Du code dans ma thèse

MetSem – 19 septembre 2019

Florence Ecormier-Nocca, florence.ecormiernocca@sciencespo.fr

Centre d'Études Européennes et de Politique Comparée, Sciences Po

Outline

1. Présentation de la thèse
2. Ma boîte à outils
3. Statut du code dans ma thèse
4. Coder en sciences sociales : difficultés et idées reçues

Présentation de la thèse

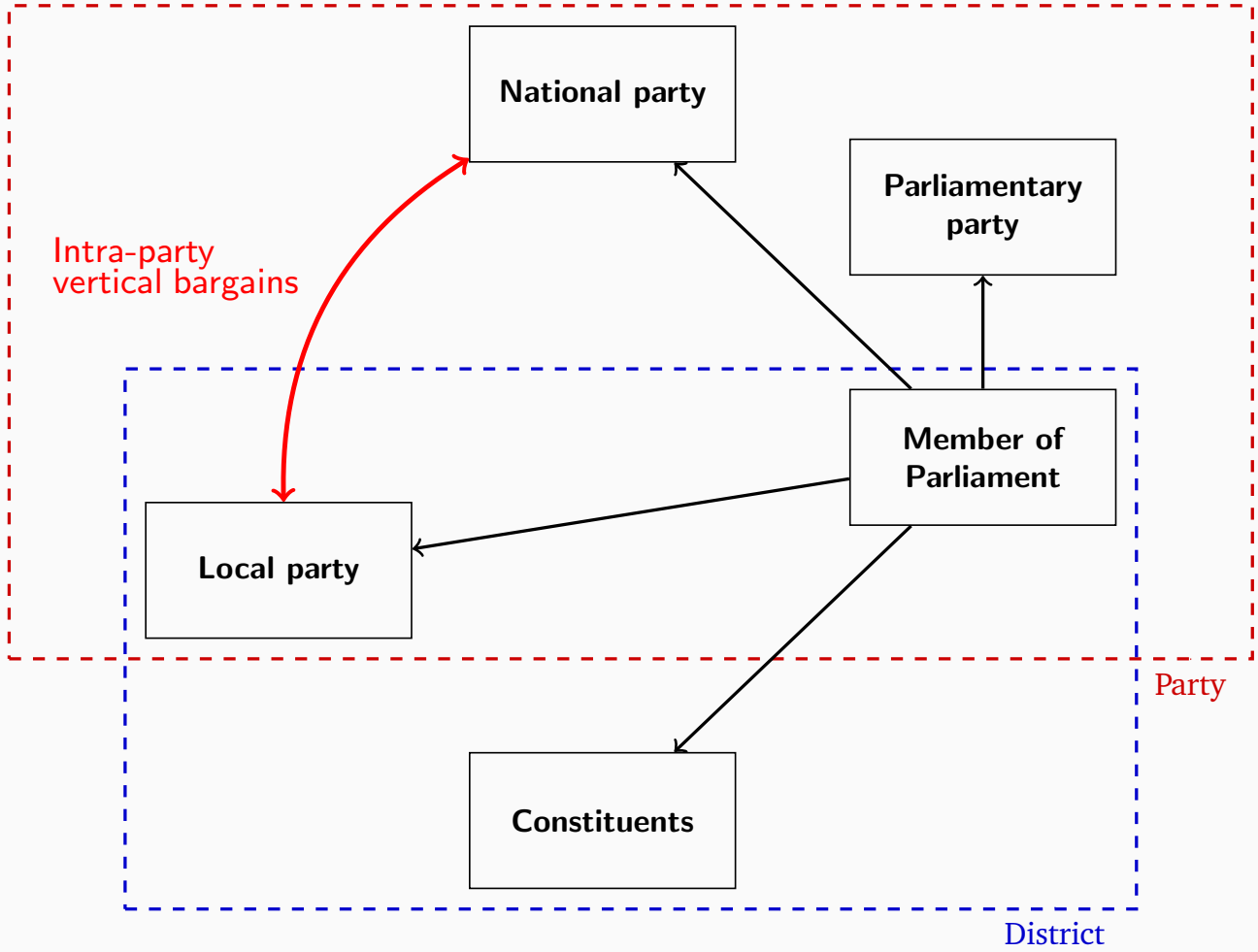


Figure 1: Un agent de représentation pour plusieurs principaux

En bref...

- Mesurer et expliquer la diversité idéologique au sein des partis politiques (IPD) : pourquoi certains partis sont-ils systématiquement plus cohérents que d'autres ?
- Définition de IPD au niveau individuel : déviation idéologique d'un député par rapport à l'organisation nationale de son parti et par rapport à ses pairs
- Données : sondages d'experts (en Europe) et tweets produits par les députés (en France, Espagne et Royaume-Uni)
- Méthodologie : analyse de texte, machine learning

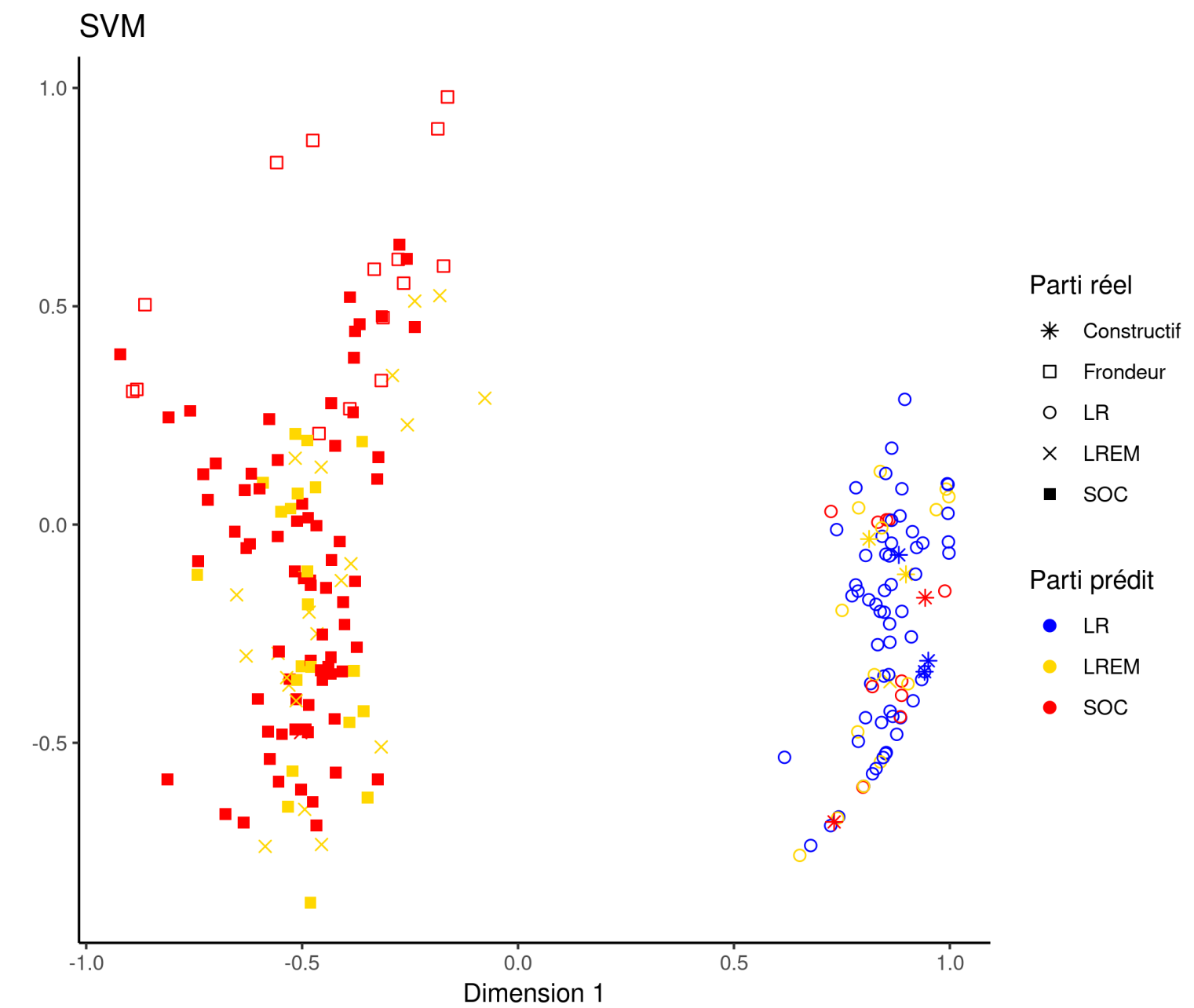


Figure 2: Les députés français au Parlement et sur Twitter (ML)

Source : Ecormier-Nocca et Louis-Sidois 2019

Ma boîte à outils

Utilisation du code dans ma thèse

- Point de départ méthodologique : Internet comme source de données
- Deux objectifs principaux du code :
  - Recueil des données
  - Traitement des données (données textuelles et numériques, nettoyage, analyse, modèles, visualisation)

Formation

- Formation autodidacte (mémoire de Master : Python et R)
- Financement “double culture” de l’USPC, cours de machine learning à Descartes
- Ressources : manuels et vignettes de paquets R, Stackoverflow...
- Apprentissage principalement par copie et adaptation de codes existants
- Formation quotidienne et sans cesse à renouveler (nouveaux outils, amélioration des pratiques...)

Mon “setup”

Logique générale :

- Privilégier les logiciels libres
- Compatibilité Debian (distribution GNU+Linux)
- Communauté développée en stats, ML mais aussi en sciences sociales (documentation, mise à jour des paquets...)
- Limiter la multiplication des outils
- “Path dependency”

Mon “setup”

- Bash et Python (récupération des comptes, scraping)
- R (recueil des tweets, analyses statistiques)
- LaTeX pour le texte
- Un éditeur de texte commun à tous les langages : Emacs
- Serveur Rstudio pour récupérer les données en temps réél et les stocker, fourni par HumaNum SHS
- Pour sauvegarder le code et le texte et gérer les différentes versions : Git, en utilisant l’interface web GitLab, équivalent de GitHub (racheté par Microsoft en 2018)

Statut du code dans ma thèse

Une partie intégrante de la thèse

Beaucoup de similitudes avec la rédaction

- Un script = une grande tâche
- Organisation du script en sous-parties correspondant à de plus petites tâches
- Scripts numérotés et nommés, rangés dans des dossiers correspondant à différentes parties de la thèse
- Rédaction en plusieurs phases : brouillon, passages copiés-collés d'un script à l'autre, avant une version propre, commentée, organisée en sections
- Même éditeur de texte, passages directs de l'un à l'autre (pour les tableaux de régressions par exemple)
- Des "codes" de présentation (cf slide suivante)

```r
## 1_create_db.r removes wrongly assigned candidates and
    duplicates
rm(list = ls(all = TRUE))
setwd("~/data-analysis/datafiles/tweets")
Sys.setlocale("LC_TIME", "C") # months and days in English

## Packages
library(quanteda)
library(dplyr)
...

## Load data
tweets = read.csv(paste0(country,"_mps.tweets.csv"), header =
    TRUE, stringsAsFactors = FALSE)

## --- Remove duplicates ---
rm_duplicate = function(filename){
  ...
}
...

## Write as csv datafiles
write.csv(tweets, paste0("all_",country,"_mps.tweets.csv"),
row.names = FALSE)
```

Statuts des scripts

- Une partie intégrante de ma thèse : code hébergé sur GitLab, accès libre à la fin de ma thèse
- Les rendre les plus généraux possibles
- Les rendre facile de réutilisation pour le futur
- Exemple : un script pour 3 pays (voir slide suivante)

```
if(country == "spanish"){
    stopwords_list = unique(c(stopwords("ca","stopwords-iso"),
                              stopwords("es","stopwords-iso")
     ...))
} else if(country == "french"){
    stopwords_list = unique(c(stopwords("fr","stopwords-iso"),
    stopwords("fr","snowball")))
} else{
    mystopwords = stopwords("en","snowball")
}

mystopwords = unique(c(stopwords_list, iconv(stopwords_list,
              to='ASCII//TRANSLIT')))

toks = tokens(twCorpus, what = "word",
              remove_punct = TRUE, remove_twitter = FALSE,
              include_docvars = TRUE) %>%
              tokens_remove(mystopwords, padding = TRUE)
```

Coder en sciences sociales : difficultés et idées reçues

Du côté informatique...

- Des problèmes de terminologie (ex. VD/VI v. Y/X)...
- ... mais aussi d'objectifs (expliquer v. prédire)
- Des données "inhabituelles" : déterminer quand écouter et quand se faire confiance

Du côté science politique...

- Une confusion sur le statut de l'interdisciplinarité
- Une méthodologie qui prend le pas sur la théorie

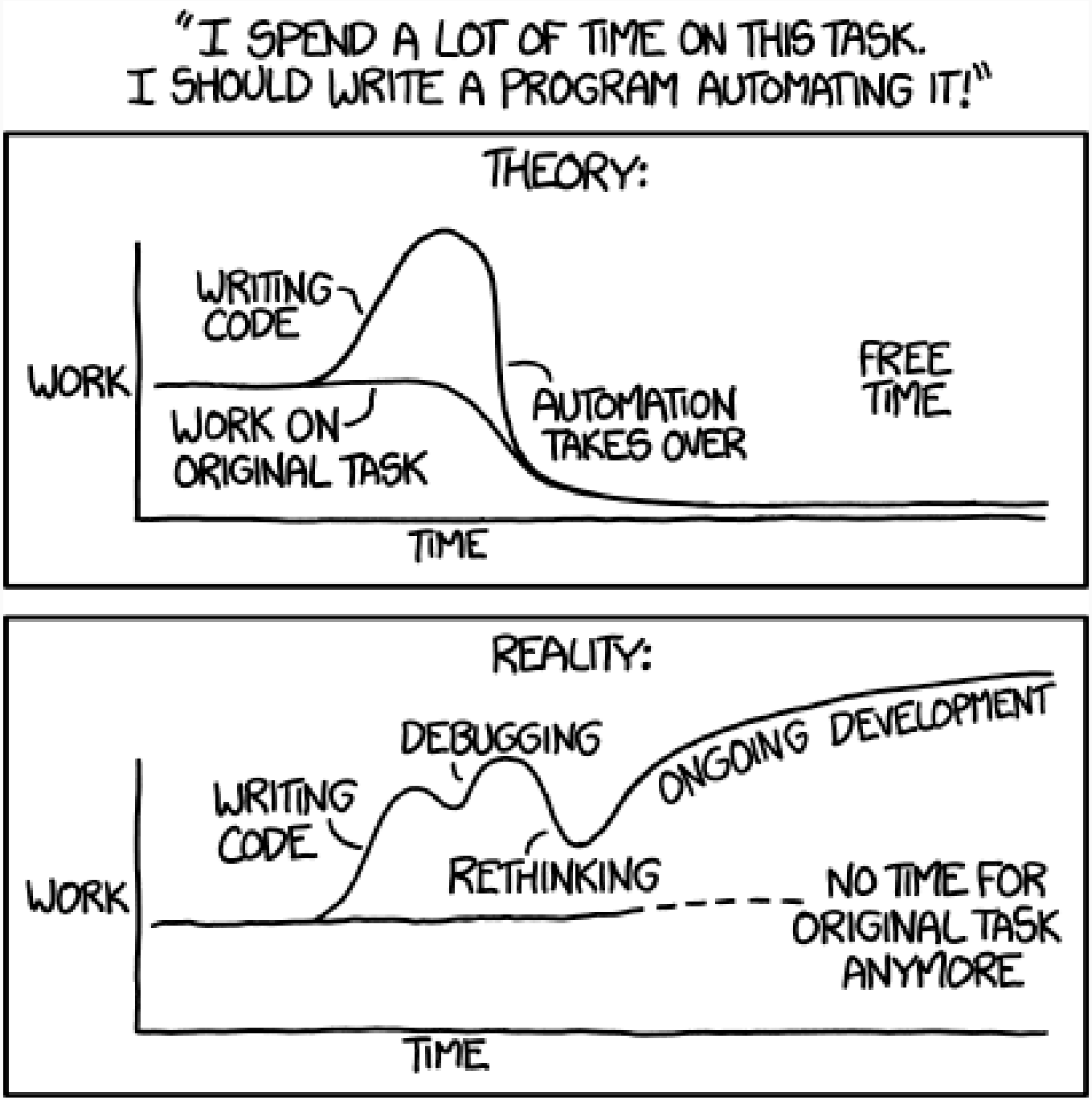
$\rightarrow$ Un enjeu de présentation

Deux extrêmes...

- “Ne t’embête pas à m’expliquer, c’est trop compliqué pour moi !”

- “Il suffit d’appuyer sur un bouton !”
- “Ta thèse ça doit aller vite, tu n’as pas de terrain !”
- “Tu peux récupérer et analyser des tweets pour moi ?”

$\rightarrow$ Propre à la France ?



Source : xkcd, Automation, https://xkcd.com/1319/

Conclusion : le code, un outil parmi d'autres ?

- Tous codeurs ?
- La différence : la place qu'on lui donne dans son travail
- Une responsabilité pédagogique de "démystification" du code, tout en en montrant les difficultés
- Finalement similaire à toute autre démarche de recherche

Du code dans ma thèse

MetSem – 19 septembre 2019

Florence Ecormier-Nocca, florence.ecormiernocca@sciencespo.fr

Centre d'Études Européennes et de Politique Comparée, Sciences Po