



Graphes, statistiques au format Web

Laurent Barthon

► **To cite this version:**

Laurent Barthon. Graphes, statistiques au format Web. 42èmes Journées de Statistique, 2010, Marseille, France, France. 2010. <inria-00494784>

HAL Id: inria-00494784

<https://hal.inria.fr/inria-00494784>

Submitted on 24 Jun 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

- **support papier** : livre, support de cours, plaquette, brochure, journal, article
- **support électronique** : format logiciel, image, web

En ce qui concerne le **support papier**, il a l'avantage de pouvoir fixer son attention sur plusieurs types d'informations (ex : voir le texte et l'image en même temps, slides Powerpoint sur une même page), donnant une vision plus générale des choses (article, journal, revue, cours).

Par contre, ce type de support a l'inconvénient d'être statique : les données ou résultats d'un schéma, graphe, histogramme, ne sont valables qu'à un instant précis, ou que dans des conditions bien particulières pour faciliter la compréhension du texte ou de l'article le(s) référençant.

Nous allons nous pencher bien évidemment sur le **support électronique** :

- * **logiciel** : pdf, ps, Word, OpenOffice pour ne citer que les plus populaires.

La plupart des présentations graphiques affichées sur ces logiciels restent statiques, sauf peut-être Powerpoint qui permet de créer des animations multimédia.

- * **image** : gif, jpg, bmp, png, tiff ...

La plupart des formats d'images affichables restent également statiques. On peut signaler le format 'gif animé' permet d'afficher sur le Web une suite d'image les unes à la suite des autres. Or cette technique surtout destinée à des fins publicitaires est peu évolutive et nécessite d'avoir des connaissances pointues et des logiciels spécialisés en graphisme.

- * **Internet** : ex : Flash, Applet, Perl, ...

En ce qui concerne l'affichage de données dynamiques, comme l'affichage de cours de bourse, il apparaît que le simple affichage d'images au format Web (gif, jpg ou png), représentant le cours d'un titre à un instant donné, reste vite limité. Or les pages web sont par définition statiques et anonymes : le serveur Web renvoie une page à des milliers de 'clients', ou internautes qui en font la demande (censée être identique pour tous), dont la (les) données ne sont uniquement valides lors de leur chargement. Ainsi, sans rafraichissement de la page sur le navigateur, ou technique permettant de mettre à jour des informations, celles-ci restent figées.

Dans ces conditions, des techniques ont vu le jour pour diffuser du contenu, ou plutôt des graphes de manière plus 'dynamique' :

- Image générée en Perl : l'image est recrée sur le serveur Web à chaque rafraichissement de la page de la part de l'utilisateur, or là encore, la modification n'est pas automatique,
- Graphisme générés par des logiciels spécialisés, comme Flash de Macromedia ou des Applets Java, nécessitant des plugins installés sur le navigateur.

Affichage sur le Web

Ressources Web

Une autre technique d'affichage de graphes ou histogrammes consiste à utiliser des ressources, techniques ou langages disponibles sur le Web, ne nécessitant pas d'installation de package ou langage spécialisé et/ou payant.

Ces ressources sont les langages propres au Web décrits ci-dessous dans les grandes lignes :

- Le **HTML** est la base du Web. C'est un langage statique ressemblant au XML.
Les liens habituellement utilisés pour 'naviguer' d'une page à une autre peuvent aussi servir à modifier / mettre à jour des valeurs dans une page. De plus, les options de formulaire nous intéressent particulièrement, à savoir : cases à cocher, bouton radio, liste déroulante, bouton de validation.
- Le langage **JavaScript** est un langage interprété, mais un langage objet !
Il permet de pallier le manque d'interactivité du HTML (exemple habituel : validation de champs d'un formulaire avant envoi), et dispose de quelques concepts puissants permettant de manipuler des nombres, des dates, des tableaux de données ou chaînes de caractères.
- Le **DOM** ou Document Object Model vient en complément du JavaScript.

Graphe et histogrammes

Ainsi, dans les grandes lignes, la technique d'affichage de graphes ou d'histogrammes repose sur un système de coordonnées à partir du coin supérieur gauche du navigateur Web en cours d'utilisation (Internet Explorer, Firefox, Safari, Opera, et dernièrement Chrome), sous la forme (left, top), avec comme unité la plus courante : le pixel.

A partir de cette hypothèse, il est possible de définir une zone de dessin dans une page Web, de redéfinir un système de coordonnées plus classique, et de tracer une série de points au pixel près représentant un graphe, courbe, ou histogramme avec barres, voire des Tartes (Pie), ou disques sur le Web.

MAIS il faut bien avoir en tête que la si technique d'affichage et surtout la syntaxe correspondante semble simple de premier abord, elle n'est pas si simple à mettre en œuvre, et n'est malheureusement pas toujours interprétée exactement de la même façon par tous les navigateurs.

De plus, même si la syntaxe permettant de positionner un point au pixel près reste relativement simple et accessible, la structure finale d'une courbe de n points reste relativement lourde (chargement par le navigateur) par rapport à d'autres technologies.

Dans ces conditions, il sera difficile pour des animations multimédia Web 'poussées' de concurrencer avec les mêmes animations développées dans des langages performants comme le C, C++ ou Java.

NEANMOINS, si les algorithmes de traitement des données à la fois graphiques et numériques sont puissants et bien organisés (JavaScript langage objet), il est tout à fait possible d'obtenir des animations performantes ou des mises à jour de données quasi-immédiates.

Framework JavaScript

Ainsi, à partir de ces remarques, j'ai créé un framework JavaScript permettant de gérer l'affichage de fonctions, ou lois de probabilité dans une zone prévue à cet effet. Je me suis aidé de différents utilitaires ou frameworks existants, présentés dans la bibliographie, notamment :

- utilitaire d'animations (durée, intervalle, transition) : *Animator.js*,
- gestion de l'événement clavier : *shortcut.js*,
- gestion et construction de graphes, barres : *Diagram Builder*,
- Le site web www.developpez.com contenant un bon nombre de ressources Web, notamment un très bon article sur la gestion des caractères spéciaux sur le Web.

Signalons deux atouts supplémentaires liés au format Web : la prise en charge des caractères spéciaux et la gestion des événements. Il est possible d'afficher certaines équations mathématiques : le Web gère les lettres grecques ainsi que quelques symboles mathématiques : voir ci-dessous.

Caractères spéciaux	Affichage/'Syntaxe Web'
Les caractères réservés pour la syntaxe web	& ' &lt;' ' >'
Les caractères non ASCII , utilisés couramment dans une langue comme le 'c cédille' 'ç' dans français.	ç ' ç' à ' à'
Les caractères grecs , très utilisés en mathématique. Il suffit d'écrire en toute lettre le nom de la lettre grecque précédée par '&' et suivie de ';'. Si la lettre est capitalisée (1 ^{ère} lettre en majuscule), alors la lettre grecque aussi ! Ainsi :	é, è, ê ' é' ' è' ' ê'
	β ' β'
	δ ' δ'
	μ ' μ'
	σ Σ ' σ' ' Σ'
	Γ Φ ' Γ' ' Φ'
Les caractères Mathématiques disponibles à partir de la syntaxe WEB. J'ai noté les principaux :	± ½ ¼ ' ±' ' ½' ' ¼'
	≤ ≥ ≠ ≈ ' ≤' ' ≥' ' ≠' ' ≈'
	Σ ∞ √ ∫ ' ∑' ' ∞' ' √' ' ∫'
	← → ↔ ' ←' ' →' ' ↔'
Enfin des caractères universels de monnaie ou de marque :	© ® ™ ' ©' ' ®' ' ™'
	€ £ ¥ ' €' ' £' ' ¥'

De plus, JavaScript peut gérer des événements divers comme :

- clic souris : rafraichissement de données sans rechargement de la page Web,
- gestion du curseur de la souris dans une zone d'affichage (coordonnées),

- action déclenchée à partir d'une touche clavier, ou d'une combinaison de touches,
- des animations de courbes ou de points en mouvement.

Enfin, il faut noter que JavaScript ne gère pas l'accès à une base de données. Il faut coupler le langage Web (HTML, CSS, JavaScript) avec un langage plus évolué comme PHP ou ASP ou Java (jsp, jsf) pour avoir accès à des données (comme le cours de bourse en temps réel, par exemple).

Ainsi, dans le cas du graphe de table de mortalité (voir graphe plus loin), les données ont été simplement copiées d'Excel et collées dans un tableau de données au format JavaScript (Array).

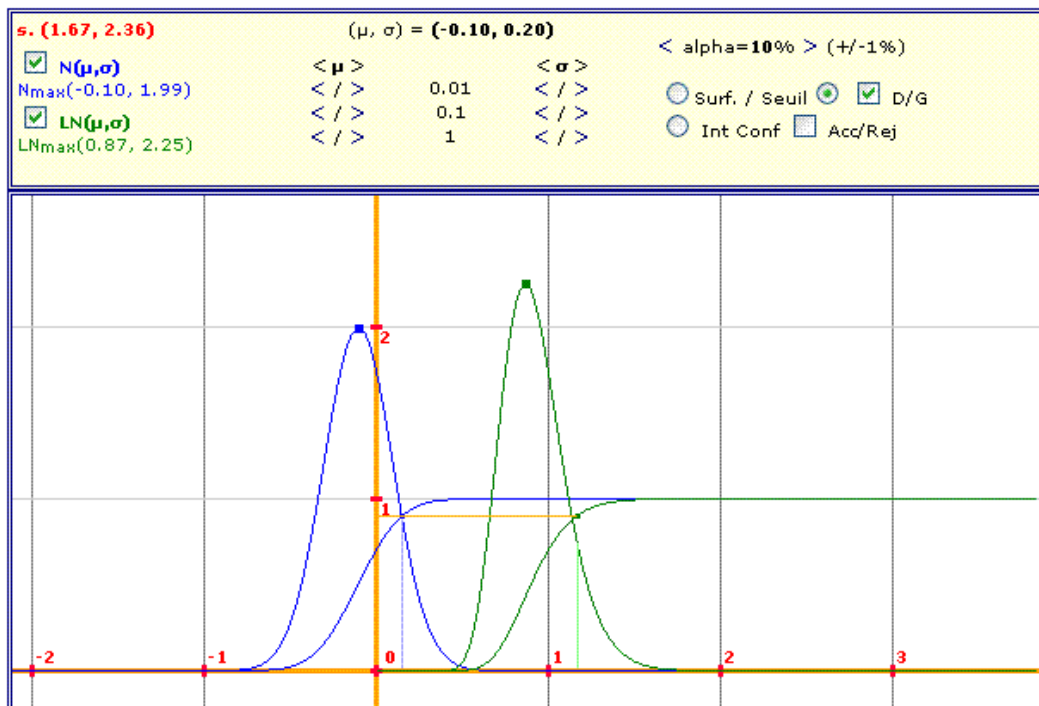
Des exemples

Les exemples et les copies écran figurant ci-dessous fonctionnent sur les pages Web que j'ai créées et que je pense bientôt mettre en ligne: les liens, cases à cocher, bouton radio, liens play() et autres boutons sont opérationnels dans un environnement Web :

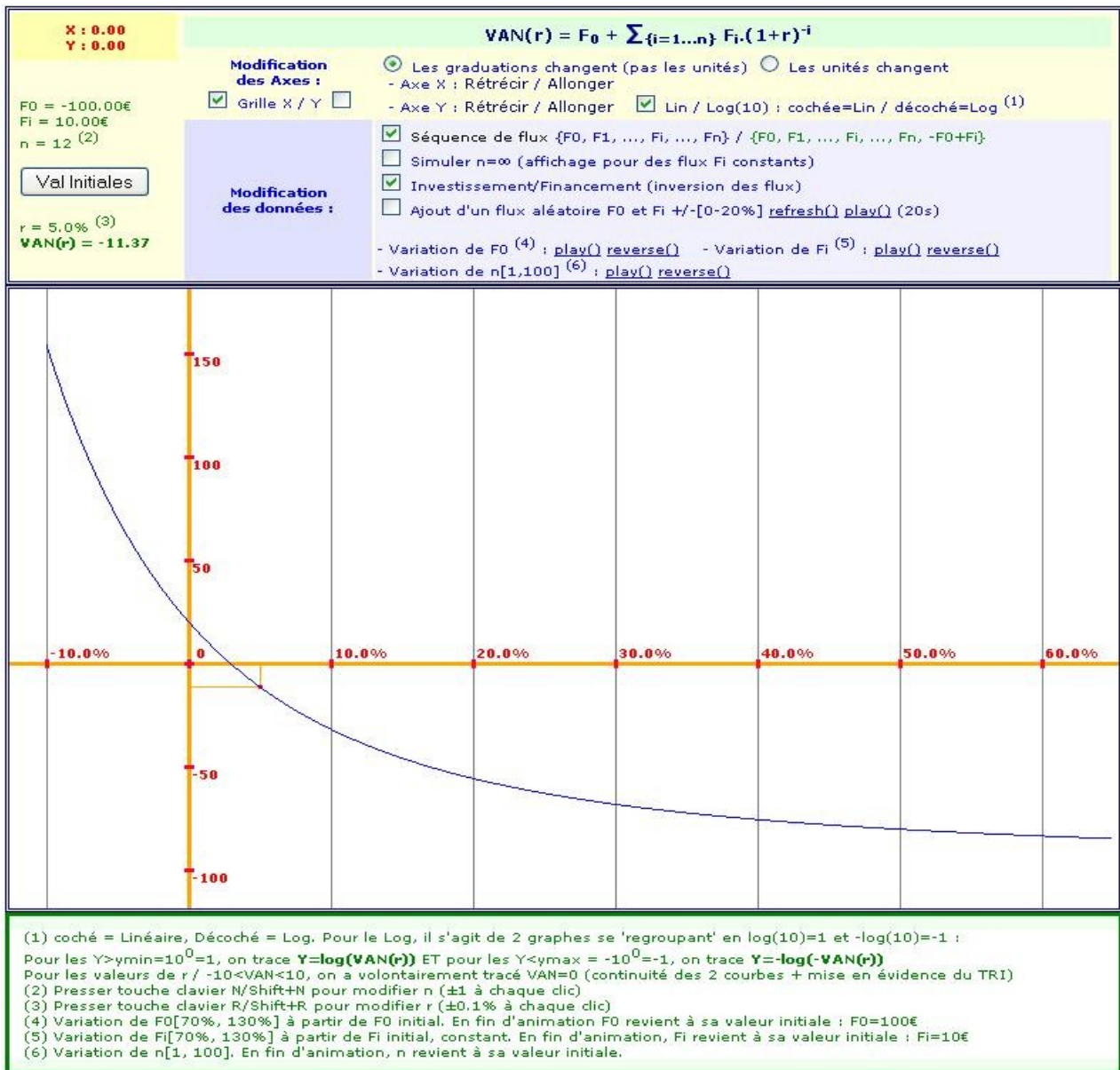
- affichage dynamique de lois de probabilité. Ex : loi Normale et log-Normale
- graphe de la VAN d'un investissement (finance),
- histogramme de table de mortalité (actuariat, assurances).
- histogramme de la loi de Poisson, ou Binomiale (pas dans ce document)
- sur le grill : graphe d'un mouvement brownien, évaluation d'options (à venir)

Ci-dessous, exemple (copie écran) d'affichage simultané de deux lois Normale (en bleu) et Log Normale (en vert) de même paramètre mu et sigma, et leur fonction de répartition.

Des liens permettent de faire varier mu, sigma, ou alpha et de visualiser le résultat instantanément. Il est aussi possible d'afficher un rejet sous forme de trait (seuil) ou de surface à droite comme à gauche, comme un intervalle de confiance (alpha/2).



Graphe interactif de la VAN d'un investissement et visualisation graphique du TRI :



Tables de mortalité TG05H/TG05F, année 2005 : données...

Age x	Nb vivants à l'age x H/F	Nb décès entre x et x+1 H/F
0	$l_0 = 100000/100000$	$d_0 = 317/255$
1	$l_1 = 99683/99745$	$d_1 = 100/82$
2	$l_2 = 99583/99663$	$d_2 = 43/37$
3	$l_3 = 99540/99626$	$d_3 = 25/21$
...
x=50... x=100...	$l_{50} = 98587/98187...$ $l_{100} = 35628/51956...$	$d_{50} = 52/99...$ $d_{100} = 4150/4649...$
ω	$l_\omega = 181/516$ $l_{\omega+1} = 0/0$	$d_\omega = 181/516$

... et histogramme correspondant :

Table de mortalité TC05H/TC05F

H/F(1) Axe Y Lin/Log (2)

l_x 100000-l_x d_x FRd_x
 Age limite : ω=120 ans
 Taux instantané de mortalité
 $\mu_x \approx -\frac{1}{2}(l_{x+1}-l_{x-1})/l_x$
 Taux central annuel de décès
 $m_x = 2d_x/(l_x+l_{x+1})$
 Espérance de vie
 $e_x = (l_{x+1} + \dots + l_{x+\omega})/l_x$

Individu vivant à l'âge x
 vit encore à / meurt avant x+k
 ${}_k p_x (=l_{x+k}/l_x) / {}_k q_x (=1-{}_k p_x)$ < k=1 > (4)
 Individu age x meurt la x+k^{ième} année
 ${}_{k-1} | q_x = d_{x+k-1}/l_x$
 Individu age x meurt entre [x+k, x+k+t] < t=40 > (5)
 ${}_k | t q_x (=k p_x - k+t p_x)$

A.N. : Age test = 50 ans⁽³⁾ μ₅₀ = 5.173e-4 / 9.777e-4 e₅₀ = 45.4 / 48.9 m₅₀ = 0.001 / 0.001
 < a=50 > 1p₅₀ = 9.995e-1 / 9.990e-1

(1) case à cocher interactive (H=coché), ou presser touches clavier 'H'/'F'
 (2) case à cocher interactive (Lin=coché) : pas encore fonctionnel
 (3) Age = 50 ans (chargement de la page), touches 'Shift+A'/'A' pour age ±1
 (4) clic sur lien '<'/'>' (ou touches 'Shift+K'/'K') déclenche k -/+1
 (5) clic sur lien '<'/'>' (ou touches 'Shift+T'/'T') déclenche t -/+1

Bibliographie

- [1] Animation sur le Web : Animator.js
<http://berniecode.com/writing/animator.html>
- [2] Gestion du clavier : Shortcut.js
http://www.openjs.com/scripts/events/keyboard_shortcuts/
- [3] Affichage de graphe : Diagram Builder.
<http://www.lutano.net/diagram/>
- [4] Caractères web sur developpez.com
<http://gdumas.developpez.com/articles/xhtml/caracteres-speciaux/>