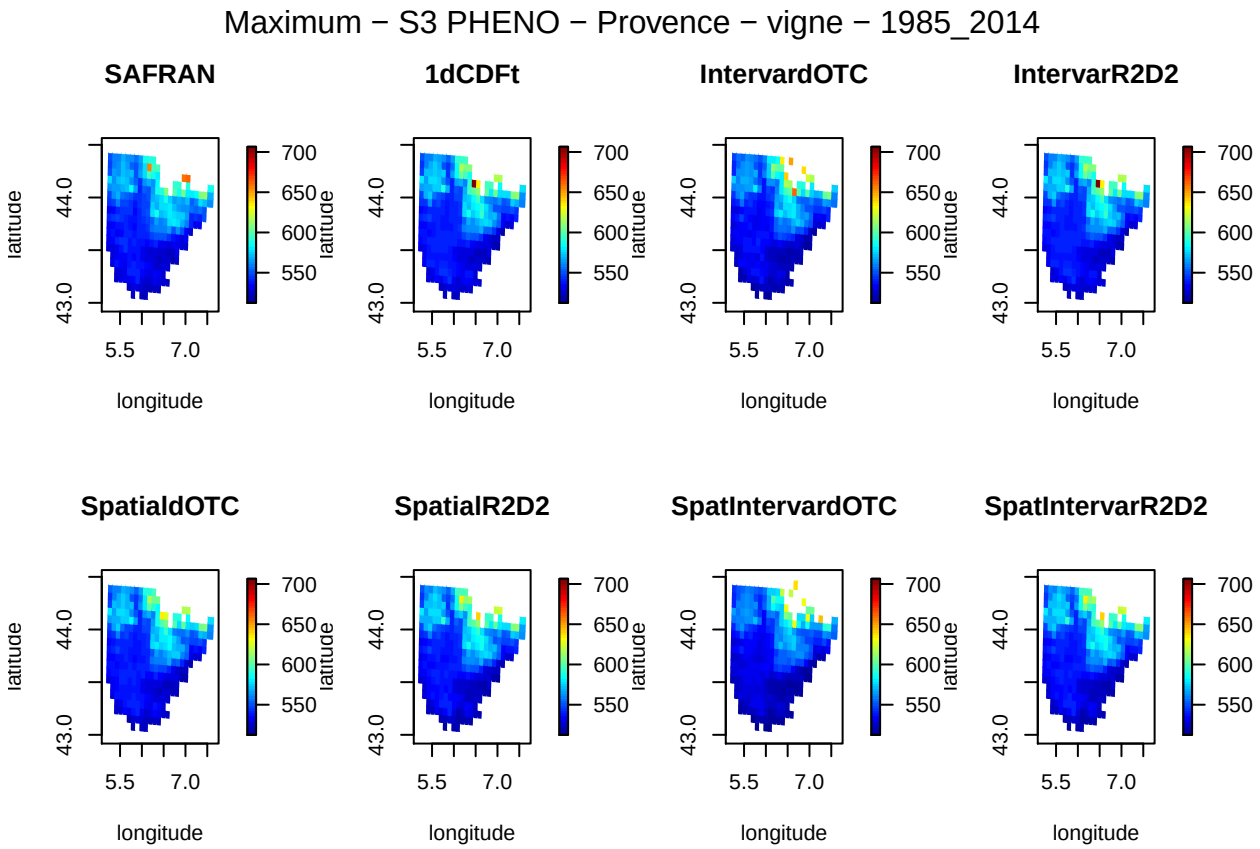


Figure 1.17

```
source("fonction_cartographie_phenologie.R")
cartographie.indicateur("Provence", "vigne", "2036_2065", "S3",
    max)
```

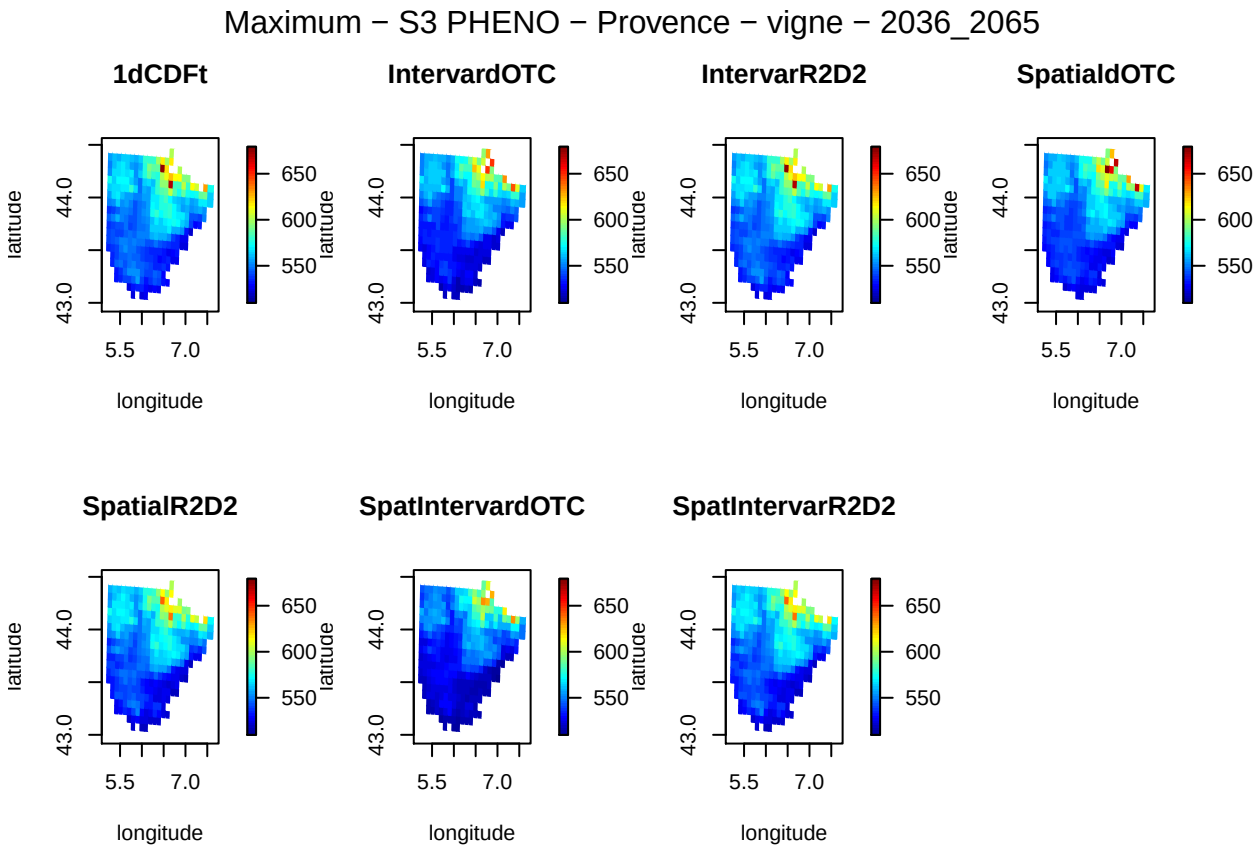


Figure 1.18

2) Observations

1. La figure 1 représente l'écart-type de l'évapotranspiration au mois de juillet de 1985 à 2014 pour le maïs en Bretagne. 1dCDFt, SpatialR2D2 et SpatialdOTC sont relativement différent de SAFRAN. A vu d'oeil, IntervardOTC, IntervarR2D2, SpatIntervardOTC, SpatIntervarR2D2 sont identiques à SAFRAN.

2. La figure 2 reprend les mêmes arguments que la figure 1 mais pour la période future. IntervarR2D2 et SpatIntervarR2D2 sont identiques, IntervardOTC et SpatialdOTC sont presque similaires, mais IntervardOTC a des sites avec des écart-types plus grands. SpatIntervardOTC et SpatIntervarR2D2, sont aussi presque similaires, mais SpatIntervarR2D2 a des sites avec des écart-types plus grands. 1dCDFt et SpatialR2D2 sont assez différents des autres.

3. La figure 3 montre l'écart-type de l'évapotranspiration au mois de juillet de 1985 à 2014 pour le blé en Ile-de-France. IntervardOTC et SpatIntervardOTC semblent identiques, et pareillement pour IntervarR2D2 et SpatIntervarR2D2. Même s'il y a certains sites où l'écart-type est différent, ceux qui ressemblent plus à SAFRAN sont IntervardOTC et SpatIntervardOTC. 1dCDFt, SpatialdOTC et SpatialR2D2 sont différents.

4. La figure 4 montre l'écart-type de l'évapotranspiration au mois de juillet de 2036 à 2065 pour le blé en Ile-de-France. IntervarR2D2 et SpatIntervarR2D2 sont similaires. 1dCDFt et IntervardOTC ont une légère ressemblance. SpatialdOTC, SpatialR2D2 , SpatIntervardOTC sont très différents.

5. La figure 5, l'écart-type de l'évapotranspiration au mois de juillet de 1985 à 2014 pour la vigne en Provence. IntervardOTC et SpatIntervardOTC sont identiques, et pareillement pour IntervarR2D2 et SpatIntervarR2D2. IntervardOTC et SpatIntervardOTC ressemblent à SAFRAN, et 1dCDFt, SpatialdOTC et SpatialR2D2 sont différents.

6. La figure 6, l'écart-type de l'évapotranspiration au mois de juillet de 2036 à 2065 pour la vigne en Provence. Seuls IntervarR2D2 et SpatIntervarR2D2 se ressemblent.

7. Dans la figure 7, il y a la moyenne de réserve en eau du sol en juillet de 1985 à 2014 pour le maïs en Bretagne. Il y a une légère ressemblance entre 1dCDFt et IntervardOTC. Aucune correction ne ressemble à SAFRAN.

8. La figure 8 représente les mêmes arguments de la figure 7, mais de 2036 à 2065. Toutes les corrections sont différentes, il n'y qu'une ressemble minime entre 1dCDFt et SpatialR2D2 .

9. La figure 9, la moyenne de réserve en eau du sol en juillet de 1985 à 2014 pour le blé en Ile-de-France. Toutes les corrections sont plus ou moins différentes et aucune n'est identique à SAFRAN. Il y a une ressemblance entre 1dCDFt et IntervardOTC et SpatialdOTC et SpatialR2D2.

10. La figure 10, la moyenne de réserve en eau du sol en juillet de 2036 à 2065 pour le blé en Ile-de-France. Toutes les corrections sont différentes. Il y a une légère similarité au niveau de l'échelle pour IntervarR2D2 et SpatIntervarR2D2, et 1dCDFt et SpatialR2D2.

11. Dans la figure 11, il y a la moyenne de réserve en eau du sol en juillet de 1986 à 2014 pour la vigne en Provence. Toutes les correction et SAFRAN sont très ressemblantes. IntervarR2D2 et SpatIntervarR2D2 sont identiques, et aussi, IntervardOTC et SpatIntervardOTC.

12. Dans la figure 12, il y a la moyenne de réserve en eau du sol en juillet de 2037 à 2065 pour la vigne en Provence. Les corrections se ressemblent, surtout 1dCDFt et SpatialR2D2.

13. La figure 13 représente le maximum du stade 3 de la phénologie de 1985 à 2014 pour le maïs en Bretagne. SpatIntervardOTC est différente des autres cartes. Il y a des différences entre toutes les cartes, mais il y a une forte ressemblance entre SAFRAN et IntervardOTC. 1dCDFt et IntervarR2D2 semblent identiques, c'est aussi le cas pour, SpatialR2D2 et SpatIntervarR2D2.

14. La figure 14 représente le maximum du stade 3 de la phénologie de 2036 à 2065 pour le maïs en Bretagne. Les ressemblances entre les corrections sont les mêmes que ceux de la figure 13. SpatIntervardOTC ne ressemble à aucune autre correction.

15. Dans la figure 15, il y a le maximum du stade 3 de la phénologie de 1986 à 2014 pour le pour le blé en Ile-de-France. Aucune des corrections ne ressemble à SAFRAN. 1dCDFt et IntervarR2D2 sont similaires, et SpatialR2D2 et SpatIntervarR2D2 aussi. SpatialdOTC ressemble aux autres corrections de protocole Spatial, mais avec une légère différence sur quelques sites. SpatIntervardOTC est très différent des autres cartes, avec une échelle relativement basse.

16. La figure 16 montre 13 représente le maximum du stade 3 de la phénologie de 2037 à 2065 pour le pour le blé en Ile-de-France. Les observations sont identiques à ceux de la figure 15.

17. Dans la figure 17, le maximum du stade 3 de la phénologie de 1986 à 2014 pour la vigne en Provence est représentée. 1dCDFt et IntervarR2D2 sont identiques. SpatialR2D2 et SpatIntervarR2D2 sont aussi identiques. SpatialdOTC ressemble aux autres corrections de protocole Spatial, mais avec une légère différence sur quelques sites.IntervardOTC et SpatIntervardOTC sont différents. Aucune correction n'est identique à SAFRAN.

18. La figure 18, représente le maximum du stade 3 de la phénologie de 2037 à 2065 pour la vigne en Provence. 1dCDFt et IntervarR2D2 sont identiques. SpatialR2D2 et SpatIntervarR2D2 sont aussi identiques. IntervardOTC, SpatIntervardOTC, SpatialdOTC sont différents.

L'écart-type de l'évapotranspiration montre que la période, seulement, peut changer les similarités entre les graphiques. Pour cette statistique d'indicateur, les corrections qui ressemblent le plus à SAFRAN sont IntervardOTC et SpatIntervardOTC. Aussi, IntervarR2D2 et SpatIntervarR2D2 sont identiques pour toutes régions, cultures, et période présentées.

La moyenne de réserve en eau permet de voir que dans le futur, 1dCDFt et SpatialR2D2 sont similaires.

La moyenne de réserve en eau de la vigne en Provence ne permet pas d'illustrer les différences entre corrections, surtout dans le passé. En effet, même lorsque les corrections sont affichées

par année (et pas sur 29 ans comme dans la cartographie), il n'y a pas de différence notable (voir Annexe, Figure 1.19-1.21), sauf pour quelques années (1986, 1987, 1989, 2006).

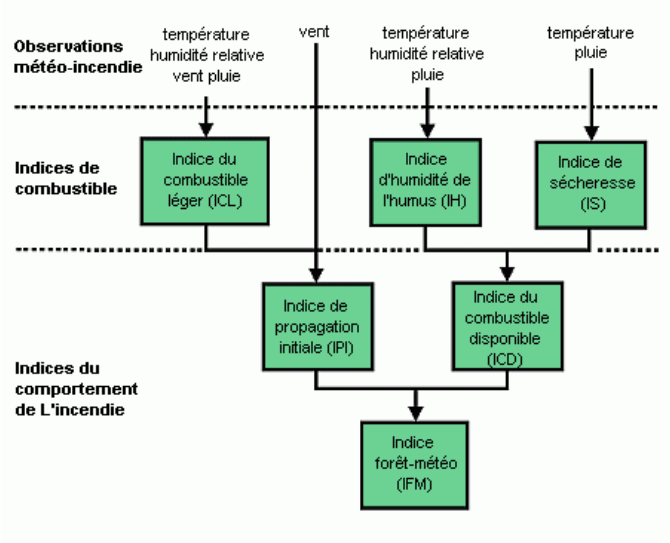
Pour le maximum du stade 3 de la phénologie, toutes cultures, régions et périodes, Spat-IntervardOTC toujours très différentes des autres graphiques. Dans les mêmes conditions, 1dCDFt et IntervarR2D2 sont identiques. Pour le maïs en Bretagne et la vigne en Provence, il y a une certaine similarité entre SAFRAN et IntervardOTC.

Partie II. Indicateur de risque de feux de forêt

Cette partie du stage consiste à calculer L'Indice Forêt Météo (IFM) pour les 3 régions (Bretagne, Ile-de-France, Provence). L'IFM est un indicateur du risque d'occurrence d'un feu de forêt. Il est calculé à partir de données météorologiques simples : température, humidité de l'air, vitesse du vent et précipitations.

Sur R, l'IFM se calcul avec la fonction fwi(). Cette fonction génère jusqu'à 7 indicateurs des feux de forêts, mais nous considérons seulement les indicateurs IFM et IJS (l'indice journalier de sévérité).

Les indicateurs générés par la fonction fwi sont interdépendants, le schéma ci-dessous résume cette relation:



Indice forêt météo
L'indice forêt-météo (IFM), Fire Weather Index (FWI) en anglais, est une évaluation numérique de l'intensité du feu. Il repose sur l'IPI et l'ICD, et il constitue un indice général du danger d'incendie sur un ensemble de région. Cet indicateur n'a pas d'unité de mesure. Plus il est élevé, plus il y a de chance d'avoir un feu de forêt. L'IFM n'est pas borné et peut varier de 0 à des valeurs très grandes. Les seuils critiques de danger très élevés dépendent des zones climatiques.

Indice journalier de sévérité
L'indice journalier de sévérité (IJS), Daily Severity Rating (DSR) en anglais, est un élément supplémentaire de la Méthode de l'IFM, et il constitue une évaluation numérique de la difficulté à maîtriser les incendies. Il repose sur l'IFM, mais il présente de manière plus précise les efforts attendus requis pour éteindre les incendies.

A. Création des données

Nous préparons un script, prepare_datafwi.r, qui calcule l'IFM en fonction de la région et période déclarée. Le script récupère les variables climatiques préparées dans la partie 1. Les

variables climatiques utilisées pour calculer cet indicateur sont: la température, l'humidité, le vent et les précipitations.

Cette étape a nécessité de télécharger plusieurs packages: rgdal, sp, raster, foreach, geosphere, mmap, spatial.tools, cffdrs, dplyr, tibble, tidyverse.

La fonction fwi() du package requiert que les données soient dans un dataframe de 10 colonnes, dans un ordre précis. Selon ce format, nous avons créé un dataframe avec les colonnes suivantes:

- Id : numéro des sites
- lat : latitude des sites
- long : longitude des sites
- yr : année
- mon : mois
- day : jour
- temp : température
- rh : humidité
- ws : vent
- prec : précipitations

Les feux de forêts ayant lieu généralement en été, nous choisissons de garder seulement les mois de juin, juillet et août. Donc, nous filtrons les lignes.

Le nombre lignes égales le nombre de sites par le nombre de jour. Le nombre de jour égale à 2760 (92 jours pour les trois mois par 30 ans).

Lorsque le dataframe est prêt, il est inséré dans la fonction fwi().La fonction calcule plusieurs indicateurs, mais nous enregistrons seulement le FWI et DSR dans nos fichiers.

```
load("C:/Users/Ling/Desktop/Stage INRAE/Dossier de travail/Code/Compromise/B
     retagne/fwi/fwi_1dCDFt_1985_2014.RData")
```

```
head(datalist)
```

```
##      ID   YR MON DAY       FWI         DSR
## 1     1 1985  06  01 0.7091265 0.01480300
## 93    2 1985  06  01 0.7090105 0.01479872
## 185   3 1985  06  01 0.6306577 0.01202829
## 277   4 1985  06  01 0.6773289 0.01364848
## 369   5 1985  06  01 0.6822248 0.01382358
## 461   6 1985  06  01 0.6881849 0.01403806
```

Cette étape de création de données est répétées pour chaque correction (avec une boucle). Nous recommençons ensuite pour chaque région et chaque période.

Les données sont sauvegardées dans un nouveau répertoire 'fwi'.

```
Dossier de travail
 °--Code
     °--Compromise
         ¦--Bretagne
         ¦   ¦--ble
         ¦   ¦--fwi
         ¦   ¦   °--fwi_1dCDFt_1985_2014.RData
         ¦   ¦--mais
         ¦   °--vigne
         ¦--Ile_de_France
         °--Provence
```

B. Création d'une fonction

Pour visualiser les indicateurs calculés, nous procédons à la même démarche que la partie 1.

Nous créons un fonction qui permet la statistique de l'indicateur voulu sur une période.

Fonction: cartographie.indicateur
Script: cartographie_fwi.r
Fichier source: fonction_cartographie_fwi.r

La fonction: Vectorise les valeurs de l'indicateur pour chaque site sur les 30 ans,
Calcule la statistique sur ce vecteur
Mets dans un dataframe la statistique calculée pour chaque site
Affiche la cartographie

```
cartographie.indicateur(region, periode, indicateur, statistique)
```

region	le nom de la region dans ce format: "Bretagne", "Ile_de_France", "Provence".
periode	passé ou futur en choisissant entre "1985_2014" ou "2036_2065".
indicateur	l'indicateur qui souhaite être représenté, "FWI" pour l'indice forêt-météo, "DSR" pour l'indice journalier de sévérité.
statistique	la fonction R de ce qui doit être calculé: mean, median, max, min, sd.

C. Visualisation de l'indice forêt-météo

1) Cartographies

Je choisis de représenter le maximum de l'IFM. Cela correspond à l'intensité maximale des feux de forêts potentiels sur les sites des 3 régions, pour chaque correction et SAFRAN.

```
source("fonction_cartographie_fwi.R")
cartographie.indicateur("Bretagne", "1985_2014", "FWI", max)
```

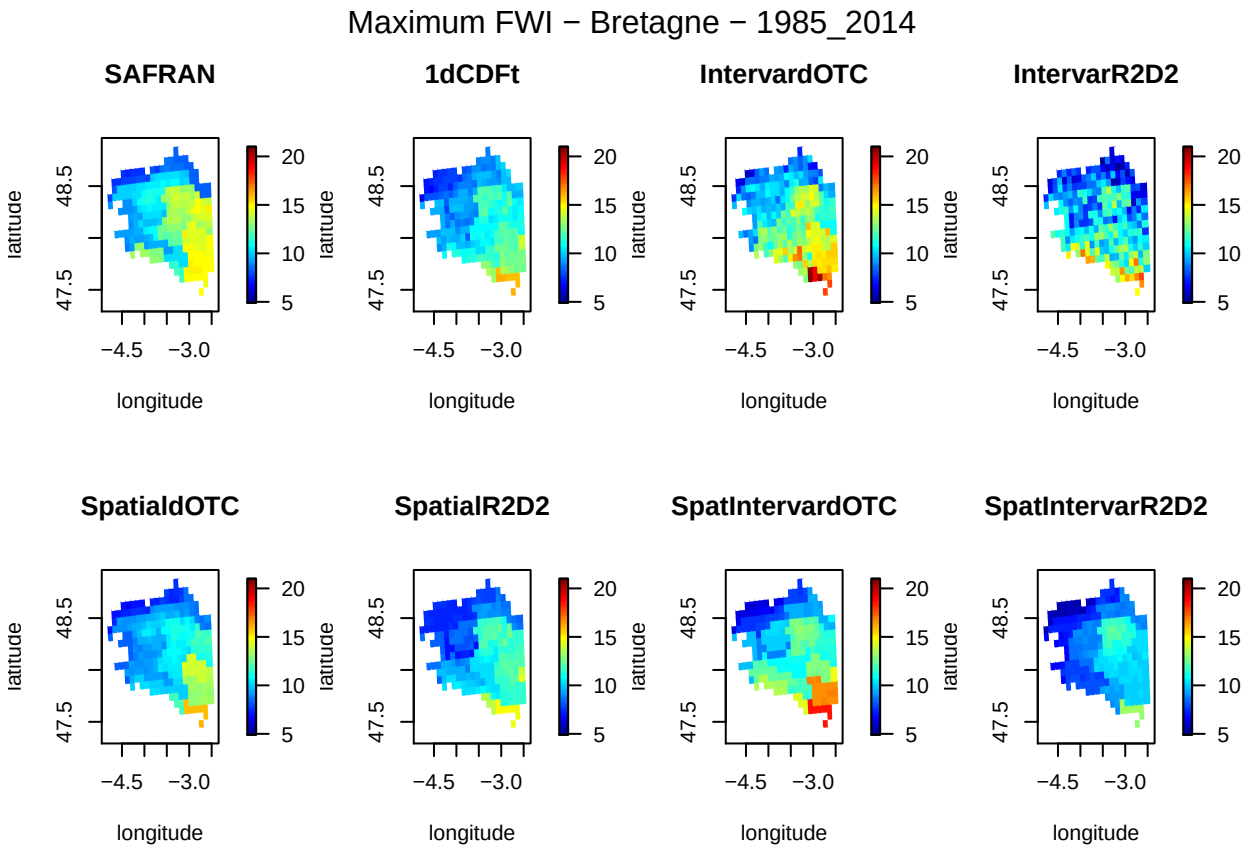


Figure 2.1

```
source("fonction_cartographie_fwi.R")
cartographie.indicateur("Bretagne", "2036_2065", "FWI", max)
```

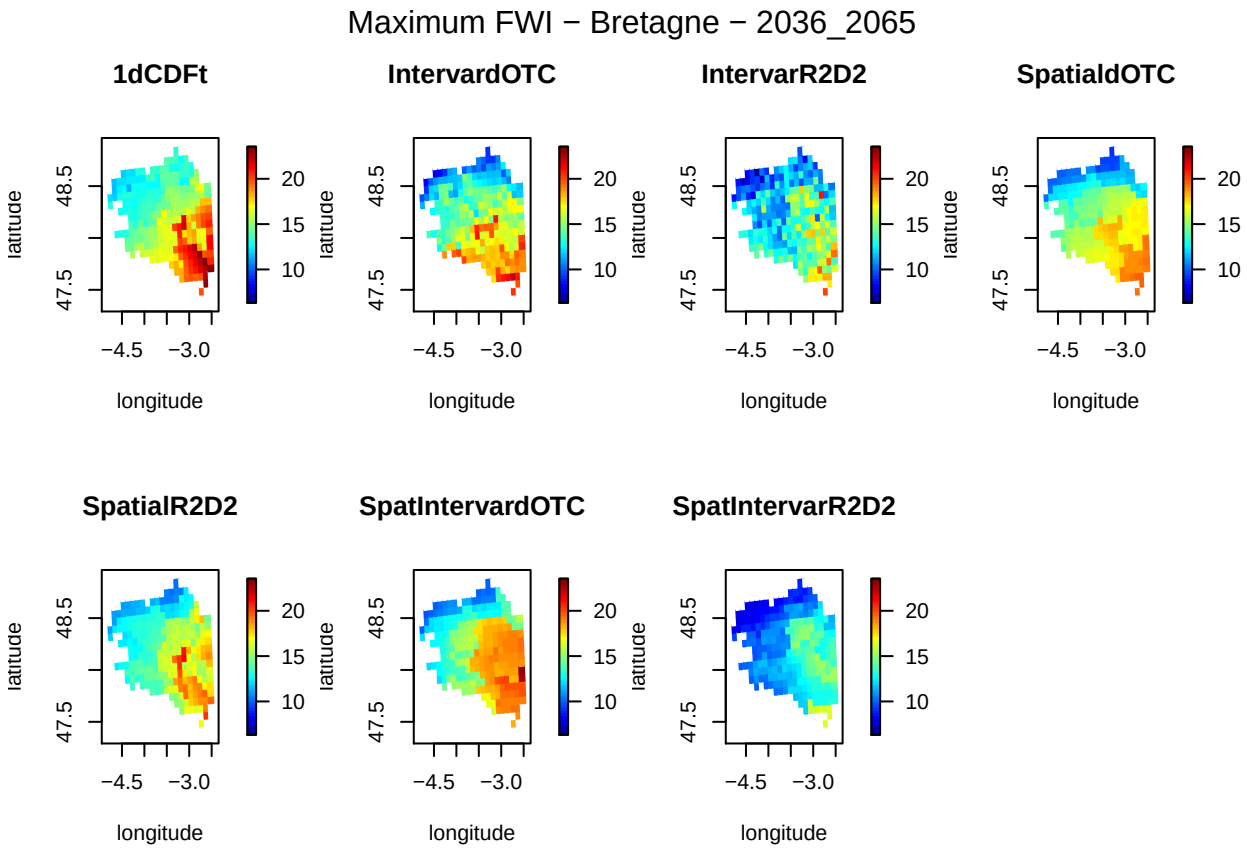


Figure 2.2

```
source("fonction_cartographie_fwi.R")
cartographie.indicateur("Ile_de_France", "1985_2014", "FWI",
    max)
```

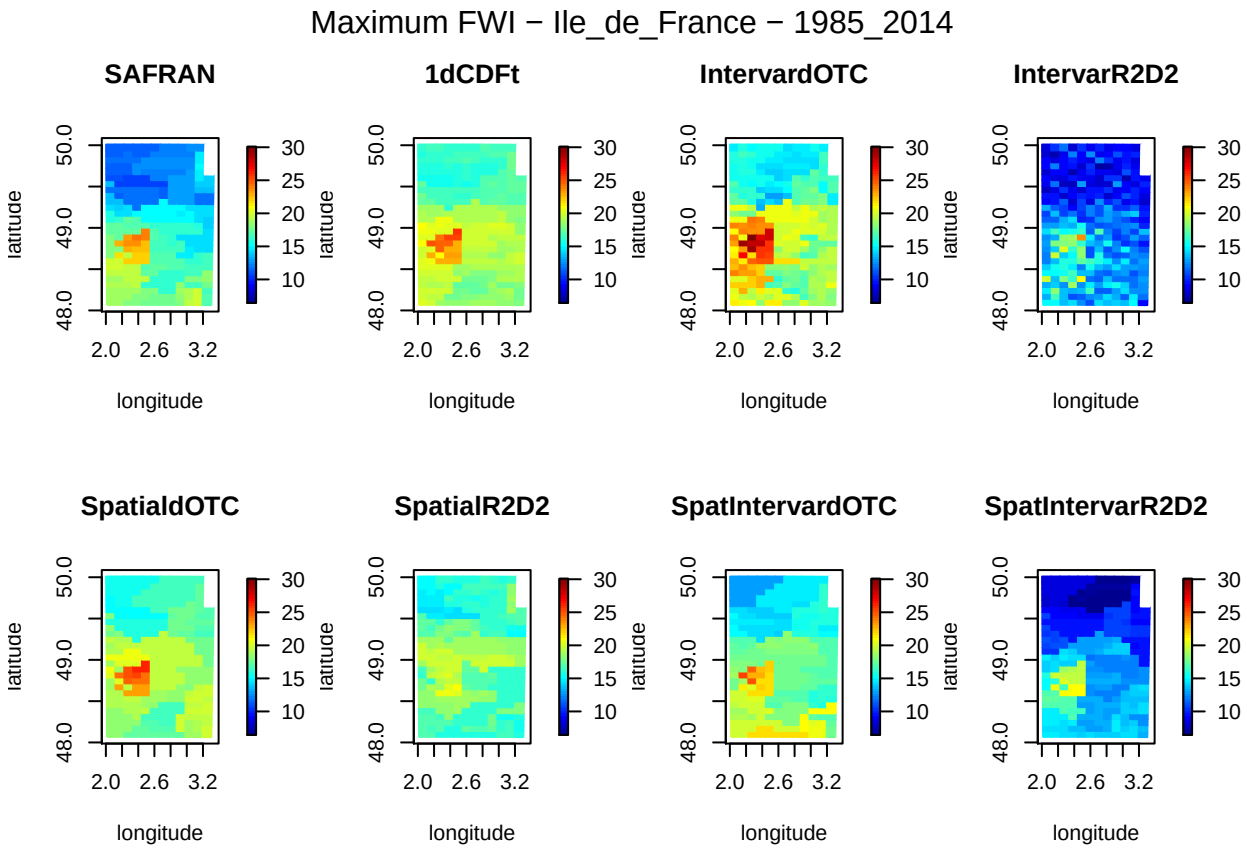


Figure 2.3

```
source("fonction_cartographie_fwi.R")
cartographie.indicateur("Ile_de_France", "2036_2065", "FWI",
    max)
```

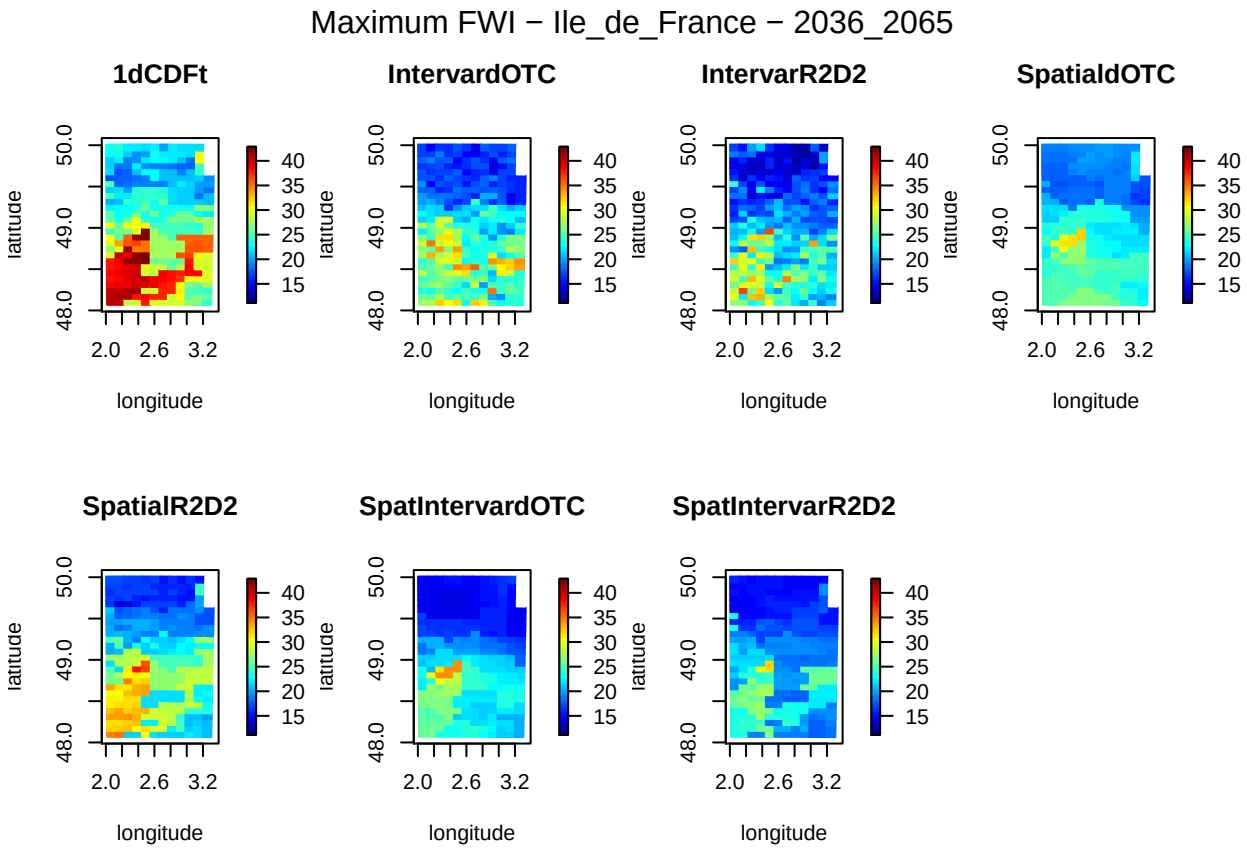


Figure 2.4

```
source("fonction_cartographie_fwi.R")
cartographie.indicateur("Provence", "1985_2014", "FWI", max)
```

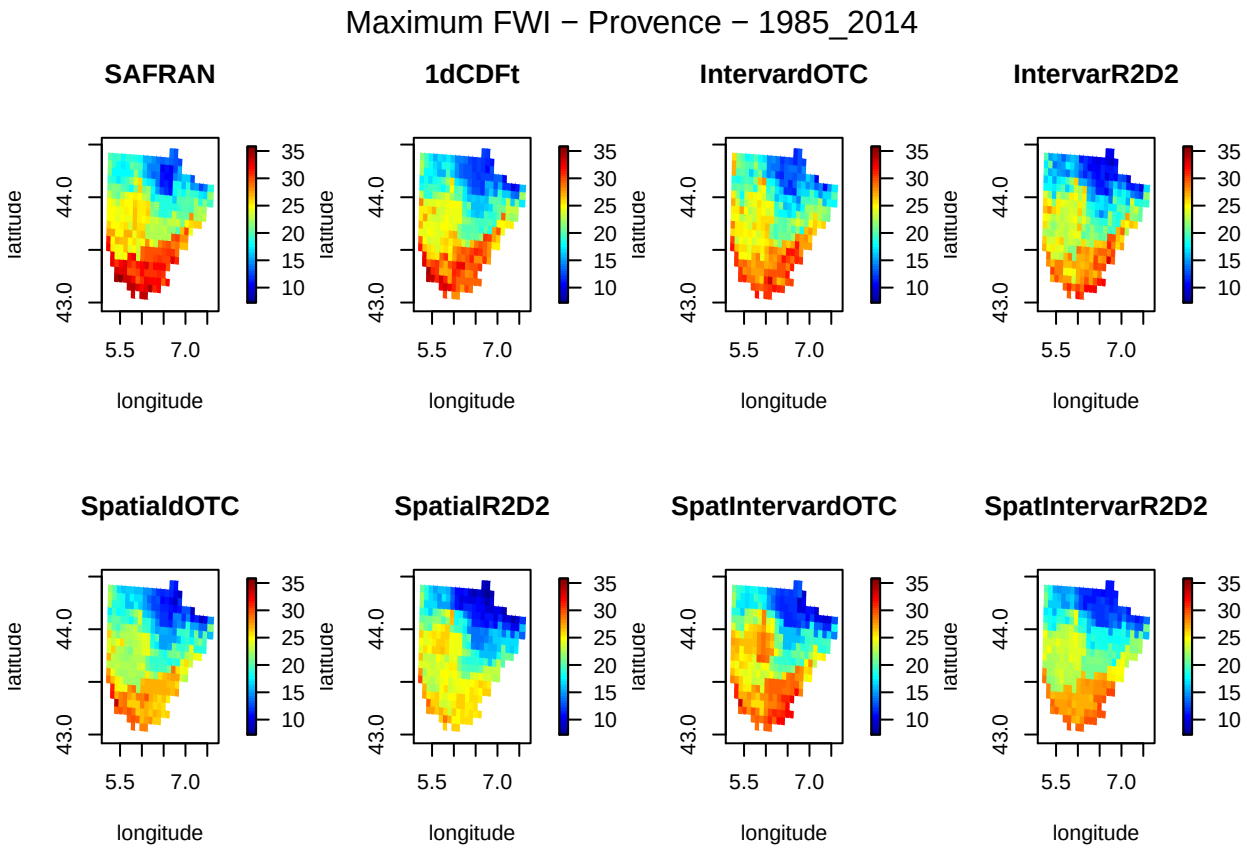


Figure 2.5

```
source("fonction_cartographie_fwi.R")
cartographie.indicateur("Provence", "2036_2065", "FWI", max)
```

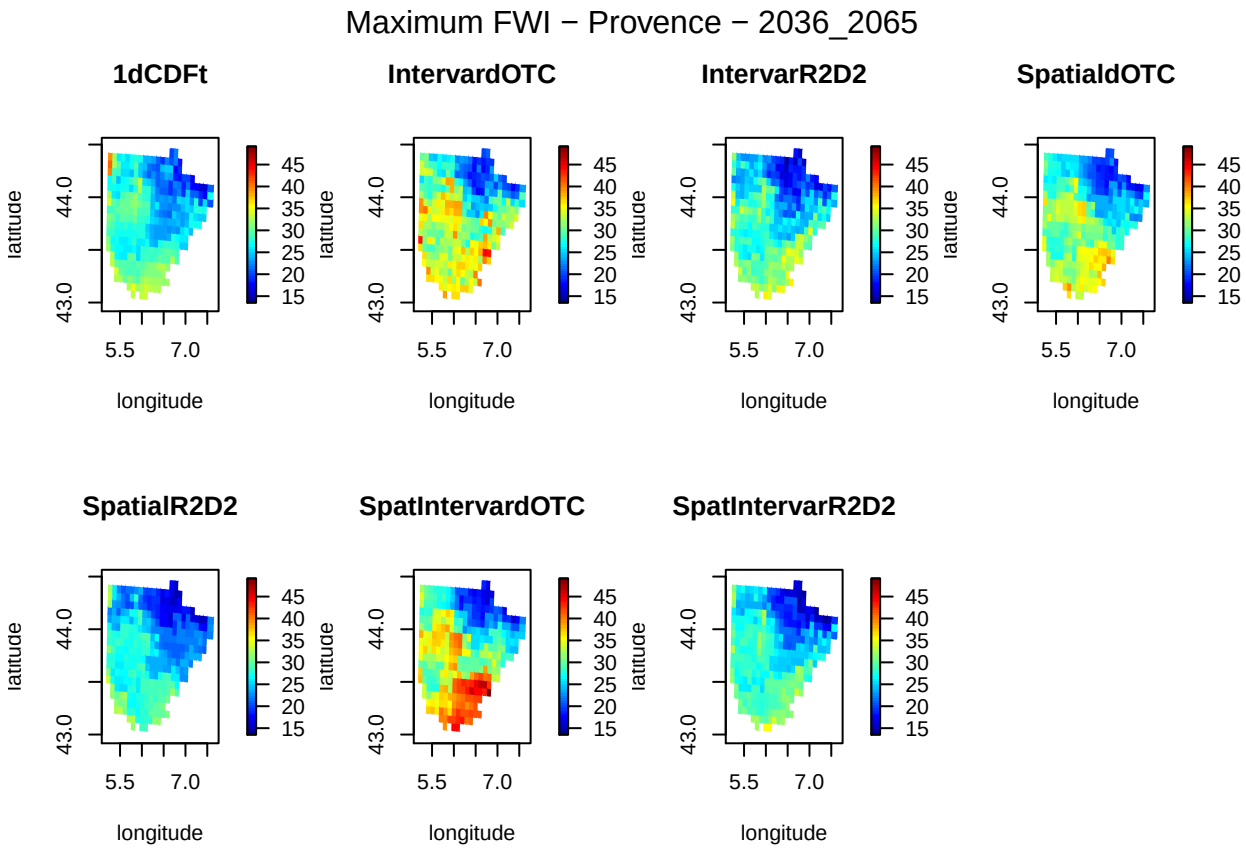


Figure 2.6

2) Observations

1. Dans la figure 2.1, il y a le maximum de l'indice forêt météo de 1985 à 2014 pour la Bretagne. Aucune des corrections ne ressemble à SAFRAN. Les corrections les plus différentes sont IntervardOTC, SpatIntervardOTC et IntervarR2D2 avec un IFM supérieur à 16 sur certains sites. Les reste des corrections ont des IFM entre 5 et 16.

2. La figure 2.2 représente le maximum de l'indice forêt météo de 2036 à 2065 pour la Bretagne. SpatIntervarR2D2 a des valeurs IFM inférieur à 16 sur l'ensemble des site, comme dans la période passée. Le reste des corrections ont des IFM qui sont différents de ceux du passé.

3. La figure 2.3 montre l'indice forêt météo de 1985 à 2014 pour la l'Ile-de-France. Il y a une différence visible entre toutes les cartes. 1dCDFt et SpatialdOTC ont une légère ressemblance. IntervarR2D2 et SpatIntervarR2D2 sont très différents.

4. Dans la figure 2.4, il y a l'indice forêt météo de 2036 à 2065 pour la l'Ile-de-France. Toutes les corrections sont différentes. 1dCDFt a des valeurs relativement très élevées en comparaison aux autres corrections.

5. La figure 2.5 représente le maximum de l'indice forêt météo de 1985 à 2014 pour la Provence. Relativement à toutes les corrections et SAFRAN, IntervardOTC et IntervarR2D2 ont une légère ressemblance. Aucune correction ne reproduit parfaitement la carte SAFRAN, mais 1dCDFt y ressemble faiblement en regardant la totalité des sites.

6. La figure 2.6 montre les mêmes arguments que la figure 2.5, mais pour le futur. SpatialR2D2 et SpatIntervarR2D2 se ressemblent faiblement, et 1dCDFt et IntervarR2D2 aussi.IntervardOTC et SpatIntervardOTC sont très différents des autres corrections.

Dans les graphiques 2.1, 2.3 et 2.6, nous pouvons observer que les corrections spatiales ont une légère ressemblance à 1dCDFt.

Pour le maximum de l'IFM, indépendamment des régions et des périodes, la différence entre les corrections est assez remarquable. La période de 2036 à 2065 accentue cette différence.

Bilan

Effectuer mon stage de fin d'études de DUT Statistique et Informatique Décisionnelle (STID) à INRAE a été une expérience enrichissante, tant sur le plan professionnelle que le plan personnel. F. Garrel et moi avons travaillé en binôme tout au long du stage. Il a fallu apprendre à s'adapter à la façon de faire de chacun, tout en prenant des décisions à deux. Il y a eu des moments où nous n'étions pas toujours en accord, mais le maître de stage a su nous conseiller dans ces moments-là.

Le stage a été grandement réalisé en télétravail et il a été impératif de s'imposer un cadre de travail. Même si nous nous retrouvions deux fois par semaine pour des sessions de travail (de deux heures environ), il a fallu apprendre à s'organiser pour rendre le travail à temps, ce qui n'est pas toujours évident lorsqu'on travail chez soi.

Avec le suivi continu de D. Allard et E. Walker, le confinement n'a pu impacter négativement notre productivité.

Lors des 9 semaines de stage, F. Garrel et moi avons mis en pratique nos connaissances acquises durant les deux années de DUT STID. Toutes les missions ont été réalisées avec le logiciel R. De mon côté, j'ai décidé ensuite de rédiger mon rapport de stage avec Rmarkdown, qui utilise le langage Latex.

Le stage a principalement consisté à la manipulation de données et programmation en R. Le tout ayant pour but de construire des fonctions permettant de visualiser des indicateurs agroclimatiques/de feux de forêts.

Nous avons écrit en R des fonctions qui permettent de visualiser des statistiques (moyenne, médiane, maximum, minimum, écart-type) d'indicateurs (phénologie, évapotranspiration, bilan hydrique) sur les 3 régions (Bretagne, Ile-de-France, Provence) pour l'ensemble des cultures (maïs, blé, vigne).

Ces fonctions permettent de visualiser les différentes méthodes de correction de biais et leurs effets sur différents processus environnementaux. Ce travail sert à pouvoir comparer les méthodes de corrections à travers des cartographies et boxplots, et éventuellement servira à créer d'autres fonctions pour générer d'autres graphiques.

Annexe

```
source("fonction_boxplot_phenologie.R")
boxplot.indicateur("Provence", "vigne", "1985_2014", "S3", 1986,
    2000)
```

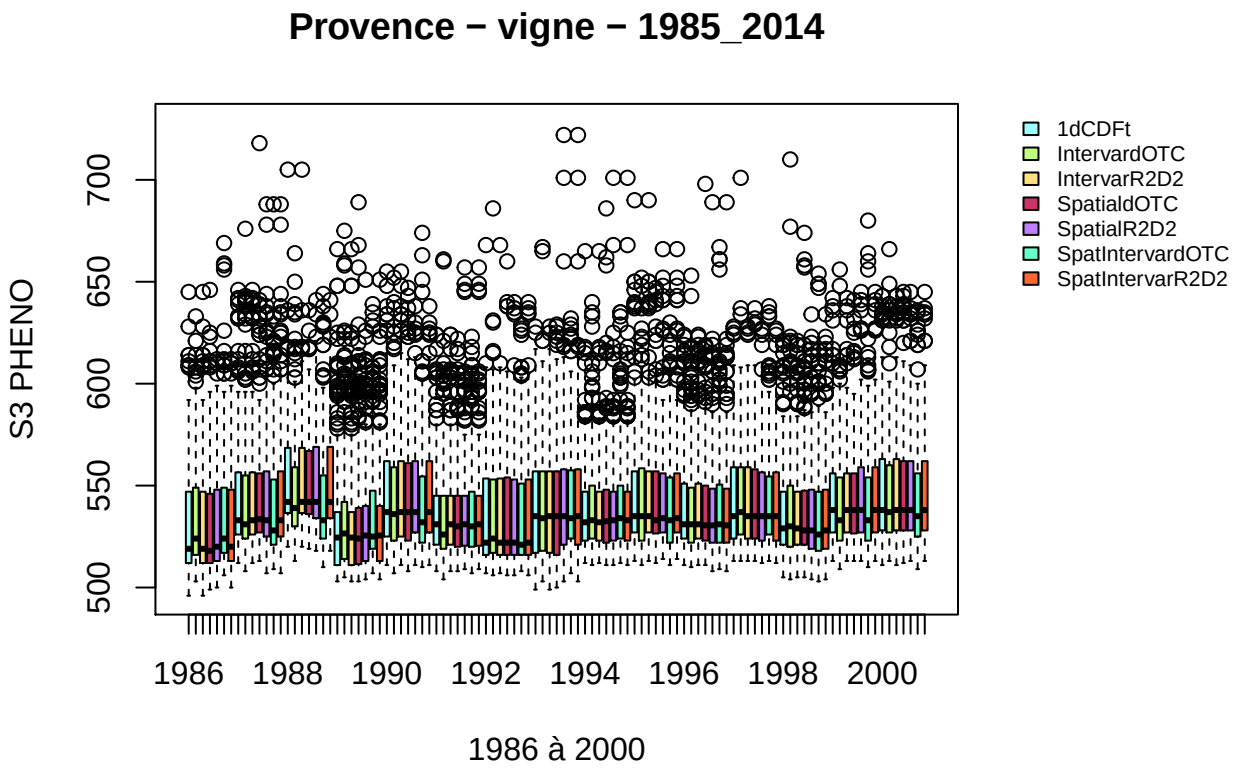


Figure 1.19

```
source("fonction_boxplot_phenologie.R")
boxplot.indicateur("Provence", "vigne", "1985_2014", "S3", 2001,
    2014)
```

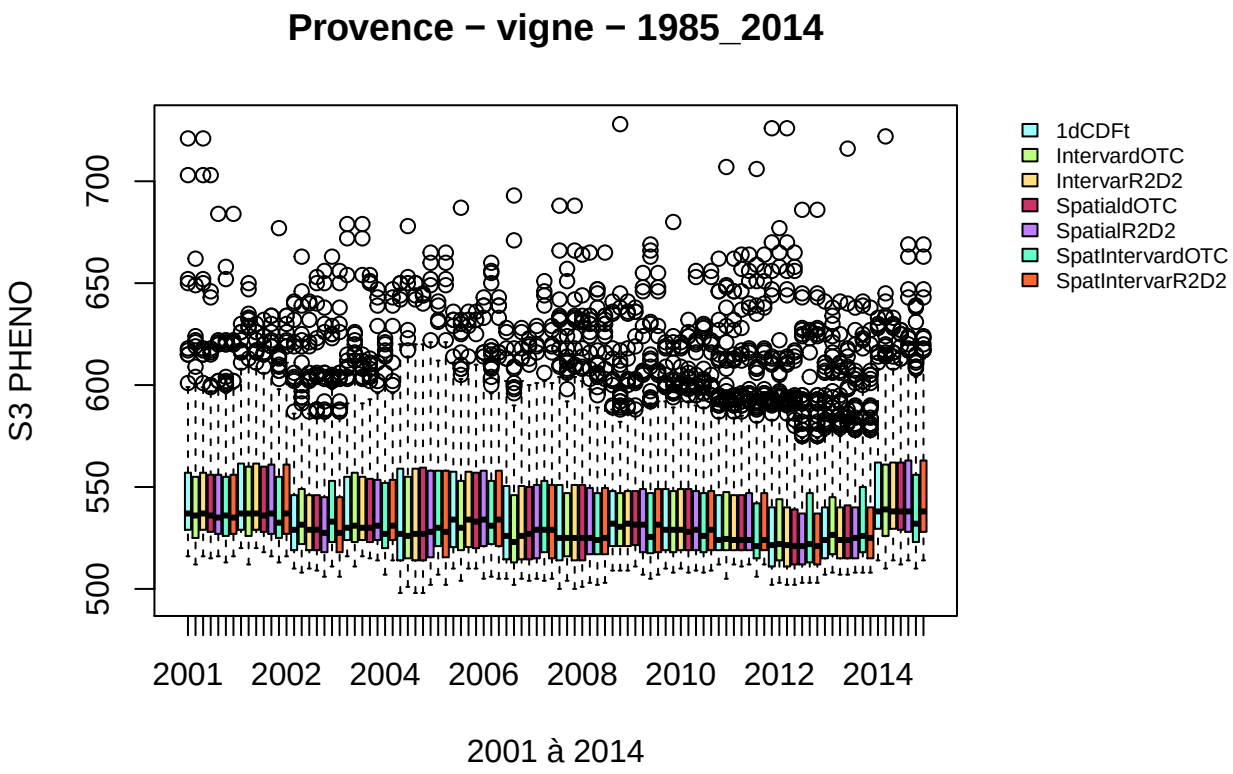


Figure 1.20

```
repinitial <- c("C:/Users/Ling/Desktop/Stage INRAE/Dossier de travail")
reptravail <- paste(repinitial, "/Code/", sep = "")
setwd(reptravail)
source("check_variable_climatiques_cartographie.R")
par(mfrow = c(2, 4), mar = c(5.1, 4, 5.1, 4))
cartographie.varclim("Ile_de_France", "1985_2014")
```

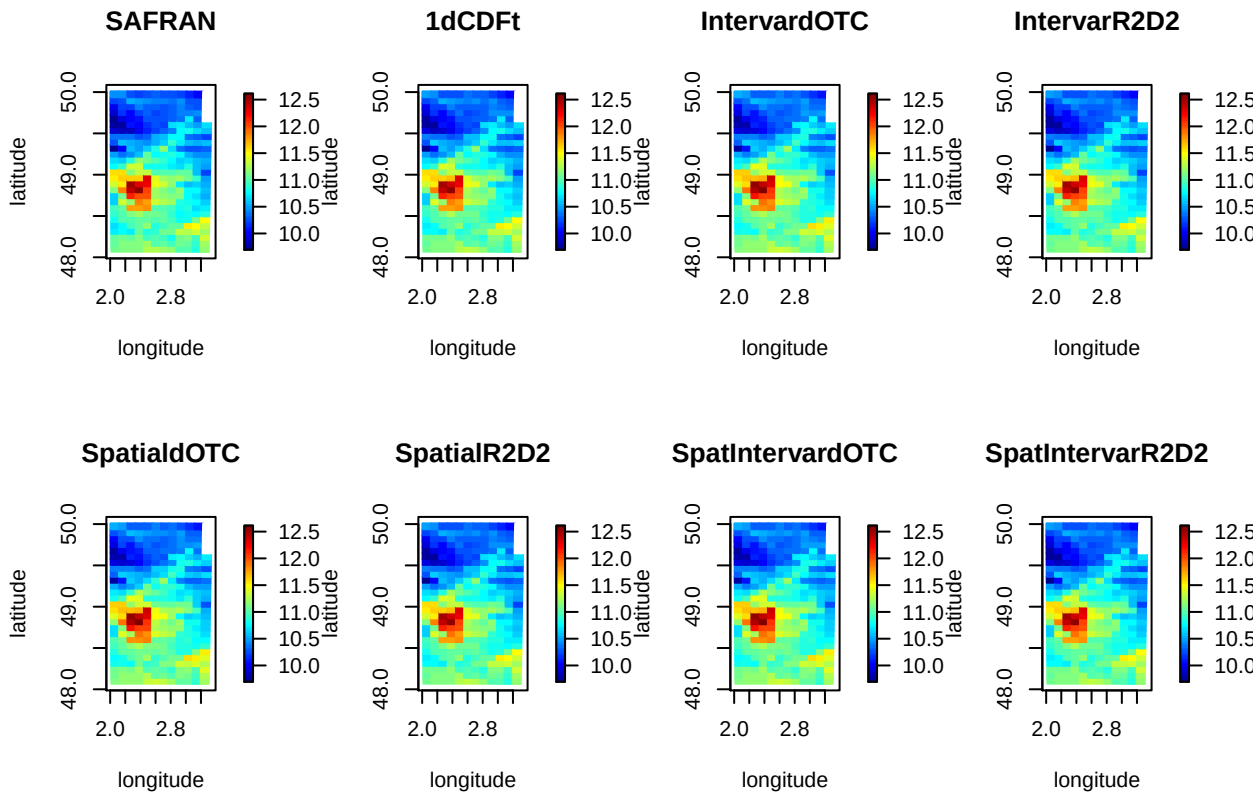
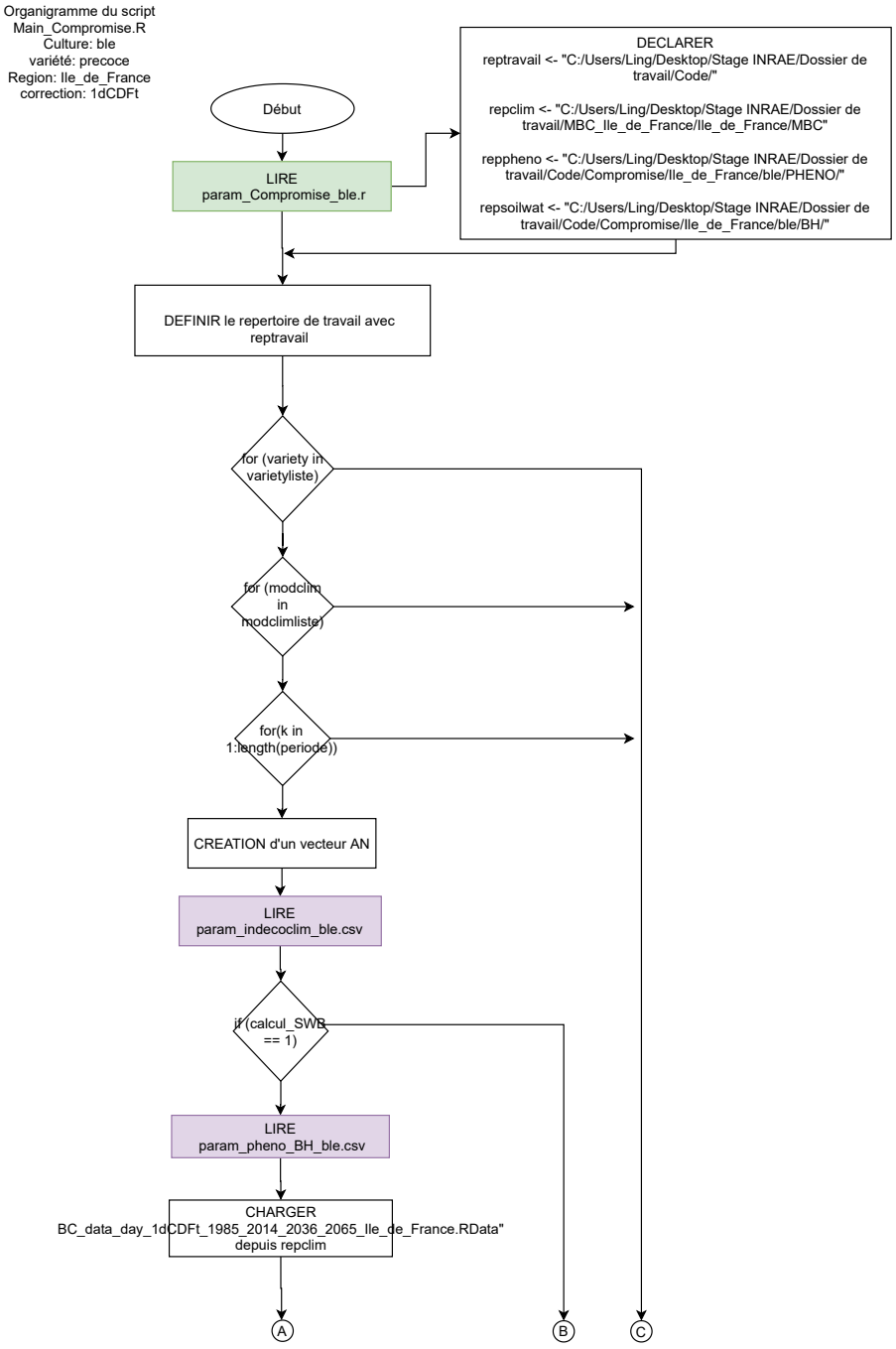
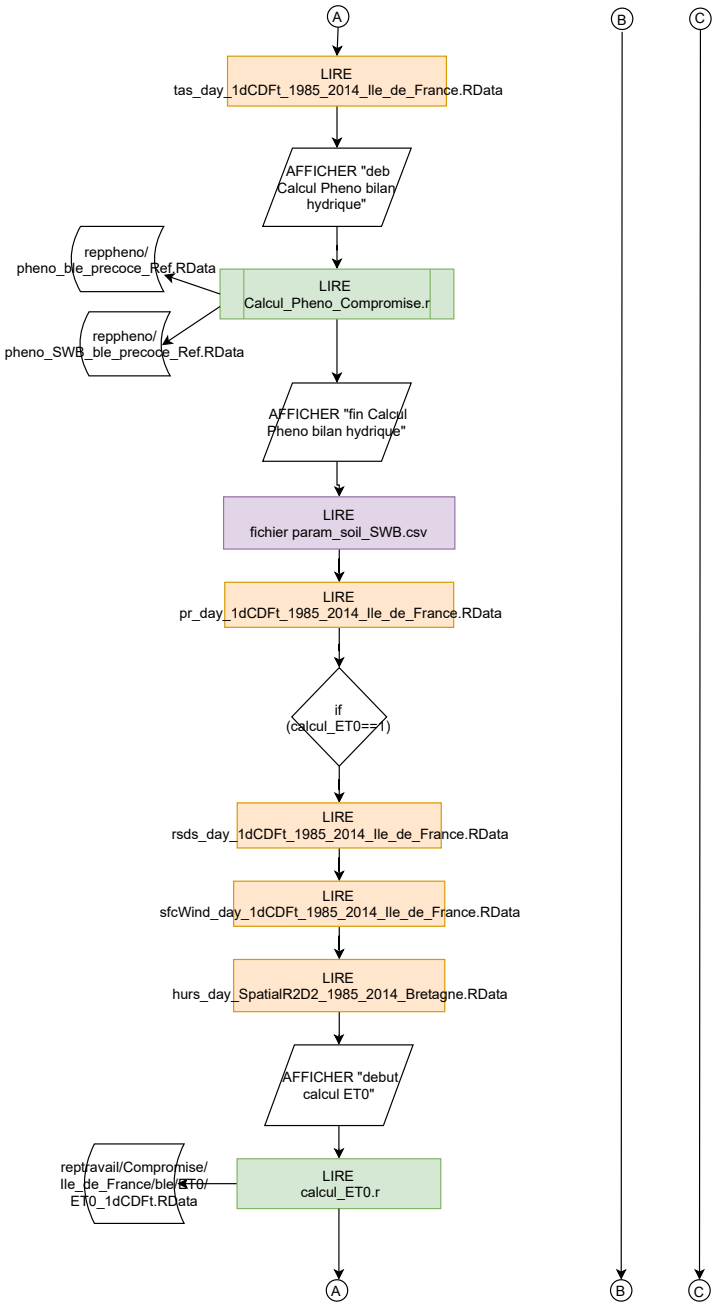
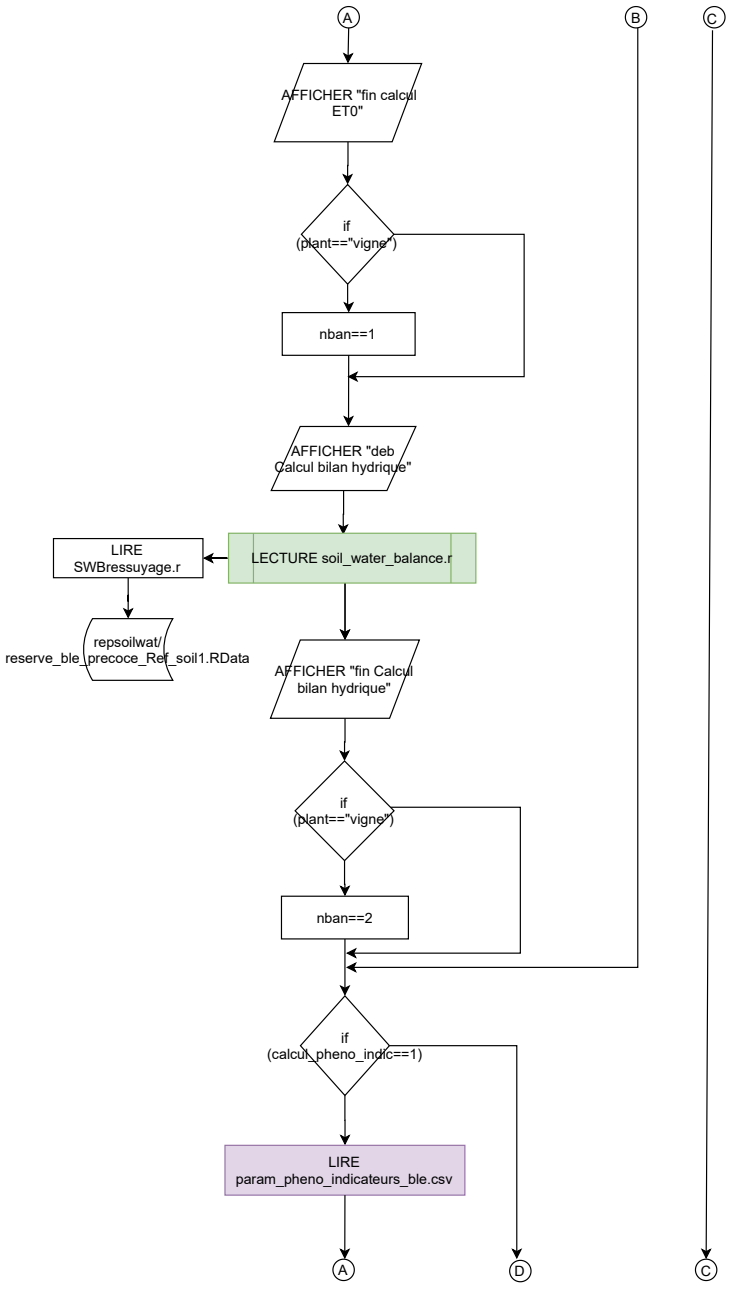


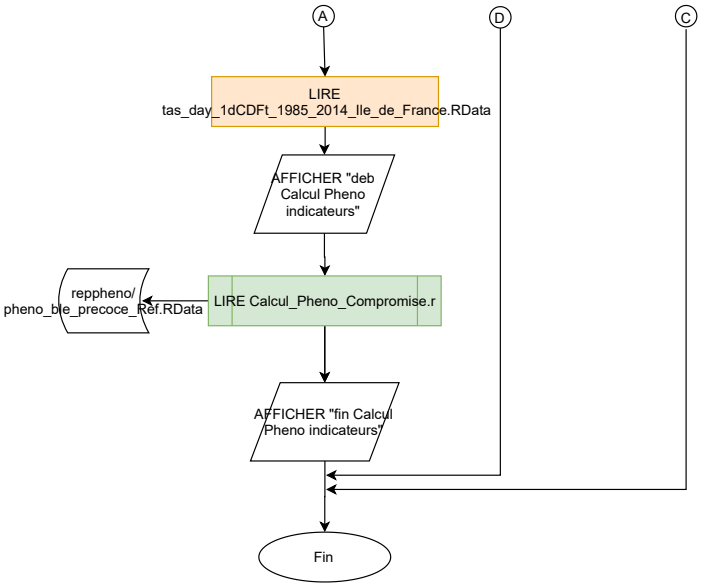
Figure 3

La script check_variable_climatiques_cartographie.R a été créé dans le but de vérifier la température en Ile-de-France. Cette vérification a eu lieu, car sur toutes les cartographies de cette région, il y a toujours les mêmes sites qui ont une valeur d'indicateur (phénologie, évapotranspiration, etc.) différente des autres sites. Avec les latitudes et longitudes, nous avons constaté que ces sites représentent la région Parisienne (latitude 48.8534; longitude 2.3488)


Organigramme du script
Main_Compromise.R
Culture: ble
variété: precoce
Region: Ile_de_France
correction: 1dCDFt
Début
LIRE
param_Compromise_ble.r
DECLARER
reptravail <- "C:/Users/Ling/Desktop/Stage INRAE/Dossier de travail/Code/"
repclim <- "C:/Users/Ling/Desktop/Stage INRAE/Dossier de travail/MBC_Ile_de_France/Ile_de_France/MBC"
reppheno <- "C:/Users/Ling/Desktop/Stage INRAE/Dossier de travail/Code/Compromise/Ile_de_France/ble/PHENO/"
repsoilwat <- "C:/Users/Ling/Desktop/Stage INRAE/Dossier de travail/Code/Compromise/Ile_de_France/ble/BH/"
DEFINIR le repertoire de travail avec reptravail
for (variety in varietyliste)
for (modclim in modclimliste)
for(k in 1:length(periode))
CREATION d'un vecteur AN
LIRE
param_indecoclim_ble.csv
if (calcul_SWB == 1)
LIRE
param_pheno_BH_ble.csv
CHARGER
BC_data_day_1dCDFt_1985_2014_2036_2065_Ile_de_France.RData"
depuis repclim
A
B
C


A
B
C
LIRE
tas_day_1dCDFt_1985_2014_Ile_de_France.RData
AFFICHER "deb
Calcul Pheno bilan
hydrique"
reppheno/
pheno_ble_precoce_Ref.RData
LIRE
Calcul_Pheno_Compromise.r
reppheno/
pheno_SWB_ble_precoce_Ref.RData
AFFICHER "fin Calcul
Pheno bilan hydrique"
LIRE
fichier param_soil_SWB.csv
LIRE
pr_day_1dCDFt_1985_2014_Ile_de_France.RData
if
(calcul_ET0==1)
LIRE
rsds_day_1dCDFt_1985_2014_Ile_de_France.RData
LIRE
sfcWind_day_1dCDFt_1985_2014_Ile_de_France.RData
LIRE
hurs_day_SpatialR2D2_1985_2014_Bretagne.RData
AFFICHER "debut
calcul ET0"
rep/travail/Compromise/
Ile_de_France/ble/ET0/
ET0_1dCDFt.RData
LIRE
calcul_ET0.r
A
B
C


A
B
C
AFFICHER "fin calcul ET0"
if (plant=="vigne")
nban==1
AFFICHER "deb Calcul bilan hydrique"
LIRE SWBressuyage.r
LECTURE soil_water_balance.r
repsoilwat/ reserve_ble_precoce_Ref_soil1.RData
AFFICHER "fin Calcul bilan hydrique"
if (plant=="vigne")
nban==2
if (calcul_pheno_indic==1)
LIRE param_pheno_indicateurs_ble.csv
A
D
C


A
D
C
LIRE
tas_day_1dCDFt_1985_2014_Ile_de_France.RData
AFFICHER "deb
Calcul Pheno
indicateurs"
reppheno/
pheno_ble_precoce_Ref.RData
LIRE Calcul_Pheno_Compromise.r
AFFICHER "fin Calcul
Pheno indicateurs"
Fin

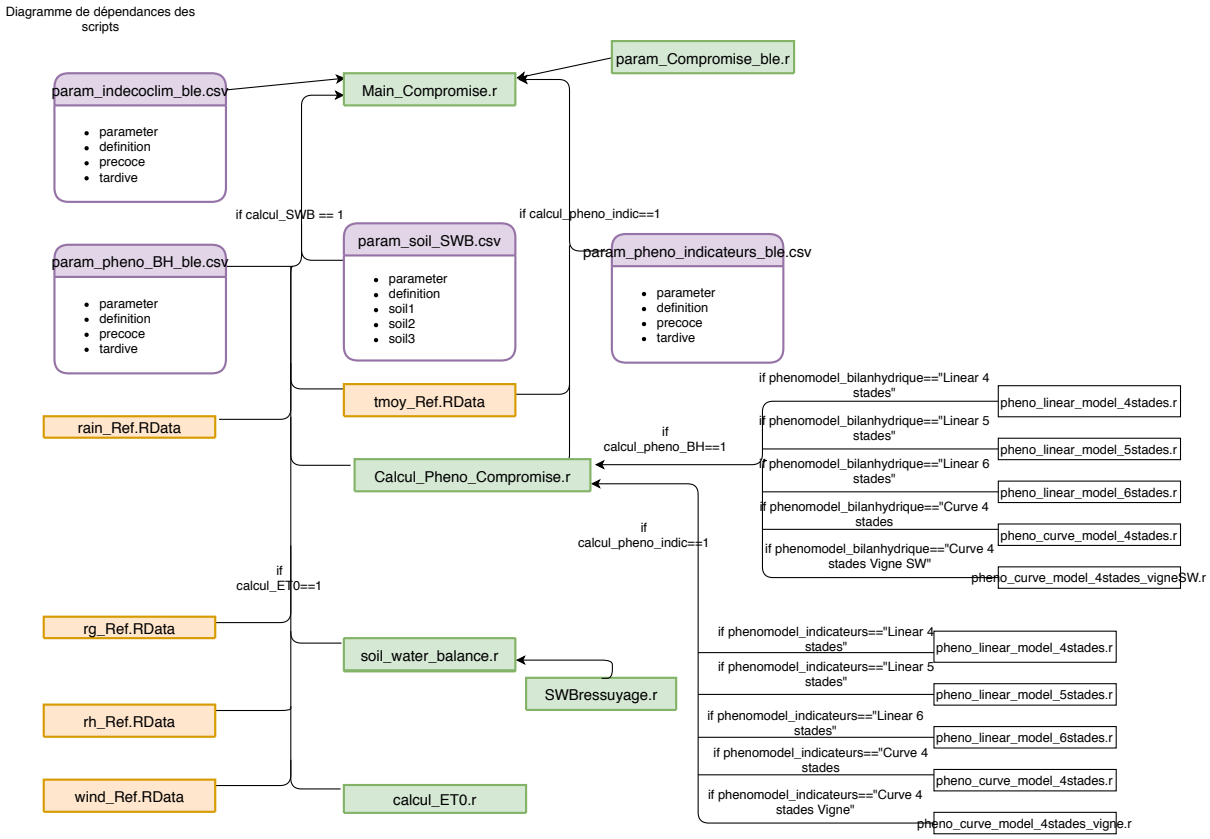


Diagramme de dépendances des scripts
param_Compromise_ble.r
param_indecoclim_ble.csv
parameter
definition
precoce
tardive
Main_Compromise.r
if calcul_SWB == 1
if calcul_pheno_indic==1
param_soil_SWB.csv
parameter
definition
soil1
soil2
soil3
param_pheno_indicateurs_ble.csv
parameter
definition
precoce
tardive
param_pheno_BH_ble.csv
parameter
definition
precoce
tardive
tmoy_Ref.RData
rain_Ref.RData
if phenomodel_bilanhydrique=="Linear 4 stades"
pheno_linear_model_4stades.r
if phenomodel_bilanhydrique=="Linear 5 stades"
pheno_linear_model_5stades.r
if phenomodel_bilanhydrique=="Linear 6 stades"
pheno_linear_model_6stades.r
if phenomodel_bilanhydrique=="Curve 4 stades
pheno_curve_model_4stades.r
if phenomodel_bilanhydrique=="Curve 4 stades Vigne SW"
pheno_curve_model_4stades_vigneSW.r
if calcul_pheno_BH==1
Calcul_Pheno_Compromise.r
if calcul_pheno_indic==1
if calcul_ET0==1
rg_Ref.RData
soil_water_balance.r
SWBressuyage.r
if phenomodel_indicateurs=="Linear 4 stades"
pheno_linear_model_4stades.r
if phenomodel_indicateurs=="Linear 5 stades"
pheno_linear_model_5stades.r
if phenomodel_indicateurs=="Linear 6 stades"
pheno_linear_model_6stades.r
if phenomodel_indicateurs=="Curve 4 stades
pheno_curve_model_4stades.r
if phenomodel_indicateurs=="Curve 4 stades Vigne"
pheno_curve_model_4stades_vigne.r
rh_Ref.RData
wind_Ref.RData
calcul_ET0.r

Script : prepare_dataclimat.r

```
##Bibliothèques
library(tibble)

##Sélection de la région
region="Ile_de_France"

##Définir les méthodes
correctionliste = c("1dCDFt","IntervardOTC","IntervarR2D2","SpatialdOTC",
                    "SpatialR2D2","SpatIntervardOTC","SpatIntervarR2D2")
#correctionliste = c("SAFRAN")

periode_passe_futur <- c("2036_2065") #définir les deux périodes passée et futur
#periode_passe_futur <- c("1985_2014")

##Définir les dimensions de la matrice
if(region=="Bretagne"){
grid_dim=c(21,20)
grid_pixel=grid_dim[1]*grid_dim[2]
}
if(region=="Ile_de_France"){
grid_dim=c(12,27)
grid_pixel=grid_dim[1]*grid_dim[2]
}
if(region=="Provence"){
grid_dim=c(24,20)
grid_pixel=grid_dim[1]*grid_dim[2]
}

##Définir le répertoire de travail initial
repinitial = c("C:/Users/Ling/Desktop/Stage INRAE/Dossier de travail")

##Définir le répertoire des corrections (répertoire climatique)
repclim  <- paste0(repinitial,"/MBC_",region,"/",region,"/","MBC")

#Création d'un répertoire pour les fichiers nettoyés
repcompromise <- paste0(repinitial,"/MBC_",region,"/",region,"/","DATACOMPROMISE")
dir.create(repcompromise, showWarnings=FALSE, recursive = TRUE)


####----Création des répertoires, nettoyages des variables, créations des nouveaux
#fichiers----####
for (correction in correctionliste){        #boucle sur les corrections
print(correction)

#####-----Créations de répetoires------####
```

```
#####-----lecture du fichier .RData avec toutes les variables climatiques------####


for (periode in periode_passe_futur){
print(periode)

if (correction != "SAFRAN"){
repcorrection <- paste0(repcompromise,"/","BC_data_day_",correction,"_",
                        periode,"_",region)
dir.create(repcorrection, showWarnings=FALSE, recursive = TRUE)

load(paste0(repclim,"/BC_data_day_",correction,"_1985_2014_2036_2065_",
            region,".RData"))
}

if (correction == "SAFRAN"){

repcorrection <- paste0(repcompromise,"/","Ref_data_day_",correction,"_",periode,
                        "_",region)
dir.create(repcorrection, showWarnings=FALSE, recursive = TRUE)

load(paste0(repinitial,"/MBC_",region,"/",region,"/SAFRAN/Ref_data_day_",correction,
            "_1985_2014_",region,".RData"))
}


if (periode == "1985_2014"){
#####-----Nettoyage des données-----------#####
##On charge chaque variable
##On insère dans une matrice de dimension "nbre de pixel" x "nbre de jour"
##On enlève les lignes non renseignées
##On rajoute le 32 décembre pour les années bissextiles
##On sauvegarde la nouvelle matrice

##température
tmoy <- eval(as.name(paste0("tas_day_",correction,"_",periode,"_",region)))
DATA_CLIMAT_tmoy <- matrix(c(tmoy[,,]),grid_pixel,10950)
DATA_CLIMAT_tmoy <- na.omit(DATA_CLIMAT_tmoy)
rm(tmoy)
DATA_CLIMAT_tmoy <- data.frame(DATA_CLIMAT_tmoy)
DATA_CLIMAT_tmoy <- add_column(DATA_CLIMAT_tmoy,X1460b=DATA_CLIMAT_tmoy$X1460,
                               .after = 1460)
DATA_CLIMAT_tmoy <- add_column(DATA_CLIMAT_tmoy,X2921b=DATA_CLIMAT_tmoy$X2921,
                               .after = 2921)
DATA_CLIMAT_tmoy <- add_column(DATA_CLIMAT_tmoy,X4382b=DATA_CLIMAT_tmoy$X4382,
                               .after = 4382)
DATA_CLIMAT_tmoy <- add_column(DATA_CLIMAT_tmoy,X5843b=DATA_CLIMAT_tmoy$X5843,
```

```
                                .after = 5843)
DATA_CLIMAT_tmoy <- add_column(DATA_CLIMAT_tmoy,X7304b=DATA_CLIMAT_tmoy$X7304,
                                .after = 7304)
DATA_CLIMAT_tmoy <- add_column(DATA_CLIMAT_tmoy,X8765b=DATA_CLIMAT_tmoy$X8765,
                                .after = 8765)
DATA_CLIMAT_tmoy <- add_column(DATA_CLIMAT_tmoy,X10226b=DATA_CLIMAT_tmoy$X10226,
                                .after =10226)
DATA_CLIMAT_tmoy <- data.matrix(DATA_CLIMAT_tmoy)
save(DATA_CLIMAT_tmoy,file = paste0(repcorrection,"/","tas_day_",correction,"_",
                                    periode, "_",region,".RData"))
print("DATA_CLIMAT_tmoy")

##pluie
rain <- eval(as.name(paste0("pr_day_",correction,"_",periode,"_",region)))
DATA_CLIMAT_rain <- matrix(c(rain[,,]),grid_pixel,10950)
DATA_CLIMAT_rain <- na.omit(DATA_CLIMAT_rain)
rm(rain)
DATA_CLIMAT_rain <- data.frame(DATA_CLIMAT_rain)
DATA_CLIMAT_rain <- add_column(DATA_CLIMAT_rain,X1460b=DATA_CLIMAT_rain$X1460,
                                .after = 1460)
DATA_CLIMAT_rain <- add_column(DATA_CLIMAT_rain,X2921b=DATA_CLIMAT_rain$X2921,
                                .after = 2921)
DATA_CLIMAT_rain <- add_column(DATA_CLIMAT_rain,X4382b=DATA_CLIMAT_rain$X4382,
                                .after = 4382)
DATA_CLIMAT_rain <- add_column(DATA_CLIMAT_rain,X5843b=DATA_CLIMAT_rain$X5843,
                                .after = 5843)
DATA_CLIMAT_rain <- add_column(DATA_CLIMAT_rain,X7304b=DATA_CLIMAT_rain$X7304,
                                .after = 7304)
DATA_CLIMAT_rain <- add_column(DATA_CLIMAT_rain,X8765b=DATA_CLIMAT_rain$X8765,
                                .after = 8765)
DATA_CLIMAT_rain <- add_column(DATA_CLIMAT_rain,X10226b=DATA_CLIMAT_rain$X10226,
                                .after = 10226)
DATA_CLIMAT_rain <- data.matrix(DATA_CLIMAT_rain)
save(DATA_CLIMAT_rain,file = paste0(repcorrection,"/","pr_day_",correction,"_",periode,
                                    "_",region,".RData"))
print("DATA_CLIMAT_rain")

##rayonnement
rg <- eval(as.name(paste0("rsds_day_",correction,"_",periode,"_",region)))
DATA_CLIMAT_rg<- matrix(c(rg[,,]),grid_pixel,10950)
DATA_CLIMAT_rg <- na.omit(DATA_CLIMAT_rg)
rm(rg)
DATA_CLIMAT_rg <- data.frame(DATA_CLIMAT_rg)
DATA_CLIMAT_rg <- add_column(DATA_CLIMAT_rg,X1460b=DATA_CLIMAT_rg$X1460,
                             .after = 1460)
DATA_CLIMAT_rg <- add_column(DATA_CLIMAT_rg,X2921b=DATA_CLIMAT_rg$X2921,
                             .after = 2921)
```

```
DATA_CLIMAT_rg <- add_column(DATA_CLIMAT_rg,X4382b=DATA_CLIMAT_rg$X4382,
                             .after = 4382)
DATA_CLIMAT_rg <- add_column(DATA_CLIMAT_rg,X5843b=DATA_CLIMAT_rg$X5843,
                             .after = 5843)
DATA_CLIMAT_rg <- add_column(DATA_CLIMAT_rg,X7304b=DATA_CLIMAT_rg$X7304,
                             .after = 7304)
DATA_CLIMAT_rg <- add_column(DATA_CLIMAT_rg,X8765b=DATA_CLIMAT_rg$X8765,
                             .after = 8765)
DATA_CLIMAT_rg <- add_column(DATA_CLIMAT_rg,X10226b=DATA_CLIMAT_rg$X10226,
                             .after = 10226)
DATA_CLIMAT_rg <- data.matrix(DATA_CLIMAT_rg)
save(DATA_CLIMAT_rg,file = paste0(repcorrection,"/","rsds_day_",correction,"_",
                                  periode,"_",region,".RData"))
print("DATA_CLIMAT_rg")

##vent
wind <- eval(as.name(paste0("sfcWind_day_",correction,"_",periode,"_",region)))
DATA_CLIMAT_wind <- matrix(c(wind[,,]),grid_pixel,10950)
DATA_CLIMAT_wind <- na.omit(DATA_CLIMAT_wind)
DATA_CLIMAT_wind <- data.frame(DATA_CLIMAT_wind)
DATA_CLIMAT_wind <- add_column(DATA_CLIMAT_wind,X1460b=DATA_CLIMAT_wind$X1460,
                               .after = 1460)
DATA_CLIMAT_wind <- add_column(DATA_CLIMAT_wind,X2921b=DATA_CLIMAT_wind$X2921,
                               .after = 2921)
DATA_CLIMAT_wind <- add_column(DATA_CLIMAT_wind,X4382b=DATA_CLIMAT_wind$X4382,
                               .after = 4382)
DATA_CLIMAT_wind <- add_column(DATA_CLIMAT_wind,X5843b=DATA_CLIMAT_wind$X5843,
                               .after = 5843)
DATA_CLIMAT_wind <- add_column(DATA_CLIMAT_wind,X7304b=DATA_CLIMAT_wind$X7304,
                               .after = 7304)
DATA_CLIMAT_wind <- add_column(DATA_CLIMAT_wind,X8765b=DATA_CLIMAT_wind$X8765,
                               .after = 8765)
DATA_CLIMAT_wind <- add_column(DATA_CLIMAT_wind,X10226b=DATA_CLIMAT_wind$X10226,
                               .after = 10226)
DATA_CLIMAT_wind <- data.matrix(DATA_CLIMAT_wind)
rm(wind)
save(DATA_CLIMAT_wind,file = paste0(repcorrection,"/","sfcWind_day_",correction,"_",
                                    periode,"_",region,".RData"))
print("DATA_CLIMAT_wind")

##humidité
rh <- eval(as.name(paste0("hurs_day_",correction,"_",periode,"_",region)))
DATA_CLIMAT_rh <- matrix(c(rh[,,]),grid_pixel,10950)
DATA_CLIMAT_rh <- na.omit(DATA_CLIMAT_rh)
rm(rh)
DATA_CLIMAT_rh <- data.frame(DATA_CLIMAT_rh)
DATA_CLIMAT_rh <- add_column(DATA_CLIMAT_rh,X1460b=DATA_CLIMAT_rh$X1460,.after = 1460)
```

```
DATA_CLIMAT_rh <- add_column(DATA_CLIMAT_rh,X2921b=DATA_CLIMAT_rh$X2921,.after = 2921)
DATA_CLIMAT_rh <- add_column(DATA_CLIMAT_rh,X4382b=DATA_CLIMAT_rh$X4382,.after = 4382)
DATA_CLIMAT_rh <- add_column(DATA_CLIMAT_rh,X5843b=DATA_CLIMAT_rh$X5843,.after = 5843)
DATA_CLIMAT_rh <- add_column(DATA_CLIMAT_rh,X7304b=DATA_CLIMAT_rh$X7304,.after = 7304)
DATA_CLIMAT_rh <- add_column(DATA_CLIMAT_rh,X8765b=DATA_CLIMAT_rh$X8765,.after = 8765)
DATA_CLIMAT_rh <- add_column(DATA_CLIMAT_rh,X10226b=DATA_CLIMAT_rh$X10226,
                             .after = 10226)
DATA_CLIMAT_rh <- data.matrix(DATA_CLIMAT_rh)
save(DATA_CLIMAT_rh,file = paste0(repcorrection,"/","hurs_day_",correction,"_",
                                  periode,"_",region,".RData"))
print("DATA_CLIMAT_rh")
}
else{

##température
tmoy <- eval(as.name(paste0("tas_day_",correction,"_",periode,"_",region)))
DATA_CLIMAT_tmoy <- matrix(c(tmoy[,,]),grid_pixel,10950)
DATA_CLIMAT_tmoy <- na.omit(DATA_CLIMAT_tmoy)
rm(tmoy)
DATA_CLIMAT_tmoy <- data.frame(DATA_CLIMAT_tmoy)
DATA_CLIMAT_tmoy <- add_column(DATA_CLIMAT_tmoy,X365b=DATA_CLIMAT_tmoy$X365,
                               .after = 365)
DATA_CLIMAT_tmoy <- add_column(DATA_CLIMAT_tmoy,X1826b=DATA_CLIMAT_tmoy$X1826,
                               .after = 1826)
DATA_CLIMAT_tmoy <- add_column(DATA_CLIMAT_tmoy,X3287b=DATA_CLIMAT_tmoy$X3287,
                               .after = 3287)
DATA_CLIMAT_tmoy <- add_column(DATA_CLIMAT_tmoy,X4748b=DATA_CLIMAT_tmoy$X4748,
                               .after = 4748)
DATA_CLIMAT_tmoy <- add_column(DATA_CLIMAT_tmoy,X6209b=DATA_CLIMAT_tmoy$X6209,
                               .after = 6209)
DATA_CLIMAT_tmoy <- add_column(DATA_CLIMAT_tmoy,X7670b=DATA_CLIMAT_tmoy$X7670,
                               .after = 7670)
DATA_CLIMAT_tmoy <- add_column(DATA_CLIMAT_tmoy,X9131b=DATA_CLIMAT_tmoy$X9131,
                               .after = 9131)
DATA_CLIMAT_tmoy <- add_column(DATA_CLIMAT_tmoy,X10592b=DATA_CLIMAT_tmoy$X10592,
                               .after = 10592)
DATA_CLIMAT_tmoy <- data.matrix(DATA_CLIMAT_tmoy)
save(DATA_CLIMAT_tmoy,file = paste0(repcorrection,"/","tas_day_",correction,"_",
                                    periode,"_",region,".RData"))
print("DATA_CLIMAT_tmoy")

##pluie
rain <- eval(as.name(paste0("pr_day_",correction,"_",periode,"_",region)))
DATA_CLIMAT_rain <- matrix(c(rain[,,]),grid_pixel,10950)
DATA_CLIMAT_rain <- na.omit(DATA_CLIMAT_rain)
rm(rain)
DATA_CLIMAT_rain <- data.frame(DATA_CLIMAT_rain)
```

```
DATA_CLIMAT_rain <- add_column(DATA_CLIMAT_rain,X365b=DATA_CLIMAT_rain$X365,
                               .after = 365)
DATA_CLIMAT_rain <- add_column(DATA_CLIMAT_rain,X1826b=DATA_CLIMAT_rain$X1826,
                               .after = 1826)
DATA_CLIMAT_rain <- add_column(DATA_CLIMAT_rain,X3287b=DATA_CLIMAT_rain$X3287,
                               .after = 3287)
DATA_CLIMAT_rain <- add_column(DATA_CLIMAT_rain,X4748b=DATA_CLIMAT_rain$X4748,
                               .after = 4748)
DATA_CLIMAT_rain <- add_column(DATA_CLIMAT_rain,X6209b=DATA_CLIMAT_rain$X6209,
                               .after = 6209)
DATA_CLIMAT_rain <- add_column(DATA_CLIMAT_rain,X7670b=DATA_CLIMAT_rain$X7670,
                               .after = 7670)
DATA_CLIMAT_rain <- add_column(DATA_CLIMAT_rain,X9131b=DATA_CLIMAT_rain$X9131,
                               .after = 9131)
DATA_CLIMAT_rain <- add_column(DATA_CLIMAT_rain,X10592b=DATA_CLIMAT_rain$X10592,
                               .after = 10592)
DATA_CLIMAT_rain <- data.matrix(DATA_CLIMAT_rain)
save(DATA_CLIMAT_rain,file = paste0(repcorrection,"/","pr_day_",correction,"_",periode,
                                    "_",region,".RData"))
print("DATA_CLIMAT_rain")

##rayonnement
rg <- eval(as.name(paste0("rsds_day_",correction,"_",periode,"_",region)))
DATA_CLIMAT_rg<- matrix(c(rg[,,]),grid_pixel,10950)
DATA_CLIMAT_rg <- na.omit(DATA_CLIMAT_rg)
rm(rg)
DATA_CLIMAT_rg <- data.frame(DATA_CLIMAT_rg)
DATA_CLIMAT_rg <- add_column(DATA_CLIMAT_rg,X365b=DATA_CLIMAT_rg$X365,.after = 365)
DATA_CLIMAT_rg <- add_column(DATA_CLIMAT_rg,X1826b=DATA_CLIMAT_rg$X1826,.after = 1826)
DATA_CLIMAT_rg <- add_column(DATA_CLIMAT_rg,X3287b=DATA_CLIMAT_rg$X3287,.after = 3287)
DATA_CLIMAT_rg <- add_column(DATA_CLIMAT_rg,X4748b=DATA_CLIMAT_rg$X4748,.after = 4748)
DATA_CLIMAT_rg <- add_column(DATA_CLIMAT_rg,X6209b=DATA_CLIMAT_rg$X6209,.after = 6209)
DATA_CLIMAT_rg <- add_column(DATA_CLIMAT_rg,X7670b=DATA_CLIMAT_rg$X7670,.after = 7670)
DATA_CLIMAT_rg <- add_column(DATA_CLIMAT_rg,X9131b=DATA_CLIMAT_rg$X9131,.after = 9131)
DATA_CLIMAT_rg <- add_column(DATA_CLIMAT_rg,X10592b=DATA_CLIMAT_rg$X10592,
                             .after = 10592)
DATA_CLIMAT_rg <- data.matrix(DATA_CLIMAT_rg)
save(DATA_CLIMAT_rg,file = paste0(repcorrection,"/","rsds_day_",correction,"_",
                                  periode,"_",region,".RData"))
print("DATA_CLIMAT_rg")

##vent
wind <- eval(as.name(paste0("sfcWind_day_",correction,"_",periode,"_",region)))
DATA_CLIMAT_wind <- matrix(c(wind[,,]),grid_pixel,10950)
DATA_CLIMAT_wind <- na.omit(DATA_CLIMAT_wind)
DATA_CLIMAT_wind <- data.frame(DATA_CLIMAT_wind)
DATA_CLIMAT_wind <- add_column(DATA_CLIMAT_wind,X365b=DATA_CLIMAT_wind$X365,
```

```
                                .after = 365)
DATA_CLIMAT_wind <- add_column(DATA_CLIMAT_wind,X1826b=DATA_CLIMAT_wind$X1826,
                                .after = 1826)
DATA_CLIMAT_wind <- add_column(DATA_CLIMAT_wind,X3287b=DATA_CLIMAT_wind$X3287,
                                .after = 3287)
DATA_CLIMAT_wind <- add_column(DATA_CLIMAT_wind,X4748b=DATA_CLIMAT_wind$X4748,
                                .after = 4748)
DATA_CLIMAT_wind <- add_column(DATA_CLIMAT_wind,X6209b=DATA_CLIMAT_wind$X6209,
                                .after = 6209)
DATA_CLIMAT_wind <- add_column(DATA_CLIMAT_wind,X7670b=DATA_CLIMAT_wind$X7670,
                                .after = 7670)
DATA_CLIMAT_wind <- add_column(DATA_CLIMAT_wind,X9131b=DATA_CLIMAT_wind$X9131,
                                .after = 9131)
DATA_CLIMAT_wind <- add_column(DATA_CLIMAT_wind,X10592b=DATA_CLIMAT_wind$X10592,
                                .after =10592)
DATA_CLIMAT_wind <- data.matrix(DATA_CLIMAT_wind)
rm(wind)
save(DATA_CLIMAT_wind,file = paste0(repcorrection,"/","sfcWind_day_",correction,"_",
                                    periode,"_",region,".RData"))
print("DATA_CLIMAT_wind")

##humidité
rh <- eval(as.name(paste0("hurs_day_",correction,"_",periode,"_",region)))
DATA_CLIMAT_rh <- matrix(c(rh[,,]),grid_pixel,10950)
DATA_CLIMAT_rh <- na.omit(DATA_CLIMAT_rh)
rm(rh)
DATA_CLIMAT_rh <- data.frame(DATA_CLIMAT_rh)
DATA_CLIMAT_rh <- add_column(DATA_CLIMAT_rh,X365b=DATA_CLIMAT_rh$X365,.after = 365)
DATA_CLIMAT_rh <- add_column(DATA_CLIMAT_rh,X1826b=DATA_CLIMAT_rh$X1826,.after = 1826)
DATA_CLIMAT_rh <- add_column(DATA_CLIMAT_rh,X3287b=DATA_CLIMAT_rh$X3287,.after = 3287)
DATA_CLIMAT_rh <- add_column(DATA_CLIMAT_rh,X4748b=DATA_CLIMAT_rh$X4748,.after = 4748)
DATA_CLIMAT_rh <- add_column(DATA_CLIMAT_rh,X6209b=DATA_CLIMAT_rh$X6209,.after = 6209)
DATA_CLIMAT_rh <- add_column(DATA_CLIMAT_rh,X7670b=DATA_CLIMAT_rh$X7670,.after = 7670)
DATA_CLIMAT_rh <- add_column(DATA_CLIMAT_rh,X9131b=DATA_CLIMAT_rh$X9131,
                             .after = 9131)
DATA_CLIMAT_rh <- add_column(DATA_CLIMAT_rh,X10592b=DATA_CLIMAT_rh$X10592,
                             .after = 10592)
DATA_CLIMAT_rh <- data.matrix(DATA_CLIMAT_rh)
save(DATA_CLIMAT_rh,file = paste0(repcorrection,"/","hurs_day_",correction,"_",
                                  periode,"_",region,".RData"))
print("DATA_CLIMAT_rh")

} #else

} #periode_passe_futur
} #correctionliste
```

Script : prepare_coord_SAFRAN.r

```
#####---- Script pour déterminer le numéro des lignes ayant des valeurs manquantes
##---- et enlever les coordonnées de ces lignes

#Définir le répertoire de travail
repinitial = c("C:/Users/Ling/Desktop/Stage INRAE/Dossier de travail")

##Définir le répertoire des corrections (répertoire climatique)
repclim  <- paste0(repinitial,"/MBC_",region,"/",region,"/","MBC")

##Définir correction, periode, region
correction="1dCDFt"
periode="1985_2014"
region="Provence"

##Définir les dimensions de la matrice
if(region=="Bretagne"){
grid_dim=c(21,20)
grid_pixel=grid_dim[1]*grid_dim[2]
}
if(region=="Ile_de_France"){
grid_dim=c(12,27)
grid_pixel=grid_dim[1]*grid_dim[2]
}
if(region=="Provence"){
grid_dim=c(24,20)
grid_pixel=grid_dim[1]*grid_dim[2]
}

##Charger le fichier dans l'environnement de travail
load(paste0(repclim,"/BC_data_day_",correction,"_1985_2014_2036_2065_",region,
            ".RData"))

##Charger la variable
tmoy <- eval(as.name(paste0("tas_day_",correction,"_",periode,"_",region)))

#Définir comme une matrice pour avoir les dimensions
DATA_CLIMAT_tmoy <- matrix(c(tmoy[,,]),grid_pixel,10950)

#transformer en dataframe pour les manipulations
DATA_CLIMAT_tmoy <- as.data.frame(DATA_CLIMAT_tmoy)

#Mettre l'indice des lignes avec NaN dans rows_NaN
rows_NaN <- DATA_CLIMAT_tmoy[apply(DATA_CLIMAT_tmoy[1:ncol(DATA_CLIMAT_tmoy)],
                                   1, function(x) all(is.nan(x))),]
```

```
#Déclarer comme numérique
num_rows_NaN <- as.numeric(rownames(rows_NaN))
print(num_rows_NaN) #numéro des lignes avec des NaN

###Charger le fichier coord_SAFRAN de la région concernée
coord <- read.table(paste0(repinitial,"/MBC_",region,"/",region,"/",
                           "coord_SAFRAN_",region,".txt"),header = T,sep="\t")
print(dim(coord))

new_coord <- coord[-num_rows_NaN,]
print(dim(new_coord))

#Enregistrer le fichier en txt
#Il faut enregistrer le fichier dans le répertoire MBC
write.table(new_coord,file=paste0(repclim,"/","coord_SAFRAN.txt"),sep="\t",
            col.names = T,row.names = F)
```

Script : fonction_dated_colnames_1985_2014.r

```
##Bibliothèques
library(dplyr)
library(tibble)

##Créer un dataframe avec toutes les dates de la période
myDates=data.frame(seq(as.Date("1985-01-01"), to=as.Date("2014-12-31"),by="days"))
names(myDates)= "Dates"
nrow(myDates) #10957

##Enlever les 29 février
myDates <- myDates[!(format(myDates$Dates,"%m") == "02" &
                       format(myDates$Dates, "%d") == "29"), ,drop = FALSE]
nrow(myDates) #10950

##Numéro des lignes des 31 décembre ##Check
# which((format(myDates$Dates,"%m") == "12" & format(myDates$Dates, "%d") == "31"))

##Convertir les dates en caractères
myDates <- myDates %>%
mutate_all(as.character)

#Rajouter les 32 décembre aux années bissextile
myDates <- add_row(myDates,Dates="1988-12-32",.after = 1460)
myDates <- add_row(myDates,Dates="1992-12-32",.after = 2921)
myDates <- add_row(myDates,Dates="1996-12-32",.after = 4382)
myDates <- add_row(myDates,Dates="2000-12-32",.after = 5843)
myDates <- add_row(myDates,Dates="2004-12-32",.after = 7304)
myDates <- add_row(myDates,Dates="2008-12-32",.after = 8765)
myDates <- add_row(myDates,Dates="2012-12-32",.after = 10226)

#Sauver dans un vecteur
nom_colonnes_datees <- myDates$Dates
```

Script : fonction_dated_colnames_2036_2065.r

```
##Bibliothèques
library(dplyr)
library(tibble)

##Créer un dataframe avec toutes les dates de la période
myDates=data.frame(seq(as.Date("2036-01-01"), to=as.Date("2065-12-31"),by="days"))
names(myDates)= "Dates"
nrow(myDates) #10958

##Enlever les 29 février
myDates <- myDates[!(format(myDates$Dates,"%m") == "02" &
                       format(myDates$Dates, "%d") == "29"), ,drop = FALSE]
nrow(myDates) #10950

##Numéro des lignes des 31 décembre ##Check
# which((format(myDates$Dates,"%m") == "12" & format(myDates$Dates, "%d") == "31"))

##Convertir les dates en caractères
myDates <- myDates %>%
mutate_all(as.character)

#Rajouter les 32 décembre aux années bissextile
myDates <- add_row(myDates,Dates="2036-12-32",.after = 365)
myDates <- add_row(myDates,Dates="2040-12-32",.after = 1826)
myDates <- add_row(myDates,Dates="2044-12-32",.after = 3287)
myDates <- add_row(myDates,Dates="2048-12-32",.after = 4748)
myDates <- add_row(myDates,Dates="2052-12-32",.after = 6209)
myDates <- add_row(myDates,Dates="2056-12-32",.after = 7670)
myDates <- add_row(myDates,Dates="2060-12-32",.after = 9131)
myDates <- add_row(myDates,Dates="2064-12-32",.after = 10592)

#Sauver dans un vecteur
nom_colonnes_datees <- myDates$Dates
```

Script : Boxplot_phenologie.r

```
#########################################################################
###Script: BOXPLOT PHENO
##Projet Compromise
##10/06/2020
#########################################################################

#Définir le répertoire initiale
#A modifier selon votre répertoire
repinitial <- c("C:/Users/Ling/Desktop/Stage INRAE/Dossier de travail")

#Définir le répertoire de travail
reptravail <- paste(repinitial,"/Code/", sep="")
setwd(reptravail)

#Définir les arguments   #Formats différents ne sont pas acceptés
# region="Bretagne"               ## "Provence" ou "Ile_de_France" ou "Bretagne"
# culture="ble"                   ## "mais" ou "vigne" ou "ble"
# periode="1985_2014"             ## "2036_2065" ou "1985_2014"
# indicateur="S3"                  ## S1 S2 S3 S4
# annee_debut=1986
# annee_fin=1996
# selection_correction=c("1dCDFt","IntervardOTC")

source("fonction_boxplot_phenologie.R")
par(mar = c(5.1, 4.1, 4.1, 2.1))
boxplot.indicateur("Provence","vigne","2036_2065","S3",2052,2065)
```

Script : fonction_boxplot_phenologie.R

```
####Définir le nbre de correction
correctionliste <<- c("1dCDFt","IntervardOTC","IntervarR2D2","SpatialdOTC",
                      "SpatialR2D2","SpatIntervardOTC","SpatIntervarR2D2")

####Fonction pour definir un dataframe pour le boxplot
definir.boxplot.df <- function(region,culture,periode,indicateur){ #remplir.df

datalist <<- list()

reppheno <- paste0(reptravail,"Compromise/",region,"/",culture,"/","PHENO")

for (correction in correctionliste){

if(culture!="vigne"){
load(paste0(reppheno,"/","pheno_",culture,"_precoce_",correction,"_",periode,".RData"))
}else{
load(paste0(reppheno,"/","pheno_",culture,"_chardonnay_",correction,"_",periode,
            ".RData"))
}

datapheno_df <<- data.frame(DATA_PHENO)

dat <<- data.frame(datapheno_df$year,datapheno_df[,indicateur],correction)
dat$correction <<- correction
datalist[[correction]] <<- dat

} #correctionliste

new_datapheno_df <<- do.call(rbind,datalist)
colnames(new_datapheno_df) <<- c("annee","valeur_indicateur","correction_namelist")
rownames(new_datapheno_df) <<- NULL

}

####Fonction qui fait les boxplots selon les arguments
boxplot.indicateur <- function(region,culture,periode,indicateur,annee_debut,
                               annee_fin,
                               selection_correction = c("1dCDFt","IntervardOTC",
                                                        "IntervarR2D2","SpatialdOTC",

options(error=NULL)

if(culture!="mais" & periode == "1985_2014" &
   as.integer(substr(periode,1,4))>=annee_debut){
stop("Erreur: L'année de début doit être à partir de 1986")
```

```
}

if(culture!="mais" & periode == "2036_2065" &
   as.integer(substr(periode,1,4))>=annee_debut){
stop("Erreur: L'année de début doit être à partir de 2036")
}

if(periode != "1985_2014" & periode != "2036_2065"){
stop("Erreur: l'argument periode n'est pas correcte")
}
if(periode == "1985_2014" & ( as.integer(substr(periode,1,4))>annee_debut |
                              as.integer(substr(periode,6,9))<annee_fin)){
stop("Erreur: annee_debut ou annee_fin pas dans la limite de periode")
}
if(periode == "2036_2065" & ( as.integer(substr(periode,1,4))>annee_debut |
                              as.integer(substr(periode,6,9))<annee_fin)){
stop("Erreur: annee_debut ou annee_fin pas dans la limite de periode")
}

definir.boxplot.df(region,culture,periode,indicateur)

#Valeur max pheno sur 30 ans
max_pheno <- max(na.omit(new_datapheno_df$valeur_indicateur))

#Valeur min pheno sur 30 ans
min_pheno <- min(na.omit(new_datapheno_df$valeur_indicateur))

#Dataframe filtrer selon arguments
new_datapheno_arg_df <<- new_datapheno_df[new_datapheno_df$annee>=annee_debut &
                                          new_datapheno_df$annee<=annee_fin &
                                          new_datapheno_df$correction_namelist  %in%
                                          selection_correction, ]

#Definir la légende
legend_text <- sort(unique(new_datapheno_arg_df$correction_namelist))

#Définir les corrections comme des facteurs
new_datapheno_arg_df$correction_namelist <- as.factor(
  new_datapheno_arg_df$correction_namelist)

#Définir les couleurs selon la correction
mycolors <-ifelse(levels(new_datapheno_arg_df$correction_namelist)=="SAFRAN",
                  #FFCC33",
ifelse(levels(new_datapheno_arg_df$correction_namelist)=="1dCDFt","#99FFFF",
ifelse(levels(new_datapheno_arg_df$correction_namelist)=="IntervardOTC","#BFFF80",
ifelse(levels(new_datapheno_arg_df$correction_namelist)=="IntervarR2D2","#FFDF80",
ifelse(levels(new_datapheno_arg_df$correction_namelist)=="SpatialdOTC","#CC3366",
```

```
ifelse(levels(new_datapheno_arg_df$correction_namelist)=="SpatialR2D2","#BF80FF",
ifelse(levels(new_datapheno_arg_df$correction_namelist)=="SpatIntervardOTC","#66FFCC",
ifelse(levels(new_datapheno_arg_df$correction_namelist)=="SpatIntervarR2D2","#FF6633",
       "white")))))))

#Défini un espace à droite pour la légende
par(mar = c(5.1, 4.1, 4.1, 7.5),xpd = TRUE)

#Boxplot
boxplot(new_datapheno_arg_df$valeur_indicateur~
          new_datapheno_arg_df$correction_namelist+
          new_datapheno_arg_df$annee, col=mycolors, names =
          c(rep(c(annee_debut:annee_fin), each=length(selection_correction))),
xlab=paste(annee_debut,"à",annee_fin,sep=" "), ylab=paste(indicateur,"PHENO",sep=" "),
main=paste(region,"-",culture,"-",periode),ylim=c(min_pheno,max_pheno))
legend("topright", inset = c(-0.38, 0), fill = mycolors,
legend = c(legend_text), horiz = F,cex=0.65,bty="n")
}
```

Script : Boxplot_ET0_BH.r

```
#########################################################################
###Script: BOXPLOT ETO BH
##Projet Compromise
##17/06/2020
#########################################################################
#------NETTOYAGE ENVIRONNEMENT TRAVAIL--------
rm(list=ls())

#Définir le répertoire initiale
#A modifier selon votre répertoire
repinitial <- c("C:/Users/Ling/Desktop/Stage INRAE/Dossier de travail")

#Définir le répertoire de travail
reptravail <- paste(repinitial,"/Code/", sep="")
setwd(reptravail)

#Définir les arguments   #Formats différents ne sont pas acceptés
# x                      ## "ET0" ou "SWC" ou "WATER_RESERVE"
# region                 ## "Provence" ou "Ile_de_France" ou "Bretagne"
# culture                ## "mais" ou "vigne" ou "ble"
# periode                ## "2036_2065" ou "1985_2014"
# mois                   ## 01,02,03,...,12
#annee_debut             ## une année de début dans la période définie
#annee_fin               ##une année de fin dans la période définie
# statistique            ##mean median max min sd
#selection_correction    ##c("1dCDFt", "IntervardOTC", "IntervarR2D2",
#                           "SpatialdOTC", "SpatialR2D2", "SpatIntervardOTC",
#                            "SpatIntervarR2D2")

source("fonction_boxplot_BH_ET0.R")
boxplot.indicateur("WATER_RESERVE","Provence","vigne","1985_2014",07,1986,1996,mean)
```

Script : fonction_boxplot_BH_ET0.r

```
#############################################################################
##Définir la liste de correction
#############################################################################
correctionliste <<- c("1dCDFt","IntervardOTC","IntervarR2D2","SpatialdOTC",
                      "SpatialR2D2","SpatIntervardOTC","SpatIntervarR2D2")

#############################################################################
##Fonction pour définir le répertoire des données
#############################################################################
definir.repindicateur <- function(x,region,culture,periode){

if(x=="ET0"){repindicateur <<- paste0(reptravail,"Compromise/",region,"/",culture,"/",
"ET0")}
if(x=="WATER_RESERVE" | x=="SWC"){ repindicateur <<- paste0(reptravail,"Compromise/",
region, "/",culture,"/","BH")}
}

#definir.repindicateur(x,region,culture,periode)

#############################################################################
##Fonction pour charger les données dans l'environnement de travail
#############################################################################

charger.donnees <- function(x,region,culture,periode){ #definir.repindicateur()

for(correction in correctionliste){

##Charger les données dans l'environnement de travail
if(x=="ET0"){
load(paste0(repindicateur,"/","ET0","_",correction,"_",periode,".RData"))
indicateur_df <- data.frame(DATA_CLIMAT_ET0)
}
if(x=="WATER_RESERVE"){
if(culture=="mais" | culture =="ble"){
load(paste0(repindicateur,"/","reserve_",culture,"_precoce_",correction,"_",
            periode,
"_soil1",".RData"))
}else{
load(paste0(repindicateur,"/","reserve_",culture,"_chardonnay_",correction,"_",
            periode,
"_soil1",".RData"))
}
indicateur_df <- data.frame(WATER_RESERVE)
}
if(x=="SWC"){
```

```
if(culture=="mais" | culture =="ble"){
load(paste0(repindicateur,"/","swc_",culture,"_precoce_",correction,"_",
            periode,
"_soil1",".RData"))
}else{
load(paste0(repindicateur,"/","swc_",culture,"_chardonnay_",correction,"_",
            periode,
"_soil1",".RData"))
}
indicateur_df <<- data.frame(SOIL_WATER_CONTENT)
}

##Création des noms de colonnes datées
if (periode=="1985_2014"){source("fonction_dated_colnames_1985_2014.R")}

if (periode=="2036_2065"){source("fonction_dated_colnames_2036_2065.R")}

##Nommer les colonnes avec les dates
##Si c'est le BH pour la vigne on a 29 ans seulement
if(x!="ET0" && culture=="vigne"){
if (periode=="1985_2014"){colnames(indicateur_df) <- nom_colonnes_datees[366:10957]}
if (periode=="2036_2065"){colnames(indicateur_df) <- nom_colonnes_datees[367:10958]}
}else{colnames(indicateur_df) <- nom_colonnes_datees}

assign(paste0("indicateur_df_",correction),indicateur_df,envir = globalenv())

}
}
#charger.donnees(x,region,culture,periode)

##############################################################################
##Fonction pour calculer l'indicateur
##############################################################################

calcul.stat.indicateur<- function(periode,mois,statistique){

#définir les années en fonction de la période
if (periode=="1985_2014"){debut_periode = 1986;fin_periode = 2014}

if (periode=="2036_2065"){debut_periode = 2037;fin_periode = 2065}

datalist <<- list()
datalist2 <<- list()

for (correction in correctionliste){

indicateur_df <- eval(as.name(paste0("indicateur_df_",correction)))
```

```
for (annee in debut_periode:fin_periode){
filtre_annee <<- substr(colnames(indicateur_df),1,4)==annee &
as.integer(substr(colnames(indicateur_df),6,7))==mois
indicateur_df_mois <<- indicateur_df[,filtre_annee]

stat_indicateur <<- apply(indicateur_df_mois,1,statistique)

df_temp <<- data.frame(annee,stat_indicateur)

df_temp$annee <<- annee

datalist[[annee]] <<- df_temp
}

annee_vs_indicateur_correction = do.call(rbind, datalist)

df_temp2 <<- annee_vs_indicateur_correction
dat <<- data.frame(df_temp2,correction)
dat$correction <<- correction
datalist2[[correction]] <<- dat
} #correctionliste

annee_vs_indicateur_all_years <<- do.call(rbind,datalist2)

rownames(annee_vs_indicateur_all_years) <<- NULL
colnames(annee_vs_indicateur_all_years) <<-c("annee", "stat_indicateur",
"correction_namelist")
}

#calcul.stat.indicateur(mois,statistique)

##############################################################################
##Fonction pour faire le boxplot
##############################################################################
boxplot.indicateur <- function(x,region,culture,periode,mois,annee_debut,annee_fin,
statistique=mean,
selection_correction=c("1dCDFt", "IntervardOTC",
"IntervarR2D2", "SpatialdOTC",
"SpatialR2D2", "SpatIntervardOTC",
"SpatIntervarR2D2")){

options(error=NULL)

if(periode != "1985_2014" & periode != "2036_2065"){
stop("Erreur: l'argument periode n'est pas correcte")
}
```

```
if(periode == "1985_2014" & (as.integer(substr(periode,1,4))>annee_debut |
as.integer(substr(periode,6,9))<annee_fin)){
stop("Erreur: annee_debut ou annee_fin pas dans la limite de periode")
}
if(periode == "2036_2065" & (as.integer(substr(periode,1,4))>annee_debut |
as.integer(substr(periode,6,9))<annee_fin)){
stop("Erreur: annee_debut ou annee_fin pas dans la limite de periode")
}

definir.repindicateur(x,region,culture,periode)
charger.donnees(x,region,culture,periode)
calcul.stat.indicateur(periode,mois,statistique)

#valeur max de l'indicateur sur 30 ans
max_ind <- max(annee_vs_indicateur_all_years$stat_indicateur)

#valeur min de l'indicateur sur 30 ans
min_ind <- min(annee_vs_indicateur_all_years$stat_indicateur)

#nouveau dataframe filtré en fonction des arguments
new_annee_vs_indicateur_arg <<-
  annee_vs_indicateur_all_years[annee_vs_indicateur_all_years$annee>=annee_debut &
annee_vs_indicateur_all_years$annee<=annee_fin &
annee_vs_indicateur_all_years$correction_namelist  %in% selection_correction, ]

#Definir la légende
legend_text <- sort(unique(new_annee_vs_indicateur_arg$correction_namelist))

#Définir les corrections comme des facteurs
new_annee_vs_indicateur_arg$correction_namelist<-as.factor(
  new_annee_vs_indicateur_arg$correction_namelist)

#Définir les couleurs selon la correction
mycolors<-ifelse(levels(new_annee_vs_indicateur_arg$correction_namelist)=="SAFRAN",
                 "#FFCC33",
ifelse(levels(new_annee_vs_indicateur_arg$correction_namelist)=="1dCDFt","#99FFFF",
ifelse(levels(new_annee_vs_indicateur_arg$correction_namelist)=="IntervardOTC",
       "#BFFF80",
ifelse(levels(new_annee_vs_indicateur_arg$correction_namelist)=="IntervarR2D2",
       "#FFDF80",
ifelse(levels(new_annee_vs_indicateur_arg$correction_namelist)=="SpatialdOTC",
       "#CC3366",
ifelse(levels(new_annee_vs_indicateur_arg$correction_namelist)=="SpatialR2D2",
       "#BF80FF",
ifelse(levels(new_annee_vs_indicateur_arg$correction_namelist)=="SpatIntervardOTC",
       "#66FFCC",
ifelse(levels(new_annee_vs_indicateur_arg$correction_namelist)=="SpatIntervarR2D2",
```

```
       "#FF6633","white")))))))


nom_indicateur <- ifelse(x=="ET0","Evapotranspiration",
ifelse(x=="SWC","Humidité du sol",
ifelse(x=="WATER_RESERVE","Réserve d'eau","")))

nom_mois <- ifelse(mois==1,"janvier",
ifelse(mois==2,"février",
ifelse(mois==3,"mars",
ifelse(mois==4,"avril",
ifelse(mois==5,"mai",
ifelse(mois==6,"juin",
ifelse(mois==7,"juillet",
ifelse(mois==8,"août",
ifelse(mois==9,"septembre",
ifelse(mois==10,"octobre",
ifelse(mois==11,"novembre",
ifelse(mois==11,"décembre","")))))))))))
statistique_text <- ifelse(identical(statistique,mean)==TRUE,"Moyenne",
ifelse(identical(statistique,median)==TRUE,"Médiane",
ifelse(identical(statistique,sd)==TRUE,"Ecart-type",
ifelse(identical(statistique,var)==TRUE,"Variance",
ifelse(identical(statistique,max)==TRUE,"Maximum",
ifelse(identical(statistique,min)==TRUE,"Minimum",""
))))))

#Si on veut la légende en dehors du plot
# par(mar = c(5.1, 4.1, 4.1, 7.5),xpd = TRUE)

#Boxplot
boxplot(new_annee_vs_indicateur_arg$stat_indicateur~
          new_annee_vs_indicateur_arg$correction_namelist +
          new_annee_vs_indicateur_arg$annee,
col=mycolors, names = c(rep(c(annee_debut:annee_fin),
each=length(selection_correction))),
xlab=paste(annee_debut,"à",annee_fin,sep=" "),
ylab=paste(nom_indicateur,"en",nom_mois,"(",statistique_text,")",sep=" "),
main=paste(region,"-",culture,"-",periode),ylim=c(min_ind,max_ind))

legend("topright", fill = mycolors,
legend = c(legend_text), horiz = F,cex=0.65,bty="n")

#Activer ligne 187 et ensuite 201-202
# legend("topright", inset = c(-0.34, 0), fill = mycolors,
#        legend = c(legend_text), horiz = F,cex=0.65,bty="n")
}
```

Script : Cartographie_phenologie.r

```
#########################################################################
###Script: CARTOGRAPHIE PHENOLOGIE
##Projet Compromise
##06/06/2020
#########################################################################

#Définir le répertoire initiale
#A modifier selon votre répertoire
repinitial <- c("C:/Users/Ling/Desktop/Stage INRAE/Dossier de travail")

#Définir le répertoire de travail
reptravail <- paste(repinitial,"/Code/", sep="")
setwd(reptravail)

#Définir les arguments   #Formats différents ne sont pas acceptés
# region="Bretagne"      ## "Provence" ou "Ile_de_France" ou "Bretagne"
# culture="ble"          ## "mais" ou "vigne" ou "ble"
# periode="1985_2014"    ## "2036_2065" ou "1985_2014"
# indicateur="S3"        ## S1 S2 S3 S4
# statistique = mean     ##mean median max min sd

source("fonction_cartographie_phenologie.R")
par(mfrow=c(2,4),mar=c(5.1, 3.7, 5.1, 4))
cartographie.indicateur("Provence","vigne","2036_2065","S3",mean)
```

Script R : fonction_cartographie_phenologie.r

```
#########################################################################
###Script: fonction CARTOGRAPHIE
##Projet Compromise
##06/06/2020
#########################################################################

#############################################################################
##Fonction pour définir le nbre de corrections
#############################################################################
definir.nb.correction <- function(periode){

if (periode=="1985_2014"){
correctionliste <<- c("SAFRAN","1dCDFt","IntervardOTC","IntervarR2D2","SpatialdOTC",
"SpatialR2D2","SpatIntervardOTC","SpatIntervarR2D2")
}

if (periode=="2036_2065"){
correctionliste <<- c("1dCDFt","IntervardOTC","IntervarR2D2","SpatialdOTC",
"SpatialR2D2","SpatIntervardOTC","SpatIntervarR2D2")
}
}
#############################################################################
##Fonction pour définir le nbre de sites
#############################################################################
definir.nb.sites <- function(region){

if(region=="Bretagne"){
nb_sites <<- 259
}
if(region=="Ile_de_France"){
nb_sites <<- 319
}
if(region=="Provence"){
nb_sites <<- 337
}

return(nb_sites)
}
#############################################################################
##Fonction pour charger les données dans l'environnement de travail
#############################################################################
charger.donnees <- function(region,culture,periode){

reppheno <- paste0(reptravail,"Compromise/",region,"/",culture,"/","PHENO")
```

```
for (correction in correctionliste){

##Charger le fichier
if(culture!="vigne"){
load(paste0(reppheno,"/","pheno_",culture,"_precoce_",correction,"_",
            periode,
".RData"))
}else{
load(paste0(reppheno,"/","pheno_",culture,"_chardonnay_",correction,"_",
            periode,
".RData"))
}

##Mettre dans un dataframe
datapheno.df <- data.frame(DATA_PHENO)

assign(paste0("datapheno.df.",correction),datapheno.df,envir = globalenv())
}
}
#############################################################################
##Fonction pour calculer l'indicateur
#############################################################################
calcul.stat.indicateur<- function(indicateur,statistique){ #charger.donnees()
  #definir.nb.sites()

for (correction in correctionliste){

#Créer un vecteur vide pour la variable
stat_indicateur <- rep(NA,nb_sites)

#Renommer la table de données
datapheno.df <- get(paste0("datapheno.df.",correction))

#Pour chaque site, la variable sur toutes les années
for (i in 1:nb_sites){
indicateur.vect <- datapheno.df[,indicateur][datapheno.df$site==i]
stat_indicateur[i] <- statistique(indicateur.vect)
}

#Chaque vecteur portera le nom de la correction
assign(paste0("stat_indicateur_",correction),stat_indicateur,envir = globalenv())
}

}
#############################################################################
##Fonction pour trouver le max et le min de l'indicateurs
#############################################################################
```

```
max_min_indicateur <- function(indicateur,statistique){ #calcul.stat.indicateur

stat_indicateurs_all_corrections <<- c(NA)

for (correction in correctionliste){
stat_indicateurs_correction <- eval(as.name(paste0("stat_indicateur_",correction)))
stat_indicateurs_all_corrections <<- c(
stat_indicateurs_all_corrections,stat_indicateurs_correction)
}

stat_indicateurs_all_corrections <<- na.omit(stat_indicateurs_all_corrections)

max_indicateur <<- max(stat_indicateurs_all_corrections)
min_indicateur <<- min(stat_indicateurs_all_corrections)

}
##############################################################################
##Fonction pour retrouver le placements des sites avec les données initiales
##############################################################################

definir.grille <- function(region,periode){ #calcul.stat.indicateur()

load(paste0(repinitial,"/MBC_",region,"/",region,"/MBC/BC_data_day_","1dCDFt",
"_1985_2014_2036_2065_",region,".RData"))

longitude <<- get(paste0("LON_SAFRAN_",region))
latitude <<- get(paste0("LAT_SAFRAN_",region))

##On prend une variable pour retrouver les dimensions
pr <- eval(as.name(paste0("pr_day_1dCDFt_",periode,"_",region)))
df <- apply(pr,c(1,2),mean)

##Check de la dimension
dim(df)

##Créer un booléen
grid_NA <- is.na(df)

##Transposé le tableau car les données qui seront insérées ont été lues en colonnes
t.grid_NA <- t(grid_NA)

##Créer un nouveau tableau qui sera rempli avec les valeurs
t.imageplot_stat_indicateur <- t.grid_NA

for (correction in correctionliste){

##On déclare la variable stat_indicateur pour faciliter l'usage
```

```
stat_indicateur <- eval(as.name(paste0("stat_indicateur_",correction)))

##Boucle qui remplie le tableau
##Si la case contenait une valeur initialement, on insère la médiane
##Sinon on vide la case
k=1
for (i in 1:nrow(t.grid_NA)){
for (j in 1:ncol(t.grid_NA)){
if (t.grid_NA[i,j]==FALSE){
t.imageplot_stat_indicateur[i,j] <- stat_indicateur[k]
k=k+1
}else{
t.imageplot_stat_indicateur[i,j] <- NA
}
}
}

##On retranspose pour avoir le tableau initale
imageplot_stat_indicateur <- t(t.imageplot_stat_indicateur)
assign(paste0("imageplot_stat_indicateur_",correction),imageplot_stat_indicateur,
       envir = globalenv())

} #correctionliste
} #definir.grille

#definir.grille(region,periode)

##############################################################################
##Fonction pour faire la cartographie
##############################################################################

cartographie.indicateur <- function(region,culture,periode,indicateur,statistique){

library(fields)

definir.nb.correction(periode)
definir.nb.sites(region)
charger.donnees(region,culture,periode)
calcul.stat.indicateur(indicateur,statistique)
max_min_indicateur(indicateur,statistique)
definir.grille(region,periode)

statistique_text <- ifelse(identical(statistique,mean)==TRUE,"Moyenne",
ifelse(identical(statistique,median)==TRUE,"Médiane",
ifelse(identical(statistique,sd)==TRUE,"Ecart-type",
ifelse(identical(statistique,var)==TRUE,"Variance",
ifelse(identical(statistique,max)==TRUE,"Maximum",
```

```
ifelse(identical(statistique,min)==TRUE,"Minimum",""
))))))

for (correction in correctionliste){

imageplot_stat_indicateur <-  eval(as.name(paste0("imageplot_stat_indicateur_",
                                                  correction)))

image.plot(longitude,latitude,imageplot_stat_indicateur,
main=correction,zlim=c(min_indicateur,max_indicateur),
xlab="longitude",ylab="latitude")
}

mtext(paste(statistique_text,"-",indicateur,"PHENO","-",region,"-",culture,"-",
            periode,
            sep=" "), outer=TRUE,  cex=1, line=-1.15)

}
```

Script : Cartographie_ET0_BH.r

```
#########################################################################
###Script: CARTOGRAPHIE ET0
##Projet Compromise
##12/06/2020
#########################################################################
#------NETTOYAGE ENVIRONNEMENT TRAVAIL-----------------------------------
rm(list=ls())

#Définir le répertoire initiale
#A modifier selon votre répertoire
repinitial <- c("C:/Users/Ling/Desktop/Stage INRAE/Dossier de travail")

#Définir le répertoire de travail
reptravail <- paste(repinitial,"/Code/", sep="")
setwd(reptravail)

#Définir les arguments   #Formats différents ne sont pas acceptés
# x="ET0"                 ## "SWC" ou "WATER_RESERVE"
# region="Bretagne"       ## "Provence" ou "Ile_de_France" ou "Bretagne"
# culture="ble"           ## "mais" ou "vigne" ou "ble"
# periode="1985_2014"     ## "2036_2065" ou "1985_2014"
# mois=07                 ## 01,02,03,...,12
# statistique = mean      ##mean median max min sd

source("fonction_cartographie_ET0_BH.R")
par(mfrow=c(2,4),mar=c(5.1, 3.7, 5.1, 4))
cartographie.indicateur("WATER_RESERVE","Bretagne","mais","2036_2065",07,sd)
```

Script : fonction_cartographie_ET0_BH.r

```
##############################################################################
##Fonction pour définir le nbre de corrections
##############################################################################
definir.nb.correction <- function(periode){

if (periode=="1985_2014"){
correctionliste <<- c("SAFRAN","1dCDFt","IntervardOTC","IntervarR2D2",
                      "SpatialdOTC","SpatialR2D2","SpatIntervardOTC","SpatIntervarR2D2")
}

if (periode=="2036_2065"){
correctionliste <<- c("1dCDFt","IntervardOTC","IntervarR2D2","SpatialdOTC","SpatialR2D2"
                      "SpatIntervardOTC","SpatIntervarR2D2")
}
}
##############################################################################
##Fonction pour définir le répertoire des données
##############################################################################
definir.repindicateur <- function(x,region,culture,periode){

if(x=="ET0"){repindicateur <<- paste0(reptravail,"Compromise/",region,"/",culture,"/",
                                      "ET0")}
if(x=="WATER_RESERVE" | x=="SWC"){ repindicateur <<- paste0(reptravail,"Compromise/",
                                                           region,"/",culture,"/","BH")
}

##############################################################################
##Fonction pour charger les données dans l'environnement de travail
##############################################################################

charger.donnees <- function(x,region,culture,periode){ #definir.repindicateur()

for(correction in correctionliste){

##Charger les données dans l'environnement de travail
if(x=="ET0"){
load(paste0(repindicateur,"/","ET0","_",correction,"_",periode,".RData"))
indicateur_df <- data.frame(DATA_CLIMAT_ET0)
}
if(x=="WATER_RESERVE"){
if(culture=="mais" | culture =="ble"){
load(paste0(repindicateur,"/","reserve_",culture,"_precoce_",correction,"_",
            periode,
            "_soil1",".RData"))
}else{
```

```
load(paste0(repindicateur,"/","reserve_",culture,"_chardonnay_",correction,"_",
            periode,
            "_soil1",".RData"))
}
indicateur_df <- data.frame(WATER_RESERVE)
}
if(x=="SWC"){
if(culture=="mais" | culture =="ble"){
load(paste0(repindicateur,"/","swc_",culture,"_precoce_",correction,"_",
            periode,
            "_soil1",".RData"))
}else{
load(paste0(repindicateur,"/","swc_",culture,"_chardonnay_",correction,"_",
            periode,
            "_soil1",".RData"))
}
indicateur_df <<- data.frame(SOIL_WATER_CONTENT)
}

##Création des noms de colonnes datées
if (periode=="1985_2014"){source("fonction_dated_colnames_1985_2014.R")}

if (periode=="2036_2065"){source("fonction_dated_colnames_2036_2065.R")}

##Nommer les colonnes avec les dates
##Si c'est le BH pour la vigne on a 29 ans seulement
if(x!="ET0" && culture=="vigne"){
if (periode=="1985_2014"){colnames(indicateur_df) <- nom_colonnes_datees[366:10957]}
if (periode=="2036_2065"){colnames(indicateur_df) <- nom_colonnes_datees[367:10958]}
}else{colnames(indicateur_df) <- nom_colonnes_datees}

assign(paste0("indicateur_df_",correction),indicateur_df,envir = globalenv())

}
}

#############################################################################
##Fonction pour calculer l'indicateur
#############################################################################

calcul.stat.indicateur<- function(mois,statistique){

for(correction in correctionliste){

indicateur_df <- eval(as.name(paste0("indicateur_df_",correction)))

##Filtre selon le mois sélectionné
```

```
filtre_mois <<- as.integer(substr(colnames(indicateur_df),6,7))==mois

##Filtrer le dataframe
indicateur_df_mois <- indicateur_df[,filtre_mois]

##La statistique sur 30 ans pour chaque site
stat_indicateur <- apply(indicateur_df_mois,1,statistique)

assign(paste0("stat_indicateur_",correction),stat_indicateur,envir = globalenv())

}
}

#############################################################################
##Fonction pour trouver le max et le min de l'indicateurs
#############################################################################

max_min_indicateur <- function(mois,statistique){ #calcul.stat.indicateur

stat_indicateurs_all_corrections <<- c(NA)

for (correction in correctionliste){
stat_indicateurs_correction <- eval(as.name(paste0("stat_indicateur_",correction)))
stat_indicateurs_all_corrections <<-c(
  stat_indicateurs_all_corrections,stat_indicateurs_correction)
}

stat_indicateurs_all_corrections <<- na.omit(stat_indicateurs_all_corrections)

max_indicateur <<- max(stat_indicateurs_all_corrections)
min_indicateur <<- min(stat_indicateurs_all_corrections)

}

#############################################################################
##Fonction pour retrouver le placements des sites avec les données initiales
#############################################################################

definir.grille <- function(region,periode){ #calcul.stat.indicateur()

load(paste0(repinitial,"/MBC_",region,"/",region,"/MBC/BC_data_day_","1dCDFt",
            "_1985_2014_2036_2065_",region,".RData"))

longitude <<- get(paste0("LON_SAFRAN_",region))
latitude <<- get(paste0("LAT_SAFRAN_",region))

##On prend une variable pour retrouver les dimensions
```

```
pr <- eval(as.name(paste0("pr_day_1dCDFt_",periode,"_",region)))
df <- apply(pr,c(1,2),mean)

##Check de la dimension
dim(df)

##Créer un booléen
grid_NA <- is.na(df)

##Transposé le tableau car les données qui seront insérées ont été lues en colonnes
t.grid_NA <- t(grid_NA)

##Créer un nouveau tableau qui sera rempli avec les valeurs
t.imageplot_stat_indicateur <- t.grid_NA

for (correction in correctionliste){

##On déclare la variable stat_indicateur pour faciliter l'usage
stat_indicateur <- eval(as.name(paste0("stat_indicateur_",correction)))

##Boucle qui remplie le tableau
##Si la case contenait une valeur initialement, on insère la médiane
##Sinon on vide la case
k=1
for (i in 1:nrow(t.grid_NA)){
for (j in 1:ncol(t.grid_NA)){
if (t.grid_NA[i,j]==FALSE){
t.imageplot_stat_indicateur[i,j] <- stat_indicateur[k]
k=k+1
}else{
t.imageplot_stat_indicateur[i,j] <- NA
}
}
}

##On retranspose pour avoir le tableau initale
imageplot_stat_indicateur <- t(t.imageplot_stat_indicateur)
assign(paste0("imageplot_stat_indicateur_",correction),imageplot_stat_indicateur,
       envir = globalenv())

} #correctionliste
} #definir.grille

################################################################################
##Fonction pour faire la cartographie
################################################################################
cartographie.indicateur <- function(x,region,culture,periode,mois,statistique){
```

```
library(fields)

definir.nb.correction(periode)
definir.repindicateur(x,region,culture,periode)
charger.donnees(x,region,culture,periode)
calcul.stat.indicateur(mois,statistique)
max_min_indicateur(mois,statistique)
definir.grille(region,periode)

statistique_text <- ifelse(identical(statistique,mean)==TRUE,"Moyenne",
ifelse(identical(statistique,median)==TRUE,"Médiane",
ifelse(identical(statistique,sd)==TRUE,"Ecart-type",
ifelse(identical(statistique,var)==TRUE,"Variance",
ifelse(identical(statistique,max)==TRUE,"Maximum",
ifelse(identical(statistique,min)==TRUE,"Minimum",""
))))))

nom_mois <- ifelse(mois==1,"janvier",
ifelse(mois==2,"février",
ifelse(mois==3,"mars",
ifelse(mois==4,"avril",
ifelse(mois==5,"mai",
ifelse(mois==6,"juin",
ifelse(mois==7,"juillet",
ifelse(mois==8,"août",
ifelse(mois==9,"septembre",
ifelse(mois==10,"octobre",
ifelse(mois==11,"novembre",
ifelse(mois==11,"décembre","")))))))))))

for (correction in correctionliste){

imageplot_stat_indicateur <-  eval(as.name(paste0("imageplot_stat_indicateur_",
                                                  correction)))

image.plot(longitude,latitude,imageplot_stat_indicateur,main=correction,
zlim=c(min_indicateur,max_indicateur),xlab="longitude",ylab="latitude")
}

mtext(paste(statistique_text,x,"en",nom_mois,"-",region,"-",culture,"-",
            periode,
            sep=" "), outer=TRUE,  cex=1, line=-1.15)
}
```

Script R : check_variable_climatiques_cartographies.R

```
#Définir le répertoire initiale
#A modifier selon votre répertoire
repinitial <- c("C:/Users/Ling/Desktop/Stage INRAE/Dossier de travail")

#Définir le répertoire de travail
reptravail <- paste(repinitial,"/Code/", sep="")
setwd(reptravail)

##############################################################################
##Fonction pour définir le nbre de corrections
##############################################################################
definir.nb.correction <- function(periode){

if (periode=="1985_2014"){
correctionliste <<- c("SAFRAN","1dCDFt","IntervardOTC","IntervarR2D2",
                      "SpatialdOTC","SpatialR2D2","SpatIntervardOTC","SpatIntervarR2D2")
}

if (periode=="2036_2065"){
correctionliste <<- c("1dCDFt","IntervardOTC","IntervarR2D2","SpatialdOTC",
                      "SpatialR2D2","SpatIntervardOTC","SpatIntervarR2D2")
}
}
##############################################################################
##Fonction pour définir la grille
##############################################################################

definir.grille <- function(region){

load(paste0(repinitial,"/MBC_",region,"/",region,"/MBC/BC_data_day_","1dCDFt",
            "_1985_2014_2036_2065_",region,".RData"))

longitude <<- get(paste0("LON_SAFRAN_",region))
latitude <<- get(paste0("LAT_SAFRAN_",region))

##On prend une variable pour retrouver les dimensions
pr <- eval(as.name(paste0("pr_day_1dCDFt_","1985_2014","_",region)))
df <- apply(pr,c(1,2),mean)

##Check de la dimension
dim(df)

##Créer ue grille booléene
grid_NA <- is.na(df)
```

```
##Transposé le tableau car les données qui seront insérées ont été lues en colonnes
t.grid_NA <<- t(grid_NA)

}
################################################################################
##Fonction pour remplir la grille
################################################################################

remplir.grille <- function(region,periode){

definir.grille(region)

#Définir le répertoire des variables climatiques compromise
repdatacompromise <- paste0(repinitial,"/MBC_",region,"/",region,"/DATACOMPROMISE")

for (correction in correctionliste){

#réinistialiser la grille booléene
t.tas_grid <- t.grid_NA

if (correction != "SAFRAN") {
load(paste0(repdatacompromise,"/BC_data_day_",correction,"_",
            periode,"_",region,"/",
            "tas_day_",correction,"_",periode,"_",region,".RData"))
}

if (correction == "SAFRAN") {
load(paste0(repdatacompromise,"/Ref_data_day_",correction,"_","1985_2014","_",region,"/"
            "tas_day_",correction,"_",periode,"_",region,".RData"))
}

#calculer la température moyenne sur 30 ans
tas.all.years <- apply(DATA_CLIMAT_tmoy,1,mean)

#remplir les cases de la grille booléene
k=1
for (i in 1:nrow(t.grid_NA)){
for (j in 1:ncol(t.grid_NA)){
if (t.grid_NA[i,j]==FALSE){
t.tas_grid[i,j] <- tas.all.years[k]
k=k+1
}else{
t.tas_grid[i,j] <- NA
}
}
} #nrow
```

```
tas_grid <- t(t.tas_grid)

assign(paste0("tas_grid_",correction),tas_grid,envir = globalenv())

} #correctionliste
}
################################################################################
##Fonction pour faire la cartographie des variables climatiques
################################################################################

par(mfrow=c(2,4))

cartographie.varclim <- function(region,periode){

definir.nb.correction(periode)
definir.grille(region)
remplir.grille(region,periode)

tas_grid_all <- c(tas_grid_1dCDFt,tas_grid_IntervardOTC,tas_grid_IntervarR2D2,
tas_grid_SAFRAN,tas_grid_SpatialdOTC,tas_grid_SpatialR2D2,
tas_grid_SpatIntervardOTC,tas_grid_SpatIntervarR2D2)
tas_grid_all <- na.omit(tas_grid_all)
max_tas <- max(tas_grid_all)
min_tas <- min(tas_grid_all)

for (correction in correctionliste){

tas_grid <- eval(as.name(paste0("tas_grid_",correction)))

image.plot(longitude,latitude,tas_grid,main=correction,
zlim=c(min_tas,max_tas),xlab="longitude",ylab="latitude")

}
}
# par(mfrow=c(2,4))
# cartographie.varclim("Ile_de_France","1985_2014")
```

Script : prepare_datafwi.r

```
##Bibliothèques
library(tibble)

##Sélection de la région
region="Ile_de_France"

##Définir les méthodes
correctionliste = c("1dCDFt","IntervardOTC","IntervarR2D2","SpatialdOTC",
                    "SpatialR2D2","SpatIntervardOTC","SpatIntervarR2D2")
#correctionliste = c("SAFRAN")

periode_passe_futur <- c("2036_2065") #définir les deux périodes passée et futur
#periode_passe_futur <- c("1985_2014")

##Définir les dimensions de la matrice
if(region=="Bretagne"){
grid_dim=c(21,20)
grid_pixel=grid_dim[1]*grid_dim[2]
}
if(region=="Ile_de_France"){
grid_dim=c(12,27)
grid_pixel=grid_dim[1]*grid_dim[2]
}
if(region=="Provence"){
grid_dim=c(24,20)
grid_pixel=grid_dim[1]*grid_dim[2]
}

##Définir le répertoire de travail initial
repinitial = c("C:/Users/Ling/Desktop/Stage INRAE/Dossier de travail")

##Définir le répertoire des corrections (répertoire climatique)
repclim  <- paste0(repinitial,"/MBC_",region,"/",region,"/","MBC")

#Création d'un répertoire pour les fichiers nettoyés
repcompromise <- paste0(repinitial,"/MBC_",region,"/",region,"/","DATACOMPROMISE")
dir.create(repcompromise, showWarnings=FALSE, recursive = TRUE)


####----Création des répertoires,
##nettoyages des variables, créations des nouveaux fichiers----####
for (correction in correctionliste){        #boucle sur les corrections
print(correction)

#####-----Créations de répetoires------####
```

```
#####-----lecture du fichier .RData avec toutes les variables climatiques------####


for (periode in periode_passe_futur){
print(periode)

if (correction != "SAFRAN"){
repcorrection <- paste0(repcompromise,"/","BC_data_day_",correction,"_",
                        periode,
                        "_",region)
dir.create(repcorrection, showWarnings=FALSE, recursive = TRUE)

load(paste0(repclim,"/BC_data_day_",correction,"_1985_2014_2036_2065_",region,
            ".RData"))
}

if (correction == "SAFRAN"){

repcorrection <- paste0(repcompromise,"/","Ref_data_day_",correction,"_",
                        periode,
                        "_",region)
dir.create(repcorrection, showWarnings=FALSE, recursive = TRUE)

load(paste0(repinitial,"/MBC_",region,"/",region,"/SAFRAN/Ref_data_day_",correction,
            "_1985_2014_",region,".RData"))
}


if (periode == "1985_2014"){
#####-----Nettoyage des données-----------#####
##On charge chaque variable
##On insère dans une matrice de dimension "nbre de pixel" x "nbre de jour"
##On enlève les lignes non renseignées
##On rajoute le 32 décembre pour les années bissextiles
##On sauvegarde la nouvelle matrice

##température
tmoy <- eval(as.name(paste0("tas_day_",correction,"_",periode,"_",region)))
DATA_CLIMAT_tmoy <- matrix(c(tmoy[,,]),grid_pixel,10950)
DATA_CLIMAT_tmoy <- na.omit(DATA_CLIMAT_tmoy)
rm(tmoy)
DATA_CLIMAT_tmoy <- data.frame(DATA_CLIMAT_tmoy)
DATA_CLIMAT_tmoy <- add_column(DATA_CLIMAT_tmoy,X1460b=DATA_CLIMAT_tmoy$X1460,
                               .after = 1460)
DATA_CLIMAT_tmoy <- add_column(DATA_CLIMAT_tmoy,X2921b=DATA_CLIMAT_tmoy$X2921,
                               .after = 2921)
DATA_CLIMAT_tmoy <- add_column(DATA_CLIMAT_tmoy,X4382b=DATA_CLIMAT_tmoy$X4382,
```

```
                                .after = 4382)
DATA_CLIMAT_tmoy <- add_column(DATA_CLIMAT_tmoy,X5843b=DATA_CLIMAT_tmoy$X5843,
                                .after = 5843)
DATA_CLIMAT_tmoy <- add_column(DATA_CLIMAT_tmoy,X7304b=DATA_CLIMAT_tmoy$X7304,
                                .after = 7304)
DATA_CLIMAT_tmoy <- add_column(DATA_CLIMAT_tmoy,X8765b=DATA_CLIMAT_tmoy$X8765,
                                .after = 8765)
DATA_CLIMAT_tmoy <- add_column(DATA_CLIMAT_tmoy,X10226b=DATA_CLIMAT_tmoy$X10226,
                                .after = 10226)
DATA_CLIMAT_tmoy <- data.matrix(DATA_CLIMAT_tmoy)
save(DATA_CLIMAT_tmoy,file = paste0(repcorrection,"/","tas_day_",correction,"_",periode,
                                    "_",region,".RData"))
print("DATA_CLIMAT_tmoy")

##pluie
rain <- eval(as.name(paste0("pr_day_",correction,"_",periode,"_",region)))
DATA_CLIMAT_rain <- matrix(c(rain[,,]),grid_pixel,10950)
DATA_CLIMAT_rain <- na.omit(DATA_CLIMAT_rain)
rm(rain)
DATA_CLIMAT_rain <- data.frame(DATA_CLIMAT_rain)
DATA_CLIMAT_rain <- add_column(DATA_CLIMAT_rain,X1460b=DATA_CLIMAT_rain$X1460,
                                .after = 1460)
DATA_CLIMAT_rain <- add_column(DATA_CLIMAT_rain,X2921b=DATA_CLIMAT_rain$X2921,
                                .after = 2921)
DATA_CLIMAT_rain <- add_column(DATA_CLIMAT_rain,X4382b=DATA_CLIMAT_rain$X4382,
                                .after = 4382)
DATA_CLIMAT_rain <- add_column(DATA_CLIMAT_rain,X5843b=DATA_CLIMAT_rain$X5843,
                                .after = 5843)
DATA_CLIMAT_rain <- add_column(DATA_CLIMAT_rain,X7304b=DATA_CLIMAT_rain$X7304,
                                .after = 7304)
DATA_CLIMAT_rain <- add_column(DATA_CLIMAT_rain,X8765b=DATA_CLIMAT_rain$X8765,
                                .after = 8765)
DATA_CLIMAT_rain <- add_column(DATA_CLIMAT_rain,X10226b=DATA_CLIMAT_rain$X10226,
                                .after = 10226)
DATA_CLIMAT_rain <- data.matrix(DATA_CLIMAT_rain)
save(DATA_CLIMAT_rain,file = paste0(repcorrection,"/","pr_day_",correction,"_",periode,
                                    "_",region,".RData"))
print("DATA_CLIMAT_rain")

##rayonnement
rg <- eval(as.name(paste0("rsds_day_",correction,"_",periode,"_",region)))
DATA_CLIMAT_rg<- matrix(c(rg[,,]),grid_pixel,10950)
DATA_CLIMAT_rg <- na.omit(DATA_CLIMAT_rg)
rm(rg)
DATA_CLIMAT_rg <- data.frame(DATA_CLIMAT_rg)
DATA_CLIMAT_rg <- add_column(DATA_CLIMAT_rg,X1460b=DATA_CLIMAT_rg$X1460,.after = 1460)
DATA_CLIMAT_rg <- add_column(DATA_CLIMAT_rg,X2921b=DATA_CLIMAT_rg$X2921,.after = 2921)
```

```
DATA_CLIMAT_rg <- add_column(DATA_CLIMAT_rg,X4382b=DATA_CLIMAT_rg$X4382,.after = 4382)
DATA_CLIMAT_rg <- add_column(DATA_CLIMAT_rg,X5843b=DATA_CLIMAT_rg$X5843,.after = 5843)
DATA_CLIMAT_rg <- add_column(DATA_CLIMAT_rg,X7304b=DATA_CLIMAT_rg$X7304,.after = 7304)
DATA_CLIMAT_rg <- add_column(DATA_CLIMAT_rg,X8765b=DATA_CLIMAT_rg$X8765,.after = 8765)
DATA_CLIMAT_rg <- add_column(DATA_CLIMAT_rg,X10226b=DATA_CLIMAT_rg$X10226,
                             .after = 10226)
DATA_CLIMAT_rg <- data.matrix(DATA_CLIMAT_rg)
save(DATA_CLIMAT_rg,file = paste0(repcorrection,"/","rsds_day_",correction,"_",periode,
                                  "_",region,".RData"))
print("DATA_CLIMAT_rg")

##vent
wind <- eval(as.name(paste0("sfcWind_day_",correction,"_",periode,"_",region)))
DATA_CLIMAT_wind <- matrix(c(wind[,,]),grid_pixel,10950)
DATA_CLIMAT_wind <- na.omit(DATA_CLIMAT_wind)
DATA_CLIMAT_wind <- data.frame(DATA_CLIMAT_wind)
DATA_CLIMAT_wind <- add_column(DATA_CLIMAT_wind,X1460b=DATA_CLIMAT_wind$X1460,
                               .after = 1460)
DATA_CLIMAT_wind <- add_column(DATA_CLIMAT_wind,X2921b=DATA_CLIMAT_wind$X2921,
                               .after = 2921)
DATA_CLIMAT_wind <- add_column(DATA_CLIMAT_wind,X4382b=DATA_CLIMAT_wind$X4382,
                               .after = 4382)
DATA_CLIMAT_wind <- add_column(DATA_CLIMAT_wind,X5843b=DATA_CLIMAT_wind$X5843,
                               .after = 5843)
DATA_CLIMAT_wind <- add_column(DATA_CLIMAT_wind,X7304b=DATA_CLIMAT_wind$X7304,
                               .after = 7304)
DATA_CLIMAT_wind <- add_column(DATA_CLIMAT_wind,X8765b=DATA_CLIMAT_wind$X8765,
                               .after = 8765)
DATA_CLIMAT_wind <- add_column(DATA_CLIMAT_wind,X10226b=DATA_CLIMAT_wind$X10226,
                               .after = 10226)
DATA_CLIMAT_wind <- data.matrix(DATA_CLIMAT_wind)
rm(wind)
save(DATA_CLIMAT_wind,file = paste0(repcorrection,"/","sfcWind_day_",correction,"_",
                                    periode, "_",region,".RData"))
print("DATA_CLIMAT_wind")

##humidité
rh <- eval(as.name(paste0("hurs_day_",correction,"_",periode,"_",region)))
DATA_CLIMAT_rh <- matrix(c(rh[,,]),grid_pixel,10950)
DATA_CLIMAT_rh <- na.omit(DATA_CLIMAT_rh)
rm(rh)
DATA_CLIMAT_rh <- data.frame(DATA_CLIMAT_rh)
DATA_CLIMAT_rh <- add_column(DATA_CLIMAT_rh,X1460b=DATA_CLIMAT_rh$X1460,.after = 1460)
DATA_CLIMAT_rh <- add_column(DATA_CLIMAT_rh,X2921b=DATA_CLIMAT_rh$X2921,.after = 2921)
DATA_CLIMAT_rh <- add_column(DATA_CLIMAT_rh,X4382b=DATA_CLIMAT_rh$X4382,.after = 4382)
DATA_CLIMAT_rh <- add_column(DATA_CLIMAT_rh,X5843b=DATA_CLIMAT_rh$X5843,.after = 5843)
DATA_CLIMAT_rh <- add_column(DATA_CLIMAT_rh,X7304b=DATA_CLIMAT_rh$X7304,.after = 7304)
```

```
DATA_CLIMAT_rh <- add_column(DATA_CLIMAT_rh,X8765b=DATA_CLIMAT_rh$X8765,.after = 8765)
DATA_CLIMAT_rh <- add_column(DATA_CLIMAT_rh,X10226b=DATA_CLIMAT_rh$X10226,
                             .after = 10226)
DATA_CLIMAT_rh <- data.matrix(DATA_CLIMAT_rh)
save(DATA_CLIMAT_rh,file = paste0(repcorrection,"/","hurs_day_",correction,"_",periode,
                                  "_",region,".RData"))
print("DATA_CLIMAT_rh")
}
else{

##température
tmoy <- eval(as.name(paste0("tas_day_",correction,"_",periode,"_",region)))
DATA_CLIMAT_tmoy <- matrix(c(tmoy[,,]),grid_pixel,10950)
DATA_CLIMAT_tmoy <- na.omit(DATA_CLIMAT_tmoy)
rm(tmoy)
DATA_CLIMAT_tmoy <- data.frame(DATA_CLIMAT_tmoy)
DATA_CLIMAT_tmoy <- add_column(DATA_CLIMAT_tmoy,X365b=DATA_CLIMAT_tmoy$X365,
                               .after = 365)
DATA_CLIMAT_tmoy <- add_column(DATA_CLIMAT_tmoy,X1826b=DATA_CLIMAT_tmoy$X1826,
                               .after = 1826)
DATA_CLIMAT_tmoy <- add_column(DATA_CLIMAT_tmoy,X3287b=DATA_CLIMAT_tmoy$X3287,
                               .after = 3287)
DATA_CLIMAT_tmoy <- add_column(DATA_CLIMAT_tmoy,X4748b=DATA_CLIMAT_tmoy$X4748,
                               .after = 4748)
DATA_CLIMAT_tmoy <- add_column(DATA_CLIMAT_tmoy,X6209b=DATA_CLIMAT_tmoy$X6209,
                               .after = 6209)
DATA_CLIMAT_tmoy <- add_column(DATA_CLIMAT_tmoy,X7670b=DATA_CLIMAT_tmoy$X7670,
                               .after = 7670)
DATA_CLIMAT_tmoy <- add_column(DATA_CLIMAT_tmoy,X9131b=DATA_CLIMAT_tmoy$X9131,
                               .after = 9131)
DATA_CLIMAT_tmoy <- add_column(DATA_CLIMAT_tmoy,X10592b=DATA_CLIMAT_tmoy$X10592,
                               .after = 10592)
DATA_CLIMAT_tmoy <- data.matrix(DATA_CLIMAT_tmoy)
save(DATA_CLIMAT_tmoy,file = paste0(repcorrection,"/","tas_day_",correction,"_",periode,
                                    "_",region,".RData"))
print("DATA_CLIMAT_tmoy")

##pluie
rain <- eval(as.name(paste0("pr_day_",correction,"_",periode,"_",region)))
DATA_CLIMAT_rain <- matrix(c(rain[,,]),grid_pixel,10950)
DATA_CLIMAT_rain <- na.omit(DATA_CLIMAT_rain)
rm(rain)
DATA_CLIMAT_rain <- data.frame(DATA_CLIMAT_rain)
DATA_CLIMAT_rain <- add_column(DATA_CLIMAT_rain,X365b=DATA_CLIMAT_rain$X365,
                               .after = 365)
DATA_CLIMAT_rain <- add_column(DATA_CLIMAT_rain,X1826b=DATA_CLIMAT_rain$X1826,
                               .after = 1826)
```

```
DATA_CLIMAT_rain <- add_column(DATA_CLIMAT_rain,X3287b=DATA_CLIMAT_rain$X3287,
                               .after = 3287)
DATA_CLIMAT_rain <- add_column(DATA_CLIMAT_rain,X4748b=DATA_CLIMAT_rain$X4748,
                               .after = 4748)
DATA_CLIMAT_rain <- add_column(DATA_CLIMAT_rain,X6209b=DATA_CLIMAT_rain$X6209,
                               .after = 6209)
DATA_CLIMAT_rain <- add_column(DATA_CLIMAT_rain,X7670b=DATA_CLIMAT_rain$X7670,
                               .after = 7670)
DATA_CLIMAT_rain <- add_column(DATA_CLIMAT_rain,X9131b=DATA_CLIMAT_rain$X9131,
                               .after = 9131)
DATA_CLIMAT_rain <- add_column(DATA_CLIMAT_rain,X10592b=DATA_CLIMAT_rain$X10592,
                               .after = 10592)
DATA_CLIMAT_rain <- data.matrix(DATA_CLIMAT_rain)
save(DATA_CLIMAT_rain,file = paste0(repcorrection,"/","pr_day_",correction,"_",periode,
                                    "_",region,".RData"))
print("DATA_CLIMAT_rain")

##rayonnement
rg <- eval(as.name(paste0("rsds_day_",correction,"_",periode,"_",region)))
DATA_CLIMAT_rg<- matrix(c(rg[,,]),grid_pixel,10950)
DATA_CLIMAT_rg <- na.omit(DATA_CLIMAT_rg)
rm(rg)
DATA_CLIMAT_rg <- data.frame(DATA_CLIMAT_rg)
DATA_CLIMAT_rg <- add_column(DATA_CLIMAT_rg,X365b=DATA_CLIMAT_rg$X365,.after = 365)
DATA_CLIMAT_rg <- add_column(DATA_CLIMAT_rg,X1826b=DATA_CLIMAT_rg$X1826,.after = 1826)
DATA_CLIMAT_rg <- add_column(DATA_CLIMAT_rg,X3287b=DATA_CLIMAT_rg$X3287,.after = 3287)
DATA_CLIMAT_rg <- add_column(DATA_CLIMAT_rg,X4748b=DATA_CLIMAT_rg$X4748,.after = 4748)
DATA_CLIMAT_rg <- add_column(DATA_CLIMAT_rg,X6209b=DATA_CLIMAT_rg$X6209,.after = 6209)
DATA_CLIMAT_rg <- add_column(DATA_CLIMAT_rg,X7670b=DATA_CLIMAT_rg$X7670,.after = 7670)
DATA_CLIMAT_rg <- add_column(DATA_CLIMAT_rg,X9131b=DATA_CLIMAT_rg$X9131,.after = 9131)
DATA_CLIMAT_rg <- add_column(DATA_CLIMAT_rg,X10592b=DATA_CLIMAT_rg$X10592,
                             .after = 10592)
DATA_CLIMAT_rg <- data.matrix(DATA_CLIMAT_rg)
save(DATA_CLIMAT_rg,file = paste0(repcorrection,"/","rsds_day_",correction,"_",periode,
                                  "_",region,".RData"))
print("DATA_CLIMAT_rg")

##vent
wind <- eval(as.name(paste0("sfcWind_day_",correction,"_",periode,"_",region)))
DATA_CLIMAT_wind <- matrix(c(wind[,,]),grid_pixel,10950)
DATA_CLIMAT_wind <- na.omit(DATA_CLIMAT_wind)
DATA_CLIMAT_wind <- data.frame(DATA_CLIMAT_wind)
DATA_CLIMAT_wind <- add_column(DATA_CLIMAT_wind,X365b=DATA_CLIMAT_wind$X365,
                               .after = 365)
DATA_CLIMAT_wind <- add_column(DATA_CLIMAT_wind,X1826b=DATA_CLIMAT_wind$X1826,
                               .after = 1826)
DATA_CLIMAT_wind <- add_column(DATA_CLIMAT_wind,X3287b=DATA_CLIMAT_wind$X3287,
```

```
                                .after = 3287)
DATA_CLIMAT_wind <- add_column(DATA_CLIMAT_wind,X4748b=DATA_CLIMAT_wind$X4748,
                                .after = 4748)
DATA_CLIMAT_wind <- add_column(DATA_CLIMAT_wind,X6209b=DATA_CLIMAT_wind$X6209,
                                .after = 6209)
DATA_CLIMAT_wind <- add_column(DATA_CLIMAT_wind,X7670b=DATA_CLIMAT_wind$X7670,
                                .after = 7670)
DATA_CLIMAT_wind <- add_column(DATA_CLIMAT_wind,X9131b=DATA_CLIMAT_wind$X9131,
                                .after = 9131)
DATA_CLIMAT_wind <- add_column(DATA_CLIMAT_wind,X10592b=DATA_CLIMAT_wind$X10592,
                                .after = 10592)
DATA_CLIMAT_wind <- data.matrix(DATA_CLIMAT_wind)
rm(wind)
save(DATA_CLIMAT_wind,file = paste0(repcorrection,"/","sfcWind_day_",correction,"_",
                                    periode,"_",region,".RData"))
print("DATA_CLIMAT_wind")

##humidité
rh <- eval(as.name(paste0("hurs_day_",correction,"_",periode,"_",region)))
DATA_CLIMAT_rh <- matrix(c(rh[,,]),grid_pixel,10950)
DATA_CLIMAT_rh <- na.omit(DATA_CLIMAT_rh)
rm(rh)
DATA_CLIMAT_rh <- data.frame(DATA_CLIMAT_rh)
DATA_CLIMAT_rh <- add_column(DATA_CLIMAT_rh,X365b=DATA_CLIMAT_rh$X365,.after = 365)
DATA_CLIMAT_rh <- add_column(DATA_CLIMAT_rh,X1826b=DATA_CLIMAT_rh$X1826,.after = 1826)
DATA_CLIMAT_rh <- add_column(DATA_CLIMAT_rh,X3287b=DATA_CLIMAT_rh$X3287,.after = 3287)
DATA_CLIMAT_rh <- add_column(DATA_CLIMAT_rh,X4748b=DATA_CLIMAT_rh$X4748,.after = 4748)
DATA_CLIMAT_rh <- add_column(DATA_CLIMAT_rh,X6209b=DATA_CLIMAT_rh$X6209,.after = 6209)
DATA_CLIMAT_rh <- add_column(DATA_CLIMAT_rh,X7670b=DATA_CLIMAT_rh$X7670,.after = 7670)
DATA_CLIMAT_rh <- add_column(DATA_CLIMAT_rh,X9131b=DATA_CLIMAT_rh$X9131,.after = 9131)
DATA_CLIMAT_rh <- add_column(DATA_CLIMAT_rh,X10592b=DATA_CLIMAT_rh$X10592,
                             .after = 10592)
DATA_CLIMAT_rh <- data.matrix(DATA_CLIMAT_rh)
save(DATA_CLIMAT_rh,file = paste0(repcorrection,"/","hurs_day_",correction,"_",periode,
                                    "_",region,".RData"))
print("DATA_CLIMAT_rh")

} #else

} #periode_passe_futur
} #correctionliste
```

Script : cartographie_fwi.r

```
#########################################################################
###Script: CARTOGRAPHIE FWI
##Projet Compromise
##06/06/2020
#########################################################################

#Définir le répertoire initiale
#A modifier selon votre répertoire
repinitial <- c("C:/Users/Ling/Desktop/Stage INRAE/Dossier de travail")

#Définir le répertoire de travail
reptravail <- paste(repinitial,"/Code/", sep="")
setwd(reptravail)

#Définir les arguments   #Formats différents ne sont pas acceptés
# region="Bretagne"      ## "Provence" ou "Ile_de_France" ou "Bretagne"
# periode="1985_2014"    ## "2036_2065" ou "1985_2014"
# indicateur="FWI"       ## FWI ou DSR
# statistique = mean     ##mean median max min

source("fonction_cartographie_fwi.R")
par(mfrow=c(2,4),mar=c(5.1, 3.7, 5.1, 4))
cartographie.indicateur("Bretagne","1985_2014","FWI",max)
```

Script fonction_cartographie_fwi.r

```
#########################################################################
###Script: fonction CARTOGRAPHIE FWI
##Projet Compromise
##18/06/2020
#########################################################################

#############################################################################################
##Fonction pour définir le nbre de corrections
#############################################################################################
definir.nb.correction <- function(periode){

if (periode=="1985_2014"){
correctionliste <<- c("SAFRAN","1dCDFt","IntervardOTC","IntervarR2D2",
                      "SpatialdOTC","SpatialR2D2","SpatIntervardOTC","SpatIntervarR2D2")
}

if (periode=="2036_2065"){
correctionliste <<- c("1dCDFt","IntervardOTC","IntervarR2D2","SpatialdOTC","SpatialR2D2"
                      "SpatIntervardOTC","SpatIntervarR2D2")
}
}
#############################################################################################
##Fonction pour définir le nbre de sites
#############################################################################################

definir.nb.sites <- function(region){

if(region=="Bretagne"){
nb_sites <<- 259
}
if(region=="Ile_de_France"){
nb_sites <<- 319
}
if(region=="Provence"){
nb_sites <<- 337
}

return(nb_sites)
}
#############################################################################################
##Fonction pour charger les données dans l'environnement de travail
#############################################################################################

charger.donnees <- function(region,periode){
```

```
reppheno <- paste0(reptravail,"Compromise/",region,"/","fwi")

for (correction in correctionliste){

##Charger le fichier
load(paste0(reppheno,"/","fwi_",correction,"_",periode,".RData"))

##Mettre dans un dataframe
datalist.df <- data.frame(datalist)

assign(paste0("datalist.df.",correction),datalist.df,envir = globalenv())
}
}
#############################################################################
##Fonction pour calculer l'indicateur
#############################################################################

calcul.stat.indicateur<- function(indicateur,statistique){ #charger.donnees()
                                                           #definir.nb.sites()

for (correction in correctionliste){

#Créer un vecteur vide pour la variable
stat_indicateur <- rep(NA,nb_sites)

#Renommer la table de données
datalist.df <- get(paste0("datalist.df.",correction))

#Pour chaque site, la variable sur 29 ans
for (i in 1:nb_sites){
indicateur.vect <- datalist.df[,indicateur][datalist.df$ID==i]
stat_indicateur[i] <- statistique(indicateur.vect)
}

#Chaque vecteur portera le nom de la correction
assign(paste0("stat_indicateur_",correction),stat_indicateur,envir = globalenv())
}

}
#############################################################################
##Fonction pour trouver le max et le min de l'indicateurs
#############################################################################
max_min_indicateur <- function(indicateur,statistique){ #calcul.stat.indicateur

stat_indicateurs_all_corrections <<- c(NA)

for (correction in correctionliste){
```

```
stat_indicateurs_correction <- eval(as.name(paste0("stat_indicateur_",correction)))
stat_indicateurs_all_corrections <<- c(
  stat_indicateurs_all_corrections,stat_indicateurs_correction)
}

stat_indicateurs_all_corrections <<- na.omit(stat_indicateurs_all_corrections)

max_indicateur <<- max(stat_indicateurs_all_corrections)
min_indicateur <<- min(stat_indicateurs_all_corrections)

}
################################################################################
##Fonction pour retrouver le placements des sites avec les données initiales
################################################################################
definir.grille <- function(region,periode){ #calcul.stat.indicateur()

load(paste0(repinitial,"/MBC_",region,"/",region,"/MBC/BC_data_day_","1dCDFt",
            "_1985_2014_2036_2065_",region,".RData"))

longitude <<- get(paste0("LON_SAFRAN_",region))
latitude <<- get(paste0("LAT_SAFRAN_",region))

##On prend une variable pour retrouver les dimensions
pr <- eval(as.name(paste0("pr_day_1dCDFt_",periode,"_",region)))
df <- apply(pr,c(1,2),mean)

##Check de la dimension
dim(df)

##Créer un booléen
grid_NA <- is.na(df)

##Transposé le tableau car les données qui seront insérées ont été lues en colonnes
t.grid_NA <- t(grid_NA)

##Créer un nouveau tableau qui sera rempli avec les valeurs
t.imageplot_stat_indicateur <- t.grid_NA

for (correction in correctionliste){

##On déclare la variable stat_indicateur pour faciliter l'usage
stat_indicateur <- eval(as.name(paste0("stat_indicateur_",correction)))

##Boucle qui remplie le tableau
##Si la case contenait une valeur initialement, on insère la médiane
##Sinon on vide la case
k=1
```

```
for (i in 1:nrow(t.grid_NA)){
for (j in 1:ncol(t.grid_NA)){
if (t.grid_NA[i,j]==FALSE){
t.imageplot_stat_indicateur[i,j] <- stat_indicateur[k]
k=k+1
}else{
t.imageplot_stat_indicateur[i,j] <- NA
}
}
}

##On retranspose pour avoir le tableau initale
imageplot_stat_indicateur <- t(t.imageplot_stat_indicateur)
assign(paste0("imageplot_stat_indicateur_",correction),imageplot_stat_indicateur,
       envir = globalenv())

} #correctionliste
} #definir.grille
##############################################################################
##Fonction pour faire la cartographie
##############################################################################

cartographie.indicateur <- function(region,periode,indicateur,statistique){

library(fields)

definir.nb.correction(periode)
definir.nb.sites(region)
charger.donnees(region,periode)
calcul.stat.indicateur(indicateur,statistique)
max_min_indicateur(indicateur,statistique)
definir.grille(region,periode)

statistique_text <- ifelse(identical(statistique,mean)==TRUE,"Moyenne",
ifelse(identical(statistique,median)==TRUE,"Médiane",
ifelse(identical(statistique,sd)==TRUE,"Ecart-type",
ifelse(identical(statistique,var)==TRUE,"Variance",
ifelse(identical(statistique,max)==TRUE,"Maximum",
ifelse(identical(statistique,min)==TRUE,"Minimum",""
))))))

for (correction in correctionliste){

imageplot_stat_indicateur <-  eval(as.name(paste0("imageplot_stat_indicateur_",
                                                  correction)))

image.plot(longitude,latitude,imageplot_stat_indicateur,main=correction,
```

```
zlim=c(min_indicateur,max_indicateur),xlab="longitude",ylab="latitude")
}

mtext(paste(statistique_text,indicateur,"-",region,"-",periode,sep=" "), outer=TRUE,
      cex=1, line=-1.15)

}
```

Script : MAIN.r (initialement Main_Compromise.r)

```
################################################################################
## R.script name: MAIN
################################################################################
#------NETTOYAGE ENVIRONNEMENT TRAVAIL-------------------------------------
rm(list=ls())
gc()

##SELECTION DE LA REGION
regionliste=c("Provence")     ##"Ile_de_France","Provence","Bretagne")

##SELECTION DE LA CULTURE
cultureliste=c("mais")                ##"vigne","mais","ble")

for (region in regionliste){
print(region)

for (culture in cultureliste){
print(culture)

#DEFINITION DU NOMBRE DE SITES
if(region=="Ile_de_France"){
nb_sites = 319
}

if(region=="Bretagne"){
nb_sites = 259
}

if(region=="Provence"){
nb_sites = 337
}

#------FICHIER PARAM-----------------------------------------------------------
repinitial <- c("C:/Users/Ling/Desktop/Stage INRAE/Dossier de travail")
param_Compromise_culture <- paste0(repinitial,"/Code/param_Compromise_",culture,".r")
source(param_Compromise_culture)

#------PACKAGES----------------------------------------------------------------
library(chron)      # annees bissextiles
# libraries pour faire des cartes
library(maptools)
library(RColorBrewer)
library(classInt)
library(scales)
library(devtools)
```

```
#------CALCUL--------------------------------------------------------------------
## Dans l ordre, on fait une boucle sur la variete, le modele climatique et la
##periode climatique
## Le calcul de la phenologie, du bilan hydrique et des indicateurs se fait donc a
#chaque fois pour (x sites* 1 plante * 1 var * 1 mod clim * 1 per clim)

plant <- plantliste

for (variety in varietyliste){   # boucle sur les varietes des plantes etudiees
print(variety)

for(correction in correctionliste){    # boucle sur les corrections de biais
print(correction)

for(k in 1:length(periode_interet)){       # boucle sur la periode d'interet
per = periode_interet[k]
print(per)

##------------- creation d un vecteur AN ----------------------------------------##
AN<-NULL; for(year in debut_periode[k]:fin_periode[k]){if(leap.year(year)==F)
  {AN<-c(AN,rep(year,365))}else{AN<-c(AN,rep(year,366))}}
## LECTURE fichier param_indecoclim_*** contenant (pour differentes varietes) les
##parametres relatifs a la plante (parametres pour indicateurs et pour calcul du
##bilan hydrique)
DATA_PARAM   <- as.data.frame(read.csv2(paste("fich_param/param_indecoclim_",plant,
                                             ".csv", sep=""),sep=";",dec=".",
                                      na.strings = "NA",header=T))
## LECTURE fichier indic_ecoclim avec les indicateurs ? faire tourner
#DATA_INDICATOR <- read.table(paste("fich_param/ind_ecoclim_",plant,".csv",sep=""),
#sep=";",na.strings = "NA",header=T)

##-------- CALCUL BILAN HYDRIQUE ------------------------------------------------##

if (calcul_SWB == 1){

##-------- CALCUL PHENOLOGIE POUR LE BILAN HYDRIQUE----------------------------##
## LECTURE fichier param_pheno_*** contenant (pour differentes varietes) les parametres
##pheno de la plante pour calculer le Bilan hydrique
PARAM_PHENO  <- read.table(paste("fich_param/param_pheno_BH_",plant,".csv",sep=""),
                           sep=";",na.strings = "NA",header=T)
PARAM_PHENO_variety=eval(parse(text=paste("PARAM_PHENO$",variety,"",sep="")))

#définir les périodes en fonction des corrections
if (correction != "SAFRAN"){
repdata <- paste0(repcompromise,"/BC_data_day_",correction,"_",per,"_",region,"/")
}
```

```
if (correction == "SAFRAN"){

repdata <- paste0(repcompromise,"/Ref_data_day_",correction,"_",per,"_",region,"/")
}

## lecture du fichier .RData avec toutes les variables climatiques
## lecture de tmoy
load(paste0(repdata,"tas_day_",correction,"_",per,"_",region,".RData"))

## Calcul de la pheno (obligatoirement 4 stades pour le calcul du bilan hydrique)
print("deb Calcul Pheno bilan hydrique")
source("Calcul_Pheno_Compromise.r")
print("fin Calcul Pheno bilan hydrique")

##-------- CALCUL BILAN HYDRIQUE-----------------------------------------------------------
## Ouverture du fichier de parametres sol
PARAM_SWB <- read.table("fich_param/param_soil_SWB.csv",na.strings="NA",sep=";",
                        header=T)

## lecture des pluies
load(paste0(repdata,"pr_day_",correction,"_",per,"_",region,".RData"))

#calcul ou lecture de ET0
if (calcul_ET0==1){

# lecture du rayonnement
load(paste0(repdata,"rsds_day_",correction,"_",per,"_",region,".RData"))

#lecture du vent
load(paste0(repdata,"sfcWind_day_",correction,"_",per,"_",region,".RData"))

#lecture de l'humidité
load(paste0(repdata,"hurs_day_",correction,"_",per,"_",region,".RData"))

print("debut calcul ET0")
source("calcul_ET0.r")
print("fin calcul ET0")

}

if(plant=="vigne") {nban=1}
## calcul du bilan hydrique
print("deb Calcul bilan hydrique")
source("soil_water_balance.r")
print("fin Calcul bilan hydrique")
if(plant=="vigne") {nban=2}
```

```
}
## ------ CALCUL INDICATEURS ------------------------------------------------##

##-------- CALCUL PHENOLOGIE POUR LES INDICATEURS-----------------------------------##

if (calcul_pheno_indic==1) {
## LECTURE fichier param_pheno_*** contenant (pour differentes varietes) les parametres
##pheno de la plante
PARAM_PHENO  <- read.table(paste("fich_param/param_pheno_indicateurs_",plant,".csv",
                                 sep=""),sep=";",na.strings = "NA",header=T)
PARAM_PHENO_variety=eval(parse(text=paste("PARAM_PHENO$",variety,"",sep="")))

## lecture de tmoy
load(paste0(repdata,"tas_day_",correction,"_",per,"_",region,".RData"))

## Calcul de la pheno
print("deb Calcul Pheno indicateurs")
source("Calcul_Pheno_Compromise.r")
print("fin Calcul Pheno indicateurs")
}

} # per
} # correctionliste
} # variety
} #regionlise
} #cultureliste
```

Sources

https://www.hydrol-earth-syst-sci-discuss.net/hess-2018-281/hess-2018-281.pdf
https://www.rdocumentation.org/packages/cffdrs/versions/1.7.1/topics/fwi
https://cwfis.cfs.nrcan.gc.ca/renseignements/sommaire/fwi
https://fr.wikipedia.org/wiki/Stades_de_croissance_des_c%C3%A9r%C3%A9ales
https://gwis.jrc.ec.europa.eu/about-gwis/technical-background/fire-danger-forecast/
https://inra-dam-front-resources-cdn.wedia-group.com/ressources/afile/393050-0383b-resource-metaprogramme-accaf-version-courte-2016-2020.pdf

Présentation d'INRAE:
https://www.inrae.fr/nous-connaitre
https://informatique-mia.inrae.fr/biosp/sites/informatique-mia.inra.fr.biosp-d7/files/Rapport_HCERES_BIOSP_2016.pdf#overlay-context

Cultures:
https://www.agro.basf.fr/fr/cultures/mais/basf_agro_et_la_filiere_mais/
https://vert-lavenir.com/ble/

IFM:
http://documents.irevues.inist.fr/bitstream/handle/2042/18163/meteo_2007_56_22.pdf