ARITHMÉTIQUE ET STATISTIQUES AVEC MathsOntologie

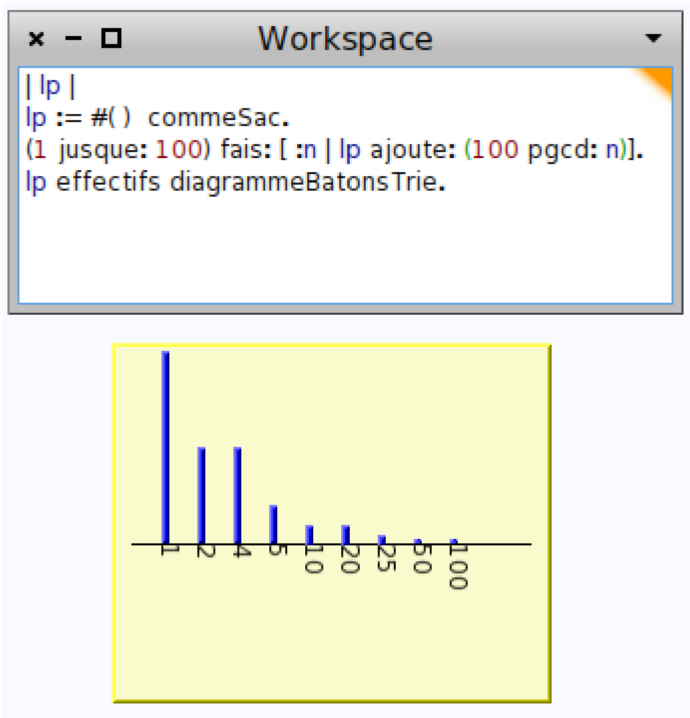
Comme MathsOntologie est un peu spécialisé en arithmétique, cet article montre comment on peut mettre en œuvre les outils de statistique de MathsOntologie pour faire de l'arithmétique.

I/ Statistiques sur les pgcd

Une liste très simple à obtenir est celle des entiers inférieurs à 100. C'est d'ailleurs aussi la liste des entiers modulo 100. Et bien on peut faire des statistiques sur cette liste... Dans cette partie, le caractère dont on va étudier la répartition statistique est le pgcd avec 100. Autrement dit, on va créer la liste des pgcd des entiers inférieurs à 100 avec 100 lui-même, et faire des statistiques sur cette liste.

1) La liste des pgcd

Comme un pgcd peut apparaître plusieurs fois, on va utiliser un multi-ensemble, ou sac, pour y stocker les pgcd. Pour ce faire, on crée un tableau vide, on le convertit en sac et on affecte la variable lp[1] avec ce sac vide. Ensuite, pour l'indice n allant de 1 à 100, on ajoute dans le sac lp, le pgcd de n et 100 :



2) diagramme en bâtons

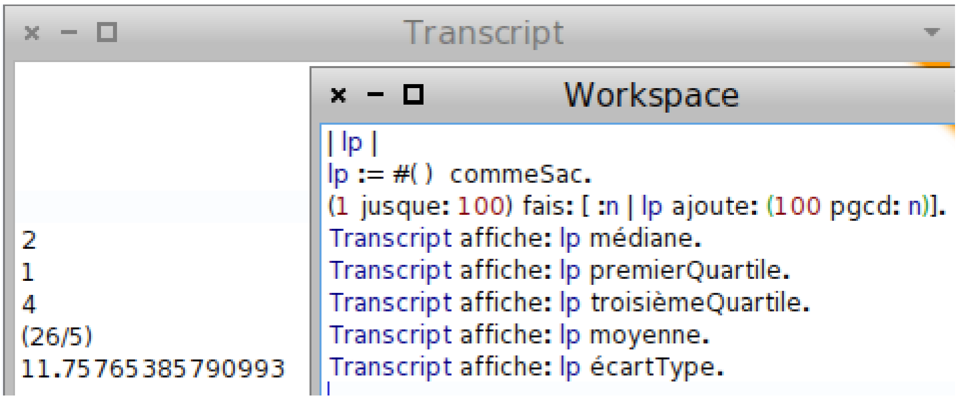
On voit ci-dessus que pour avoir le diagramme en bâtons, on envoie à lp le message « effectifs » qui répond le tableau d'effectifs, et à ce dernier, le message « diagrammeBatonsTrié » qui dessine le diagramme en bâtons après avoir trié dans l'ordre croissant les valeurs du caractère.

1 Comme « liste des pgcd »...

On constate que les valeurs du caractère sont les diviseurs de 100[2] et que le pgcd le plus fréquent est 1[3].

3) médiane et quartiles

Pour avoir la médiane d'un sac, on lui envoie le message « médiane » ; de même, les messages « premierQuartile » et « troisièmeQuartile » donnent les autres quartiles (ci-dessous, respectivement 1 et 4) :



4) moyenne et écart-type

On voit dans l'exemple ci-dessus que les messages « moyenne » et « écartType », envoyés à un tableau d'effectifs[4], donnent respectivement la moyenne (autant que possible sous forme d'une fraction irréductible) et l'écart-type de ce tableau. Ainsi, le pgcd moyen avec 100 est 5,2[5] alors que le pgcd médian est 2 qui est nettement plus petit[6].

---

2 Que l'on peut obtenir avec ***100 diviseurs***

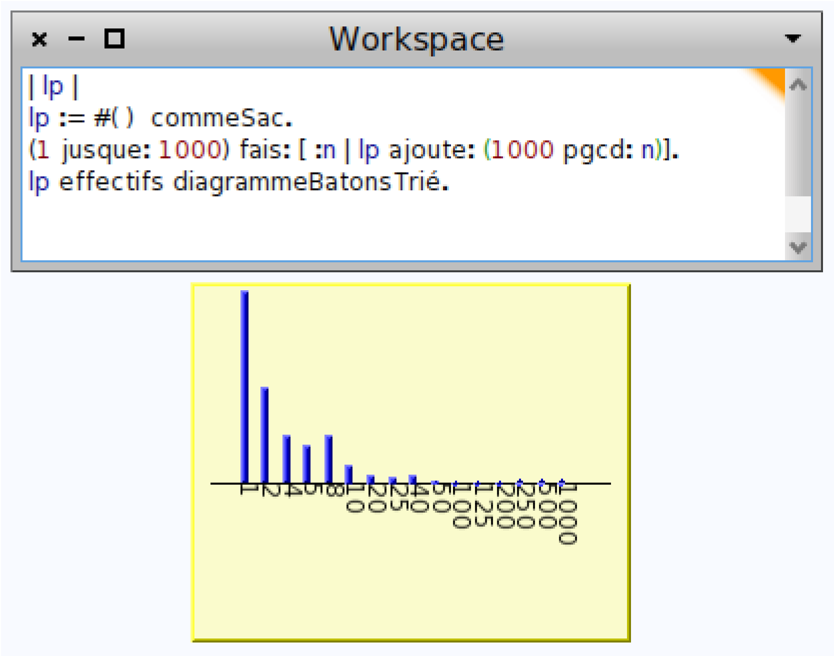
3 On dit qu'il y a beaucoup d'entiers qui sont premiers avec 100. Leur nombre est établi plus bas.

4 Le tableau d'effectifs est implémenté sous forme d'un « dictionnaire » comme ceux de Python : À chaque valeur du caractère est associé un nombre entier, son effectif. Les sacs sont d'ailleurs implémentés de façon similaire.

5 On constate que le pgcd moyen n'est même pas un entier, donc *a fortiori* pas un pgcd. Cela rappelle les remarques d'Adolphe Quételet sur le français moyen.

6 La moyenne est même supérieure au troisième quartile.

Un phénomène analogue apparaît avec les pgcd des nombres de 1 à 1000, avec 1000 :

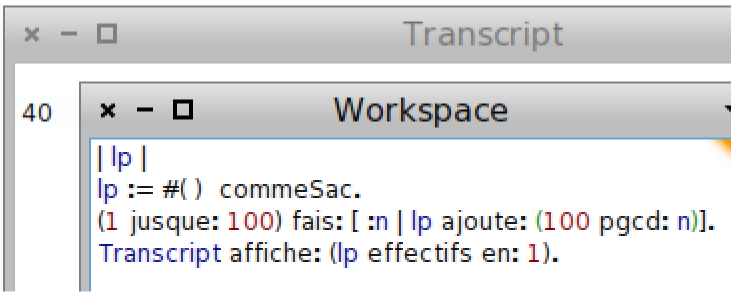


II/ L'indicatrice d'Euler

Maintenant on va s'intéresser au plus grand effectif : Le nombre de nombres premiers avec 100. C'est l'effectif du nombre 1.

1) Obtention d'un effectif

Le tableau d'effectifs est un « dictionnaire », qui, à chacune de ses « clés », associe un effectif (un entier positif). Pour savoir combien de fois le « 1 » est sorti, on demande ce que vaut le tableau d'effectifs en 1 :

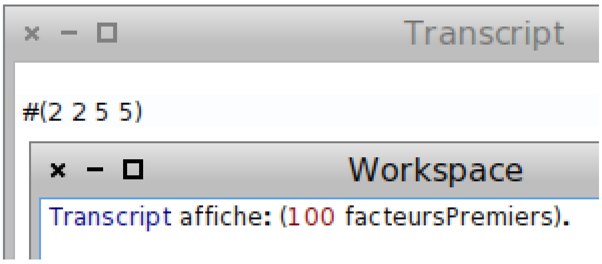


Cependant, faire ça dans une boucle est long, et dans cette partie, on va mettre en œuvre un algorithme rapide pour calculer ce nombre sans avoir à effectuer une boucle sur les pgcd possibles. Cet algorithme est dû à Leonhard Euler[7], qui a montré en 1759 que l'effectif des « 1 » est le produit de facteurs du type $\left(1-\frac{1}{p}\right)$ où p est un facteur premier de 100, ainsi que le nombre 100 lui-même. Pour mettre en œuvre cet algorithme, on a donc besoin de la décomposition de 100 en facteurs premiers.

7 Voir ici pour en savoir plus.

2) Facteurs premiers avec multiplicité

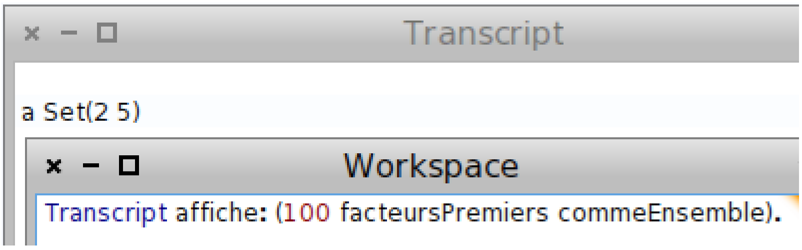
Pour obtenir les facteurs premiers de 100, il suffit de demander à 100 quels sont ses facteurs premiers :



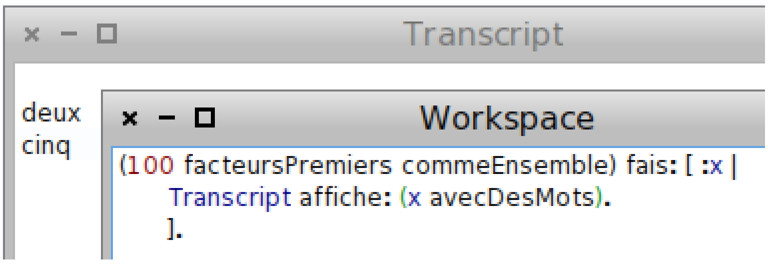
Le problème est que la liste donne tous les facteurs premiers, en comptant plusieurs fois ceux qui apparaissent plusieurs fois. Ce qui rend délicate la tâche de boucler sur ces facteurs premiers. Alors on doit d'abord trouver un moyen de n'avoir qu'une seule fois chacun des facteurs premiers.

3) Facteurs premiers sans multiplicité

Pour cela, il suffit de convertir en ensemble cette liste de facteurs premiers avec multiplicité, puisque chaque élément n'apparaît qu'une seule fois dans un ensemble :



Et MathsOntologie permet de boucler sur les éléments d'un ensemble :



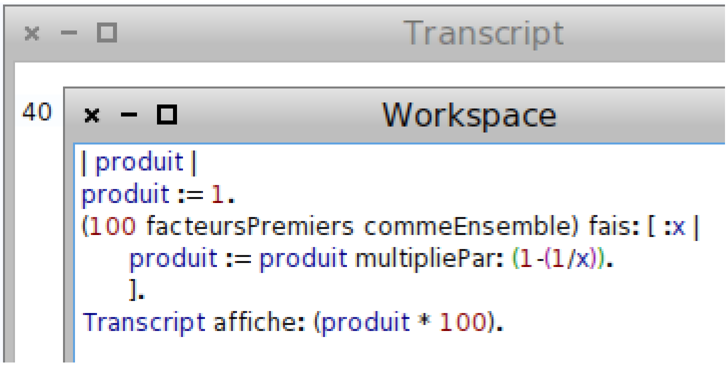
4) Calcul de l'indicatrice d'Euler

Voici l'algorithme d'Euler traduit en MathsOntologie :

- on initialise une variable *produit* à 1 ;
- pour chaque facteur p premier de 100, on multiplie le *produit* par $\left(1-\frac{1}{p}\right)$
- à la fin de la boucle, on multiplie le *produit* par 100

On obtient alors l'indicatrice d'Euler de 100 qui est bien $100\times\left(1-\frac{1}{2}\right)\times\left(1-\frac{1}{5}\right)=100\times\frac{1}{2}\times\frac{4}{5}=40$.

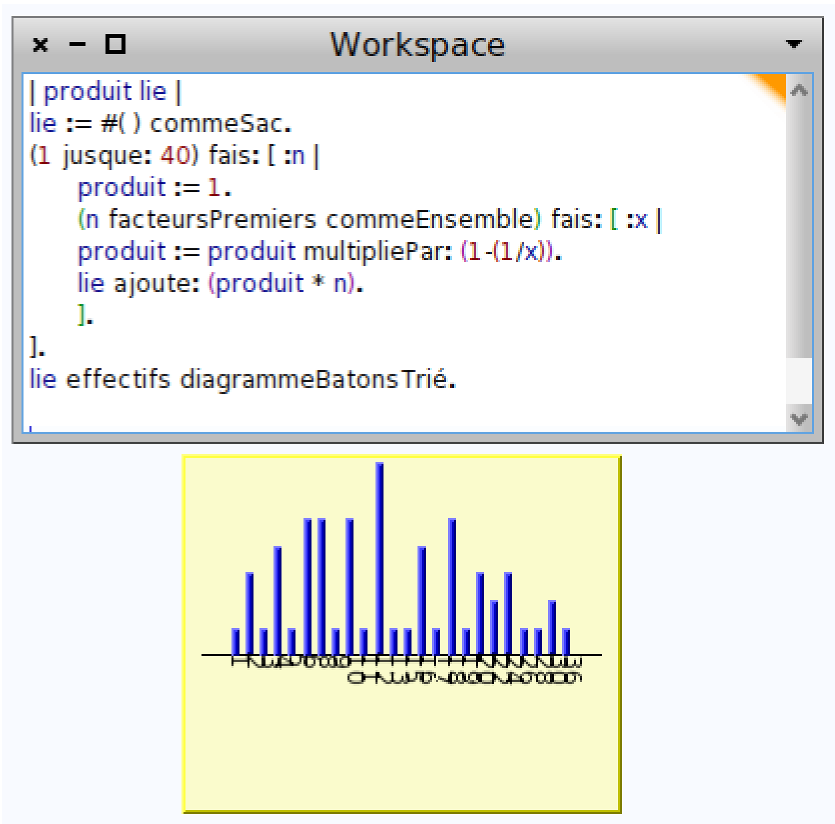
Voici le script obtenu :



III/Statistiques sur les indicatrices d'Euler

1) Diagramme en bâtons

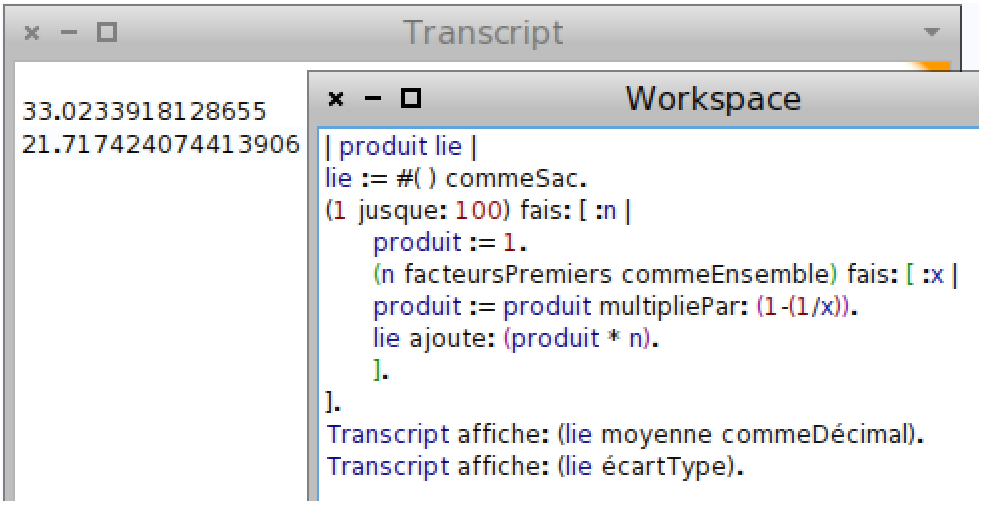
Pour faire des statistiques, on peut calculer l'indicatrice d'Euler dans une boucle, mettre les indicatrices d'Euler dans un sac appelé lie[8], puis afficher comme précédemment le diagramme en bâtons :



8 Lie comme « liste des indicatrices d'Euler »...

2) Moyenne et écart-type

La moyenne s'affiche comme précédemment :



Et bien entendu, l'écart-type aussi (ci-dessus). Ainsi, l'indicatrice moyenne est proche du tiers de 100 et l'écart-type est proche du cinquième de 100. On va conjecturer plus bas que c'est vrai aussi pour d'autres nombres que 100. Pour cela, une représentation graphique permet de voir comment ces nombres dépendent de n (jusqu'ici, n=100).

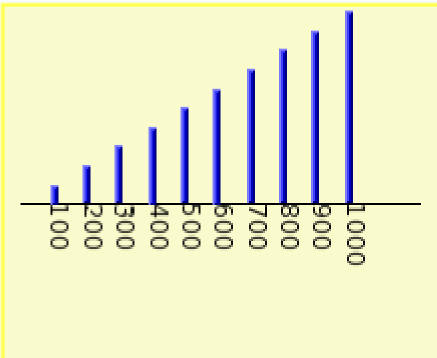
3) Moyenne fonction de n

En représentant graphiquement la moyenne fonction de n (pour n allant de 100 à 1000 par pas de 100) on voit l'alignement qui permet de conjecturer que la moyenne est proportionnelle à n :

Workspace

```
| produit lie stats |
stats := #( ) commeDictionnaire.
(100 jusque: 1000 parPasDe: 100) fais: [ :indice |
    lie := #( ) commeSac.
    (1 jusque: indice) fais: [ :n |
        produit := 1.
        (n facteursPremiers commeEnsemble) fais: [ :x |
        produit := produit multipliePar: (1-(1/x)).
        lie ajoute: (produit * n).
        ].
    ].
    stats en: indice place: (lie moyenne).
].

stats diagrammeBatonsTrié.
```

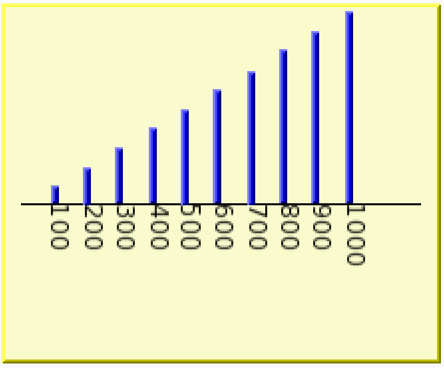


4) Écart-type fonction de n

On fait pareil qu'avec la moyenne, on voit que l'écart-type semble aussi proportionnel à n :

```
Workspace
| produit lie stats |
stats := #( ) commeDictionnaire.
(100 jusque: 1000 parPasDe: 100) fais: [ :indice |
    lie := #( ) commeSac.
    (1 jusque: indice) fais: [ :n |
        produit := 1.
        (n facteursPremiers commeEnsemble) fais: [ :x |
        produit := produit multipliePar: (1-(1/x)).
        lie ajoute: (produit * n).
        ].
    ].
    stats en: indice place: (lie écartType).
].

stats diagrammeBatonsTrié.
```



5) Conclusion

Ces deux conjectures semblent jusqu'aujourd'hui inédites. En regardant les valeurs données pour n=1000, on peut donc conjecturer que la moyenne est proche de 320n, et que l'écart-type est proche de 220n.

Remarque : On peut aussi faire des statistiques sur le nombre de nombres premiers inférieurs à n, mais cela est l'objet de recherches actives depuis plus d'un siècle, avec notamment la célèbre conjecture de Riemann, que l'on peut exprimer comme une estimation de ce nombre. Ceci dit, lorsque l'indicatrice d'Euler est élevée, cela désigne un nombre premier, donc la répartition statistique des indicatrices d'Euler peut très bien donner des renseignements inédits sur celle des nombres premiers...

Alain Busser,
LPO Roland-Garros
Le Tampon