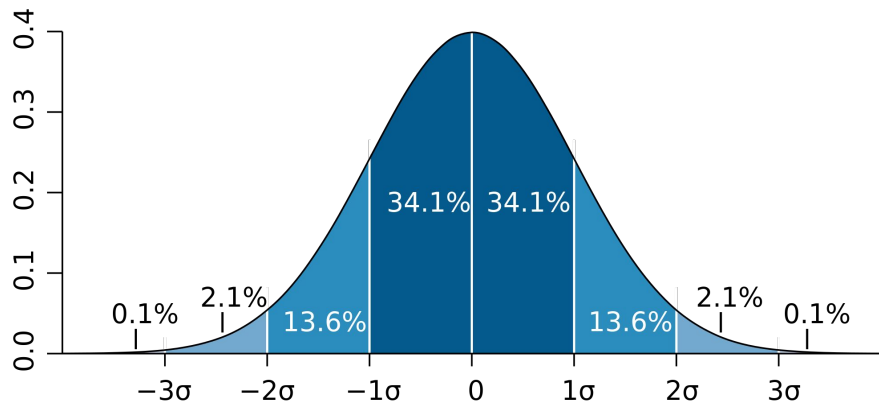




5.2 : Analyse de données

Amine Benhenni : albenhenni@gmail.com

L'aspect scientifique



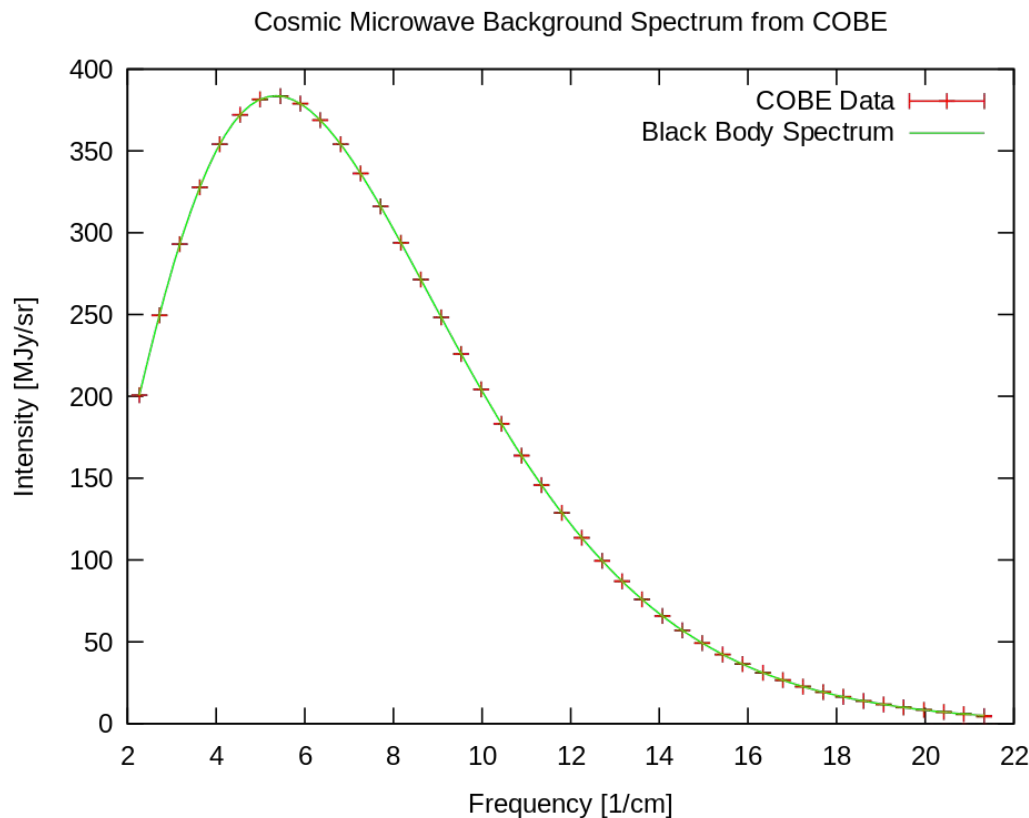
Comprendre les outils mathématiques et statistiques

Se familiariser au travail avec les données

Extraire le signal du bruit

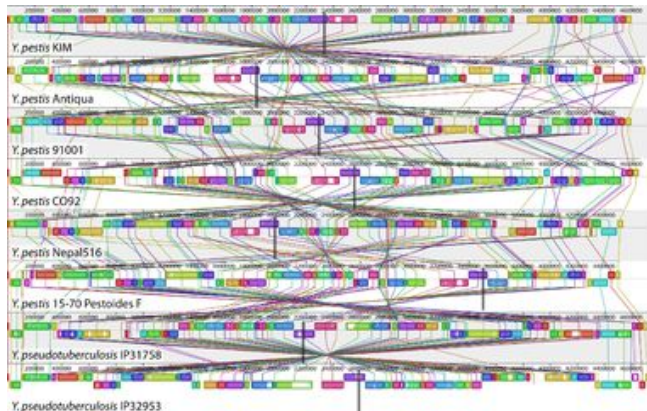
Extraire des informations spécifiques pour caractériser les données (moyenne, écart-type, ...)

Fit de modèles

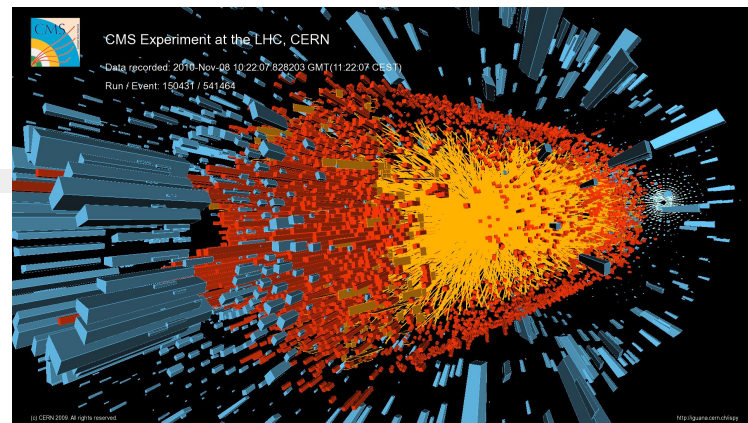


L'exception : un fit
quasi-parfait.

La règle : beaucoup de bruit
et d'incertitudes



Presented clip



Clip reconstructed
from brain activity



<https://www.youtube.com/watch?v=nsjDnYxJ0bo>

L'aspect informatique



Comprendre ce qu'offre python pour le traitement de données

Être à l'aise avec les principales bibliothèques scientifiques

Savoir penser et structurer son code

Développer la pensée algorithmique (exemple : calculer une moyenne)

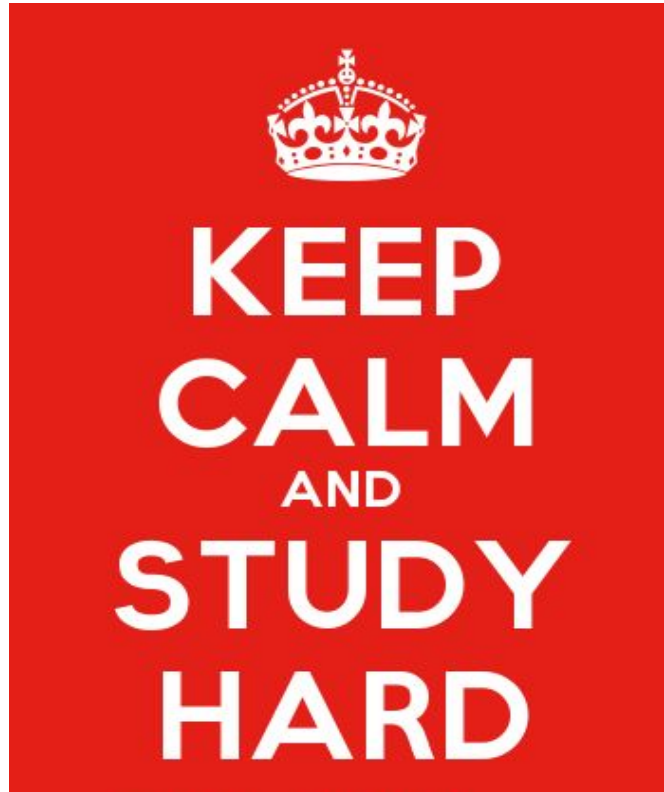
Pourquoi Python ?

Pourquoi Python ?

- **Généraliste** : nombreux domaines d'application (calcul scientifique, administration système, développement web...)
- **Multi paradigme** : un paradigme est un ensemble de règles grammaticales et d'outils permettant au développeur de décrire des algorithmes. Exemples de paradigmes applicables à Python : impératif (la structure du code est découpée en procédures, souvent appelées fonctions, qui peuvent s'appeler entre elles), fonctionnel (imbrication de fonctions), orienté objet.
- **Langage de haut niveau** : gestion mémoire automatique, contrairement au langage C.
- Très grande bibliothèque standard dont plusieurs **bibliothèques scientifiques** : SciPy, NumPy, Matplotlib... C'est un langage qui peut interagir avec d'autres langages, avec des bases de données (SQL), des fichiers de données (documents ascii, cvs, xml, images...).
- Syntaxe orientée sur la **lisibilité du code** : claire, aérée, concise.
- **Plusieurs contextes d'utilisation** : Interface interactive (shell) > scripts > programmes > bibliothèques (modules).
- Gratuit, open-source avec une grande communauté (beaucoup de documentation en ligne anglophone ou francophone (<http://python.developpez.com/>, <http://www.afpy.org/>, <http://stackoverflow.com/>)).

Infos pratiques

Modalités de contrôle des connaissances



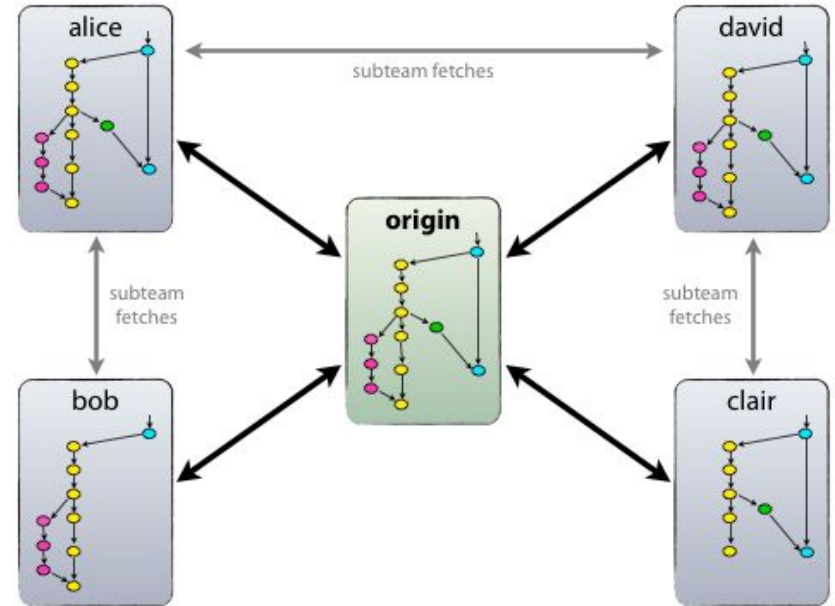
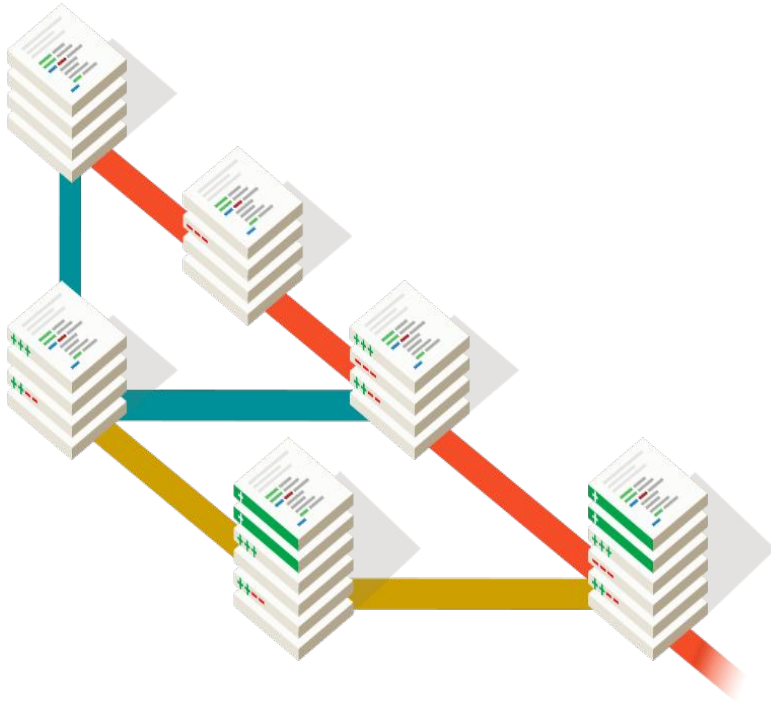
- Examen de 1h30 vers mi-novembre (coeff 3)
- Contrôles continus **tout le long du module** (coeff 4)
- Présentation des projets (coeff 3) en **décembre**

A propos des projets

- Comprendre un sujet
- Illustrer le sujet avec un **code de démonstration disponible sur github** pour les camarades
- **Courte présentation (20')** pour expliquer les concepts et le cas d'illustration
- Gestion de projet pour le travail en équipe (**note de CC**)

Sujets connexes

Gestion du code : versioning



Gestion de projet : méthodes agile / Kanban



Pour la prochaine séance

- Avoir un compte <http://github.com> et un compte <http://trello.com>
- Avoir lu les notebooks dans le répertoire S1/ du dépôt :
https://github.com/bendaizer/Villebon_5.2
- Faire les exercices 1 et 2 de ProjectEuler : <https://projecteuler.net/archives>
- S'assurer de pouvoir répondre aux questions suivantes :
 - ◆ Comment définit-on une fonction ?
 - ◆ Quelle est la différence entre une liste, un tuple, un set, un dictionnaire ?
 - ◆ Qu'est-ce que numpy et un numpy array ?
 - ◆ Comment est défini un vecteur dans numpy ? une matrice ?