

# Améliorer l'usage de nos modèles statistiques

## Gestion de versions et partage de l'expérience Ines

Mathias ANDRÉ

Division Redistribution et politiques sociales (D2E - Dese)

Séminaire SSP Lab – 5 juin 2018

# Plan de la présentation

## Aperçu rapide de la gestion de versions

- Des avantages incontestables

- Un peu de GIT

## L'exemple de l'équipe Ines

- Fonctionnement global

- Usage quotidien

## L'ouverture d'Ines sous licence libre

# Plan de la présentation

## Aperçu rapide de la gestion de versions

Des avantages incontestables

Un peu de GIT

## L'exemple de l'équipe Ines

Fonctionnement global

Usage quotidien

## L'ouverture d'Ines sous licence libre

# Pourquoi utiliser la gestion de versions

Une gestion efficace de l'ensemble des programmes d'un modèle :

- ▶ **conserver le modèle** de façon sécurisée et **archiver le code**, historique compris
- ▶ **collaborer simplement** et avec méthode :
  - ▶ travailler en parallèle et fusionner facilement du code
  - ▶ partager une documentation des programmes (commentaires des modifications)
  - ▶ trouver rapidement des erreurs et en diffuser rapidement la correction
- ▶ **améliorer la qualité du modèle** par des méta-données riches
- ▶ **permettre sa diffusion simplifiée** et harmonisée

Le meilleur moyen de partager du code (équipes internes, changements de poste, archivage, ouverture au libre, etc.).

# Des méthodes efficaces de programmation collaborative

Comparaison avec ou sans gestion de versions :

	<b>Avant</b>	<b>Gestion de versions</b>
<b>Archivage</b>	à la main	automatique
<b>Envoi de modification</b>	mail, dossier partagé	code partagé
<b>Fusion de code</b>	copié-collé	simple, sûr
<b>Versions du modèle</b>	dossiers archivés	historique riche
<b>Gestion des erreurs</b>	à la main	méthodique

- ▶ Le temps est consacré à la programmation utile :
  - ▶ Amélioration rapide du modèle
  - ▶ Partage efficace des tâches
  - ▶ Recherche puissante des erreurs
  - ▶ Archivage géré automatiquement avec méta-données
- ▶ Des outils généralisés et documentés (aide en ligne).
- ▶ Les deux logiciels les plus répandus sont GIT et SVN.

GIT est *open source*, gratuit et répandu : <http://try.github.io/>

# Plan de la présentation

## Aperçu rapide de la gestion de versions

Des avantages incontestables

Un peu de GIT

## L'exemple de l'équipe Ines

Fonctionnement global

Usage quotidien

## L'ouverture d'Ines sous licence libre

# Les fonctionnalités de base de GIT

Quelques principes pour comprendre

Les programmes s'organisent sur un répertoire **distant** commun (*repository*) et sont manipulés sur les postes individuels dans des dossiers **locaux** (*working directory*).

Les trois commandes principales :

- ▶ `diff` : fichiers modifiés, ajoutés ou supprimés
- ▶ `staging area` puis `commit` : sélection et envoi des modifications
- ▶ `push` et `pull` : mise à jour du local par rapport au distant (et réciproquement)

<https://romain-warnan.github.io/git-au-quotidien>

# Plan de la présentation

## Aperçu rapide de la gestion de versions

Des avantages incontestables

Un peu de GIT

## L'exemple de l'équipe Ines

Fonctionnement global

Usage quotidien

## L'ouverture d'Ines sous licence libre

# Architecture du projet Ines

- ▶ Un dépôt regroupant programmes et historique des commits, paramètres, archives et documentation (wiki et documents)
- ▶ Un projet local sur chaque ordinateur personnel mis à jour en temps réel

Nom	Modifié le	Type
 1Preparation	17/05/2018 14:25	Dossier de fichiers
 2Imputations	17/05/2018 14:26	Dossier de fichiers
 3Modele	17/05/2018 14:28	Dossier de fichiers
 4Basemen	27/03/2018 11:21	Dossier de fichiers
 5Sorties	27/03/2018 11:21	Dossier de fichiers

- ▶ Versions courantes instantanées vs modèles figés chaque année (exemple : « Ines 2017 » )

# Plan de la présentation

## Aperçu rapide de la gestion de versions

- Des avantages incontestables

- Un peu de GIT

## L'exemple de l'équipe Ines

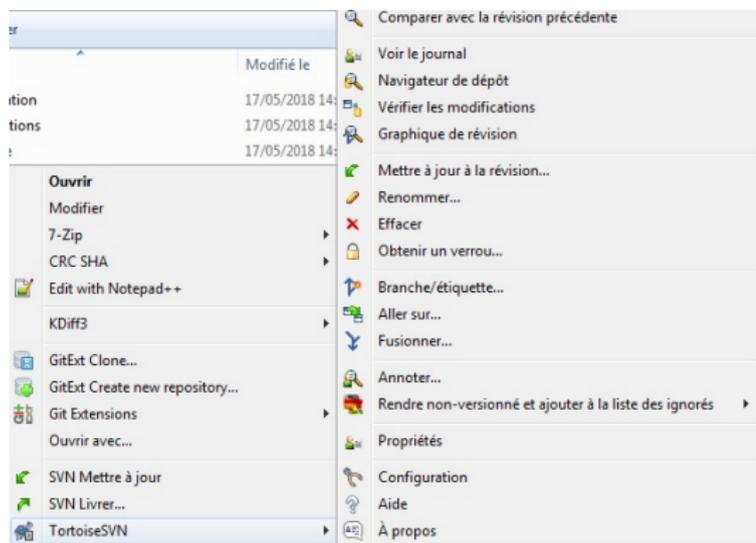
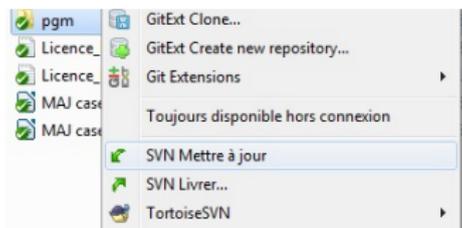
- Fonctionnement global

- Usage quotidien

## L'ouverture d'Ines sous licence libre

# Exemples d'usages quotidiens

## Menu contextuel (clic droit)



# Exemples d'usages quotidiens

Modifier un programme et le mettre en commun : `commit`

The screenshot shows the TortoiseSVN 'Commit' dialog box. The window title is 'Z:\Tortines\pgm\3Modele\Fisc - Livrer - TortoiseSVN'. The 'Livrer vers' field contains the URL: `svn+ssh://proxy-depot-distant.insee.fr/scmrepos/svn/ines/trunk/pgm/3Modele/Fisc`. The 'Commentaire' field has a button for 'Commentaires récents' and a text area containing the commit message: `[majeur] modification calcul de la PPE et suppression du programme ppe.sas`. Below this is a section for 'Changements effectués (double-cliquez pour voir les différences) :'. It features a filter bar with options: 'Vérifier: Tous', 'Aucun', 'non versionné', 'Versionné', 'Ajouté', 'Effacé', 'Modifié', 'Fichiers', and 'Répertoire'. A table lists the files to be committed:

Chemin	Extension	Statut	Statut de la propriété	Verrou
<input checked="" type="checkbox"/> 6_Impot.sas	.sas	modifié		
<input checked="" type="checkbox"/> ppe.sas	.sas	manquant		

At the bottom of the dialog, there are checkboxes for 'Voir les fichiers non versionnés' (checked) and 'Afficher les références externes d'autres dépôts' (checked). The status bar indicates '2 fichiers sélectionnés, 2 fichiers au total'. At the very bottom, there are buttons for 'Garder les verrous', 'Conserver les listes de modifications', 'Voir le journal', 'OK', 'Annuler', and 'Aide'.

# Exemples d'usages quodtiens

Consulter l'historique d'un fichier ou dossier

Z:\Tortines\pgm\3Modele\Fisc\6\_Impot.sas - Commentaires - TortoiseSVN

Filter Commentaires, Chemins, Auteu

Révision	Actions	Auteur	Date	Commentaire
<b>4179</b>		<b>al.biotteau</b>	<b>vendredi 8 septembre 2017 12:29:24</b>	<b>[Debug] Modification du calcul de</b>
4021		antoinesir	mardi 27 juin 2017 14:43:18	[mineur] dans la macro décote, il y a un
3888		al.biotteau	jeudi 18 mai 2017 16:53:56	[MàJ] Code de la modification du plafon
3744		al.biotteau	mercredi 5 avril 2017 18:28:06	[MàJ][Debug] Suite aux commits 3655 e
3730		al.biotteau	lundi 3 avril 2017 18:35:11	[MàJ][Debug] Vérification d'un taux sur
3696		al.biotteau	mardi 21 mars 2017 18:53:33	[MàJ] Suite vérification mise à jour de p
3626		simenf	vendredi 3 mars 2017 10:20:25	MàJ init. feuille Examen des cases décla

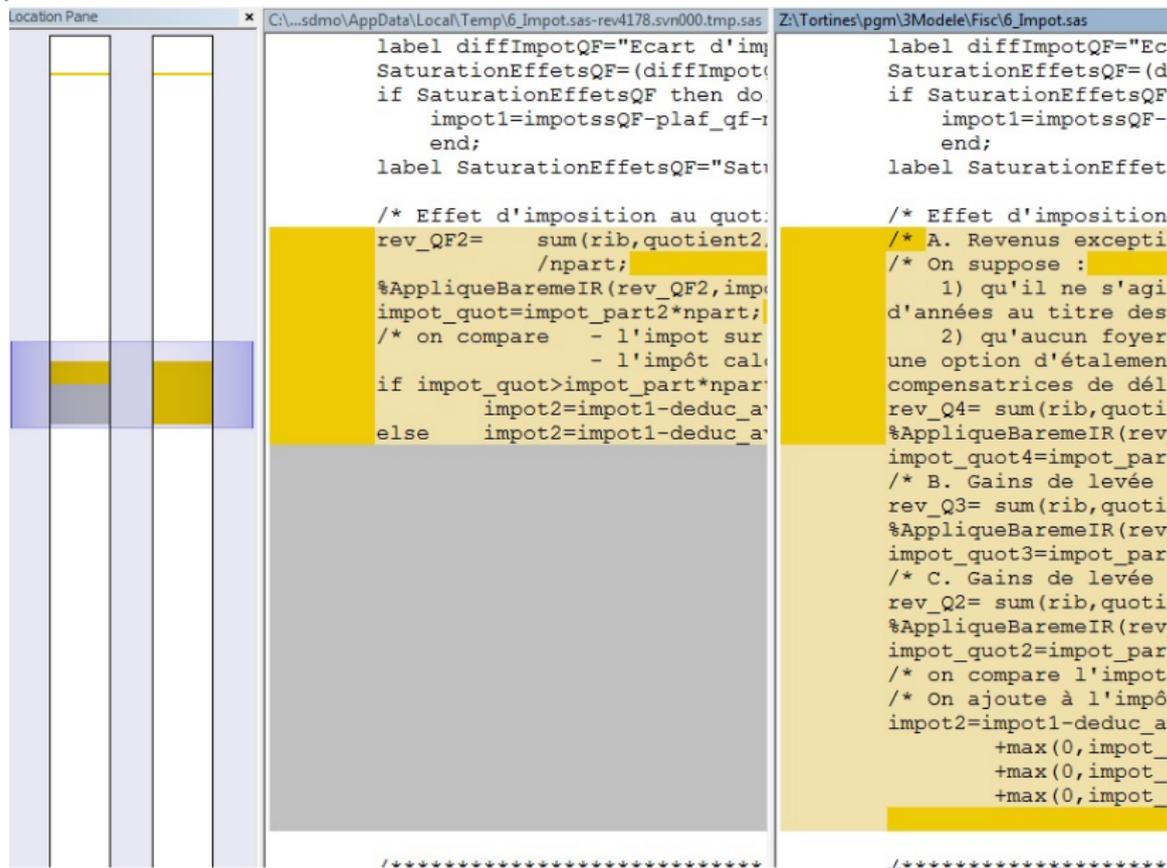
[Debug] Modification du calcul de l'imposition au quotient. Auparavant, on calculait plutôt 1, 4, 3 ou 2 la différence d'impôt avec ou sans ces revenus supplémentaires imposés au quotient. levée d'option pour lesquels le coef = 2 ou 3. On calcule bien désormais l'impôt avec 1/4, 1, en compte de ces revenus particuliers. J'ajoute un commentaire sur les revenus exceptionnels, dûs au titre de 3 années car pour ces revenus différés, le coef = Nombre d'années au titre de

=> Pour 2014/2016 : +0,2 % d'IR (124 millions d'€); négligeable sur les autres agrégats (CSG

Chemin	Action	Copier depuis le chemin	Révision
/trunk/pgm/3Modele/Fisc/6_Impot.sas	Modifié		

# Exemples d'usages quotidiens

## Comparer des révisions



```
Location Pane x C:\...sdmo\AppData\Local\Temp\6_Impot.sas-rev4178.svn000.tmp.sas Z:\Tortines\pgm\3Modele\Fisc\6_Impot.sas

label diffImpotQF="Ecart d'impo
SaturationEffetsQF=(diffImpot
if SaturationEffetsQF then do
    impot1=impotssQF-plaf_qf-1
end;
label SaturationEffetsQF="Sati

/* Effet d'imposition au quot:
rev_QF2= sum(rib,quotient2,
/npart;
%AppliqueBaremeIR(rev_QF2,impot
impot_quot=impot_part2*npart;
/* on compare - 1'impot sur
- 1'impôt calc
if impot_quot>impot_part*npart
    impot2=impot1-deduc_a
else impot2=impot1-deduc_a

label diffImpotQF="Ec
SaturationEffetsQF=(d
if SaturationEffetsQF
    impot1=impotssQF-
end;
label SaturationEffet

/* Effet d'imposition
/* A. Revenus excepti
/* On suppose :
1) qu'il ne s'agi
d'années au titre des
2) qu'aucun foyer
une option d'étalemen
compensatrices de dél
rev_Q4= sum(rib,quoti
%AppliqueBaremeIR(rev
impot_quot4=impot_par
/* B. Gains de levée
rev_Q3= sum(rib,quoti
%AppliqueBaremeIR(rev
impot_quot3=impot_par
/* C. Gains de levée
rev_Q2= sum(rib,quoti
%AppliqueBaremeIR(rev
impot_quot2=impot_par
/* on compare l'impot
/* On ajoute à l'impô
impot2=impot1-deduc_a
+max(0,impot_
+max(0,impot_
+max(0,impot_
```

# Exemples d'usages quotidiens

## Recherche une erreur effacement : blame

```
-----
-----
-----
46
47 %macro AppliqueBaremeIR(Rev, MontantImpot, TxMarg=txmarg);
48 /* @Rev : revenu à imposer
49 @MontantImpot : Montant d'impôt calculé par part
50 @TxMarg : valeur du taux de la bonne tranche d'impositio
51 &MontantImpot.=0;
52 &TxMarg.=0;
53 %do i=2 %to %eval(&nbtranche.-1);
54 %let j = %eval(&i.-1);
55 &MontantImpot.=%MontantImpot.+&tx&i.*max(min(&Rev.-&pl
56 if &Rev.>&plaf&j. then &TxMarg.=%&tx&i.;
57 %end;
58 %let l=%eval(&nbtranche.-1);
59 &MontantImpot.=%MontantImpot.+&tx&nbtranche.*max((&Rev.-&pl
60 %mend AppliqueBaremeIR;
61
62 %macro Decote(impot);
63 /* Cette macro crée DECOTE, calculée sur le montant d'impôt :
64 decote=0;
65 %if &anleg.>=2016 %then %do;
66 if mcdvo in ('C','D','V') & &impot.<&decote. then dec
67 if mcdvo in ('M','O') & &impot.<&decote_couple. then
68 %end;
69 %if &anleg.=2015 %then %do;
70 if mcdvo in ('C','D','V') & &impot.<&decote. then dec
71 if mcdvo in ('M','O') & &impot.<&decote_couple. then
72 %end;
73 %if 2001<=&anleg. & &anleg.<2015 %then %do;
74 if &impot. <= 2*&decote. then decote=&decote -&impot./2;
```

03/05/2016 15:42:53

[MàJ] tant qu'à y être, j'ai programmé la nouvelle décote pour 2015 (avec le paramètre correspondant à param\_prelev), c'est trois lignes de code qu'on déboguera si nécessaire une fois qu'on pourra faire tourner leg 15 base 13. Pas d'erreur sur leg 14 base 12 (mais c'est pas surprenant, il y a un if anleg < 2015).

## D'autres fonctionnalités utiles

- ▶ Création de branches pour développer de longs projets, en parallèle, sans perturber le dépôt commun : `git branch`
- ▶ Fusionner des projets (branches) : `git merge`
- ▶ Conserver des fichiers en privé : `git ignore`
- ▶ Sauvegarder une version figée : `git tag`

En savoir plus : <https://git-scm.com/>.

# L'ouverture d'Ines sous licence libre

## Étape par étape

1. Préparer en amont :
  - ▶ partir d'une version « initiale » stable du modèle
  - ▶ élaborer une documentation précise (« formation nouveaux arrivants »)
2. Choisir les caractéristiques :
  - ▶ logiciel de version (GIT ou SVN)
  - ▶ plateforme de diffusion (GitLab, GitHub, Forge Insee)
  - ▶ forum, wiki, licences
3. Transposer les outils :
  - ▶ de la documentation au wiki collaboratif
  - ▶ du code individuel au dépôt partagé puis au dépôt ouvert

# Perspectives

## Généraliser la gestion de versions

- ▶ L'**utilisation individuelle** de la gestion de versions est possible et souhaitable (recommandée ?) : gestion des erreurs, archivage, changement de poste, gain d'espace, etc.
- ▶ GIT est **directement intégré** dans l'interface RStudio :  
`https://git.stable.innovation.insee.eu/formation/r\_dev\_insee`
- ▶ À l'Insee, installation du Git Pack sur SI@moi ou en portable sur AUS, dépôt sur la gforge ou la plateforme innovation.

## Partager les bonnes pratiques

- ▶ Harmoniser les programmes : cf. « charte du bon codeur » et audit de l'unité qualité.
- ▶ Ne pas oublier d'autres outils pour une programmation efficace : Winmerge, Grepwin, Atom, etc.

# Merci pour votre attention !

 Forge du modèle Ines (code, wiki, documentation, etc.)

*Accès via la plateforme Adullact :*

<https://adullact.net/projects/ines-libre>