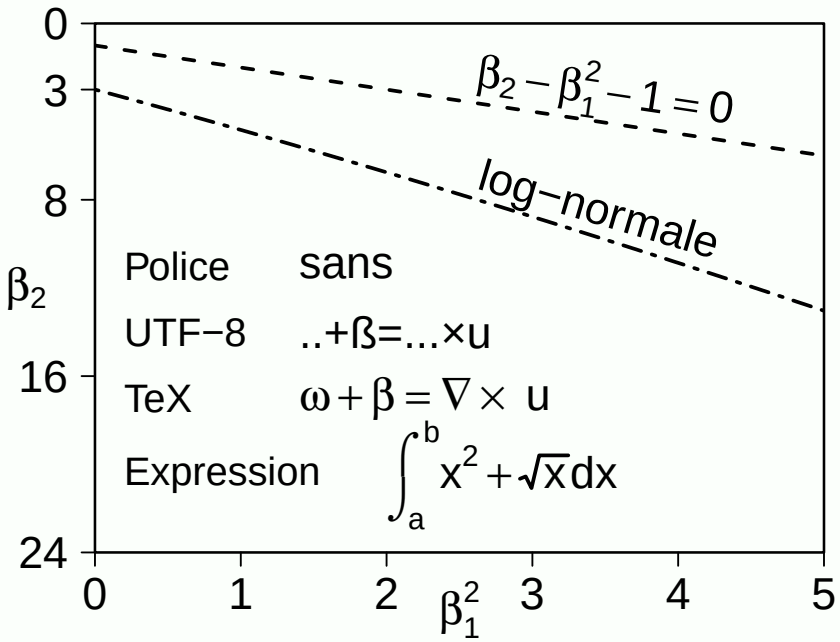
Meetup R-Addicts : Typographie avec R

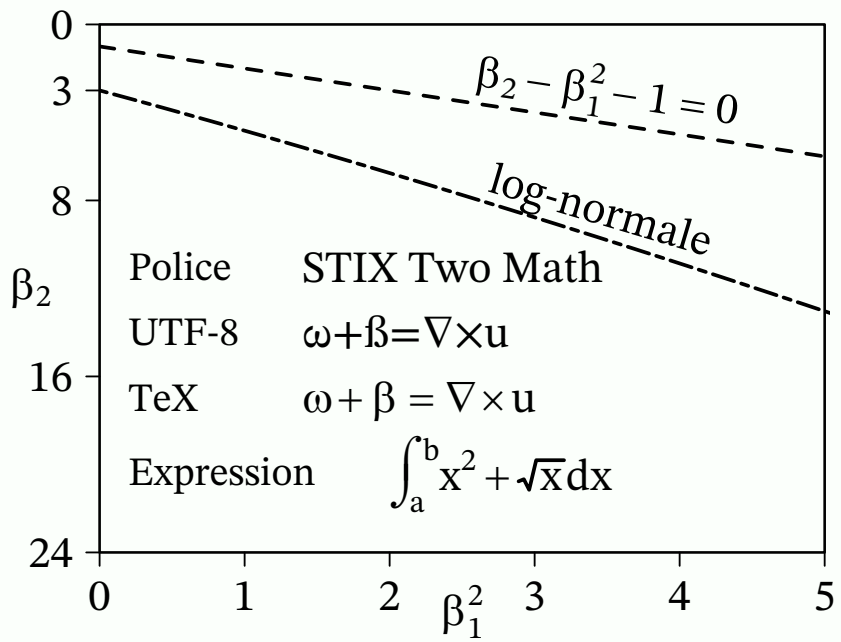
patrice.kiener@inmodelia.com

Paris – 19 juillet 2018

Typographie avec R

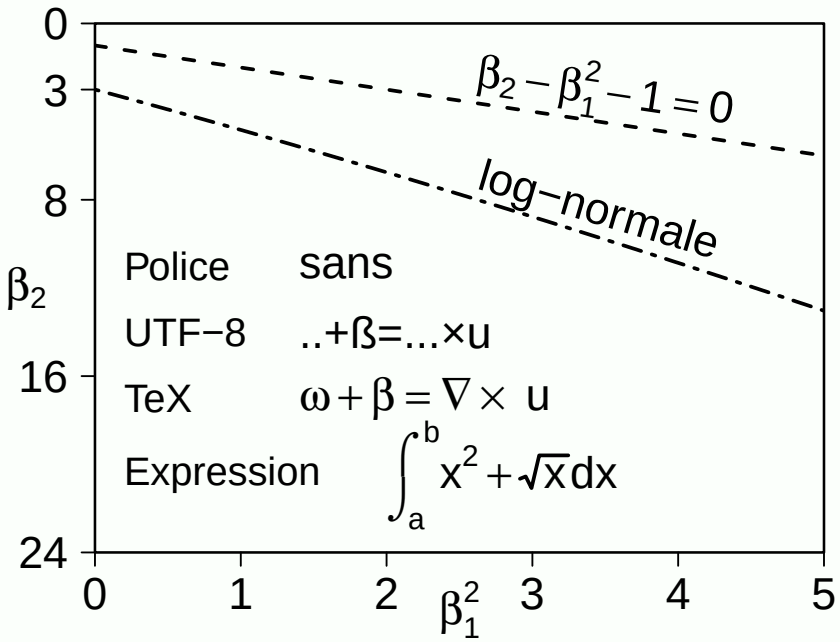
Question : Quelles différences voyez-vous entre ces deux graphiques ?

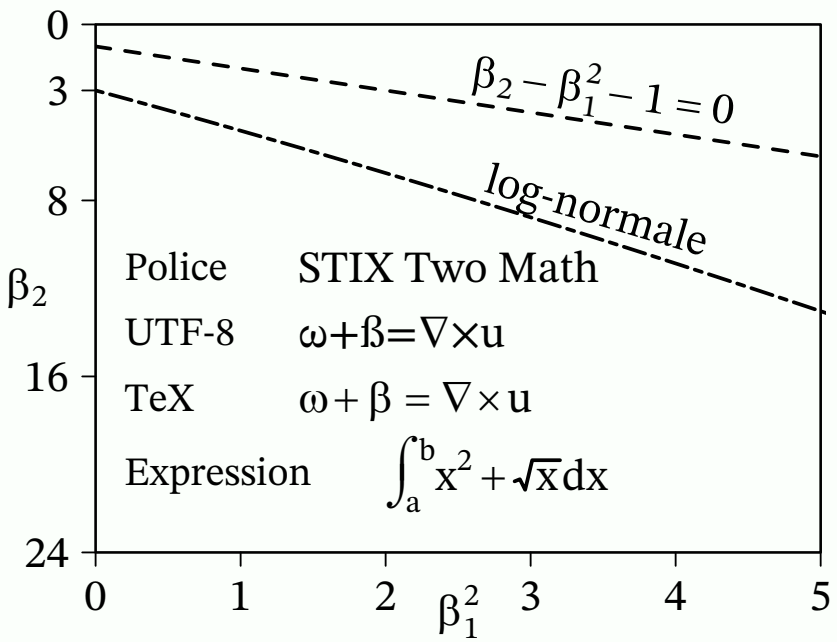




Typographie avec R

Question : Quelles différences voyez-vous entre ces deux graphiques ?





Réponse : 1) **La police de caractères** ; 2) L'instruction qui génère le fichier.

Bienvenue dans la typographie avec R !

Graphiques et polices de caractères

Les graphiques de R ont 3 chemins d'accès aux polices de caractères :
1) s+s+m+W+X+Q, 2) pdfFonts, 3) Cairo.

Familles/Polices	Instructions R
	ÉCRAN
0 choix	win.graph(), pictex()
serif+sans+mono+	
windowsFonts	windows(), win.metafile(), win.print()
X11Fonts	X11(), x11()
quartzFonts	quartz()
	FICHIERS
s+s+m+W+X+Q	png(), jpeg(), bmp(),
	tiff(..., **type = "windows", "Xlib", "quartz"**)
Cairo	png(), jpeg(), bmp(), tiff(..., **type = "cairo"**)
pdfFonts	pdf(), postscript(), bitmap(), xfig()
Cairo	cairo_pdf(), cairo_ps(), svg()
	PACKAGES TIERS : FICHIERS + *ÉCRAN*
package Cairo	Cairo(), CairoPDF(), CairoPNG(), *CairoWIN()*
package cairoDevice	Cairo(), Cairo_pdf(), Cairo_ps(), Cairo_png()

Exemples

Exemples :

- plot(1:5, 1:5, las = 1, main = "Police Stencil", family = "Stencil")
- png(..., family = "serif", font = 2)
- png(..., family = "Stencil")
- png(..., type = "cairo", family = "Stix Two Math")
- pdf(..., family = "Palatino")
- cairo_pdf(..., family = "Stix Two Math")

Point essentiel ⇒ Où trouve-t-on les polices ?

Les polices serif+sans+mono+Win+X11+Quartz

- Approche globale : Modifier le fichier *R_HOME/etc/Rdevga* qui définit les 3 premières polices *serif, sans, mono* des systèmes Windows, Linux, MacOSX, Solaris, BSD.
- Approche locale : Rajouter des polices True Type ou Open Type proposées par chaque système d'exploitation dans son répertoire /Fonts

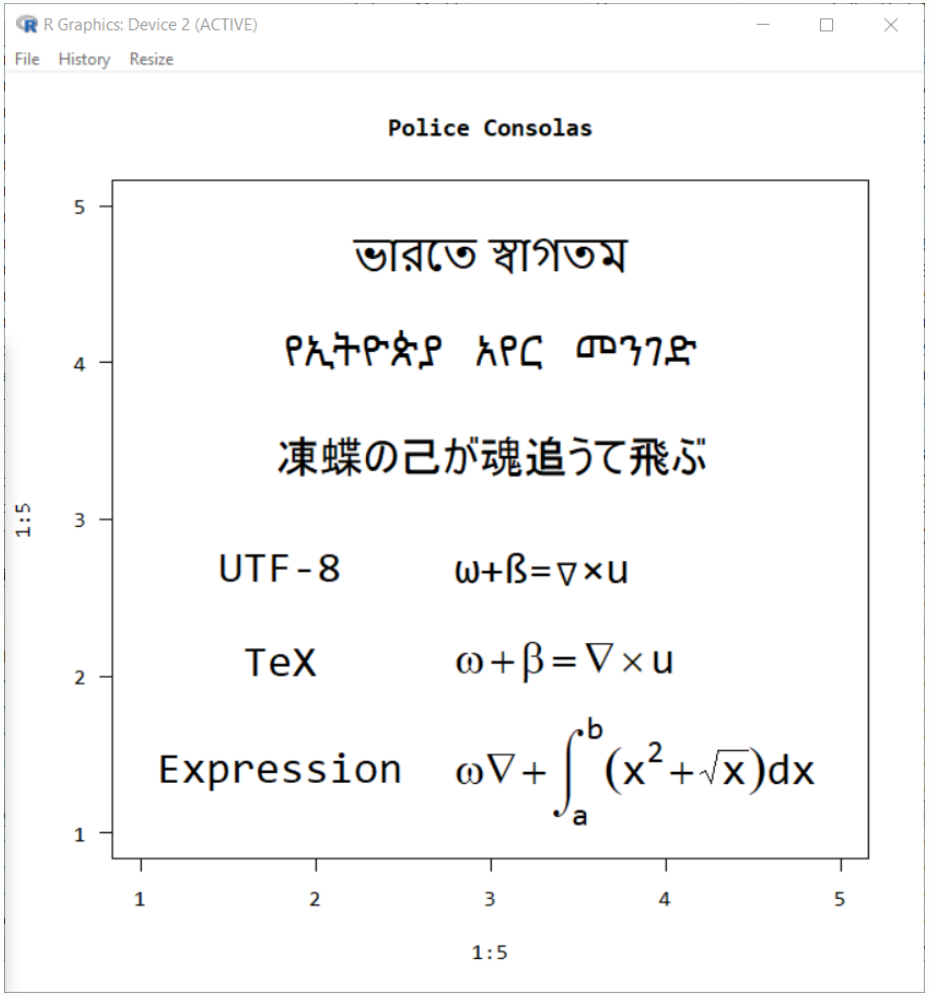
```
windowsFonts(Consolas = "Consolas",  Stencil = "Stencil",
             BookA = "Book Antiqua", StixTT  = "Stix Two Text")
windowsFonts()  #  Windows                 # Linux
$serif          # "TT Times New Roman"     # "Times
$sans           # "TT Arial"               # "Helvetica"
$mono           # "TT Courier New"         # "courier"
$Consolas       # "Consolas"
$Stencil        # "Stencil"
$BookA          # "Book Antiqua"
$StixTT         # "Stix Two Text"

# X11Fonts(Stencil = "Stencil")    ; X11Fonts()
# quartzFonts(Stencil = "Stencil") ; quartzFonts()
```

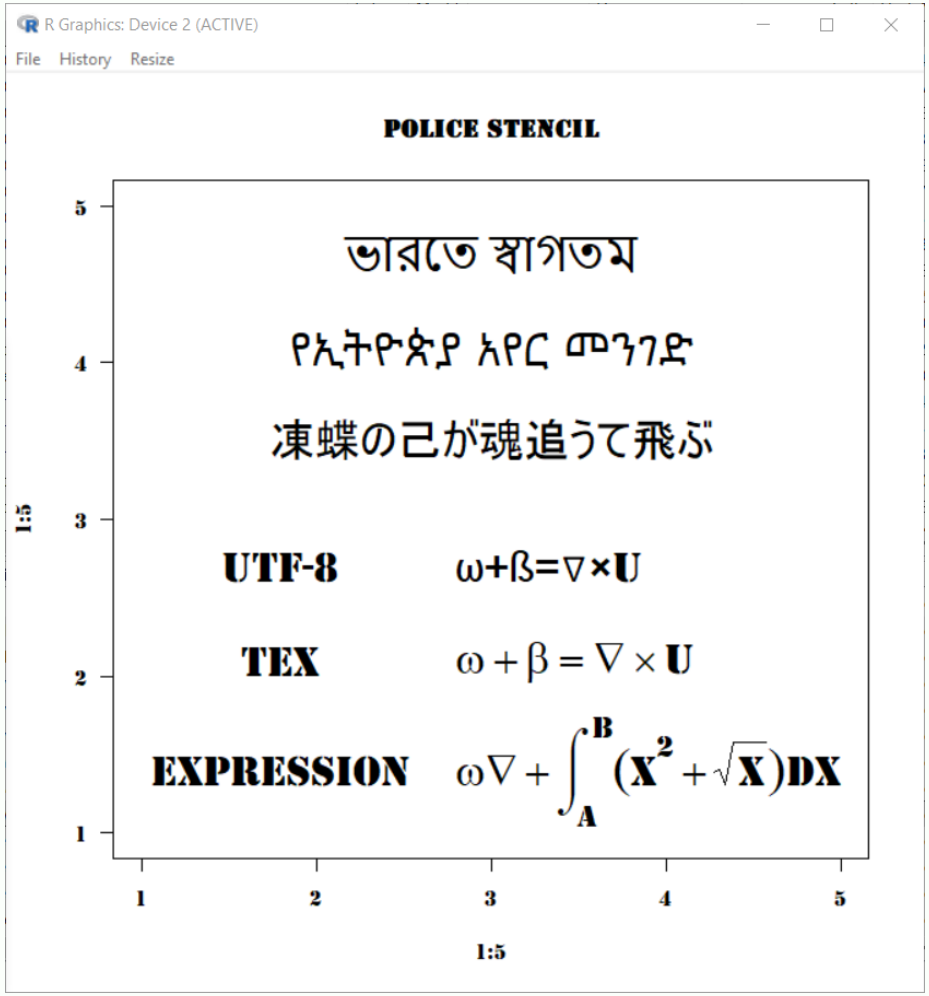
Fenêtre R et code s+s+m+W+X+Q

- Une fois la police enregistrée dans R avec un nom court, on peut l'utiliser dans un graphique avec `par(family=famille)`

```
famille <- "Consolas"
# famille <- "Stencil"
# famille <- "BookA"
# famille <- "StixTT"
op <- par(family = famille)
plot(1:5, 1:5,  col = 0, las = 1, main = paste("Police", famille))
text(3.0, 4.7, "ভারতে স্বাগতম", cex = 2.5)          # indien bengali
text(3.0, 4.1, "የኢትዮጵያ አየር መንገድ", cex = 2.5)     # ethiopien
text(3.0, 3.4, "凍蝶の己が魂追うて飛ぶ", cex = 2)  # japonais
text(1.8, 2.7, "UTF-8", cex = 2)
text(2.8, 2.7, "ω+β=▽×u", cex = 2, adj = 0)
text(1.8, 2.1, "TeX", cex = 2)
text(2.8, 2.1,  latex2exp::TeX("$\\omega + \\beta = \\nabla \\times u$"),
                cex = 2, adj = 0)
text(1.8, 1.4, "Expression", cex = 2)
text(2.8, 1.4,  expression(omega*nabla + integral((x^2+sqrt(x))*dx, a, b)),
                cex = 2, adj = 0)
par(op)
```

Fenêtre R et polices Consolas + Stencil



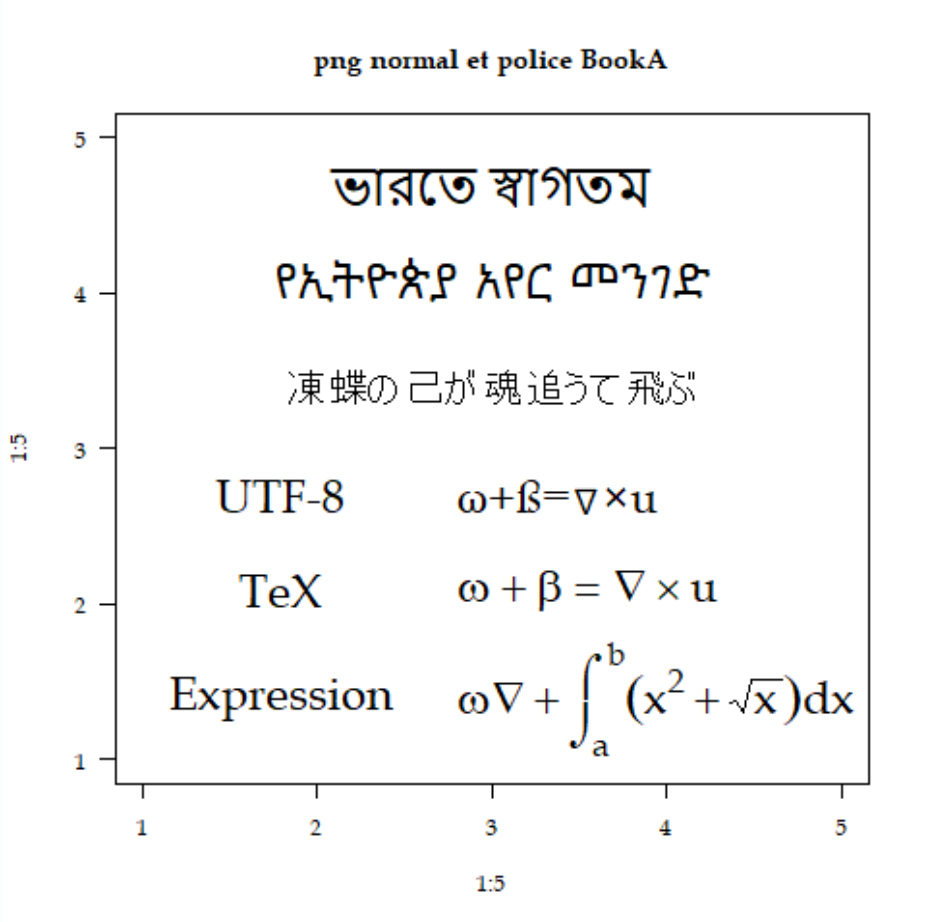


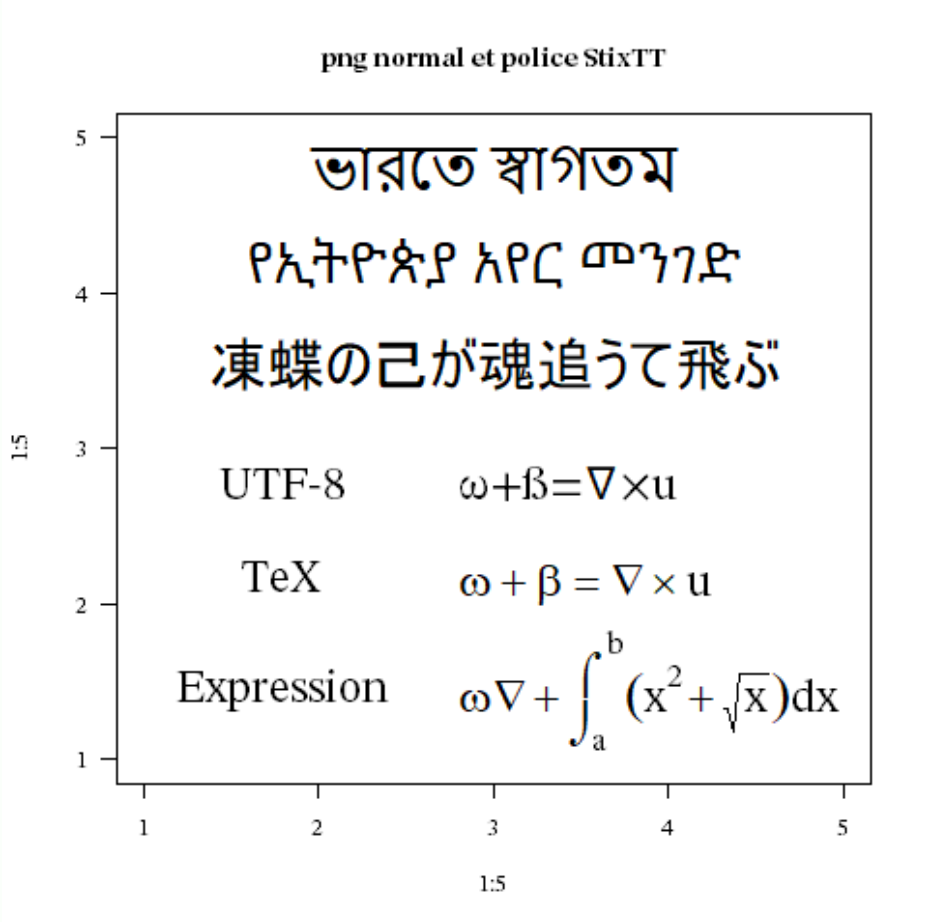
Fonction png() et polices s+s+m+W+X+Q

- Pour un graphique png ordinaire, on insère l'option `family=famille` dans png() et dans op()

```
# famille <- "Consolas"
# famille <- "Stencil"
famille <- "BookA"
# famille <- "StixTT"
png("pngnormal-BookA.png", type = "windows", family = famille)
op <- par(family = famille)
plot(1:5, 1:5,  col = 0, las = 1, main = paste("png normal et police", famille))
...
...
par(op)
dev.off()
```

Fichier png et polices Book Antiqua + Text TT





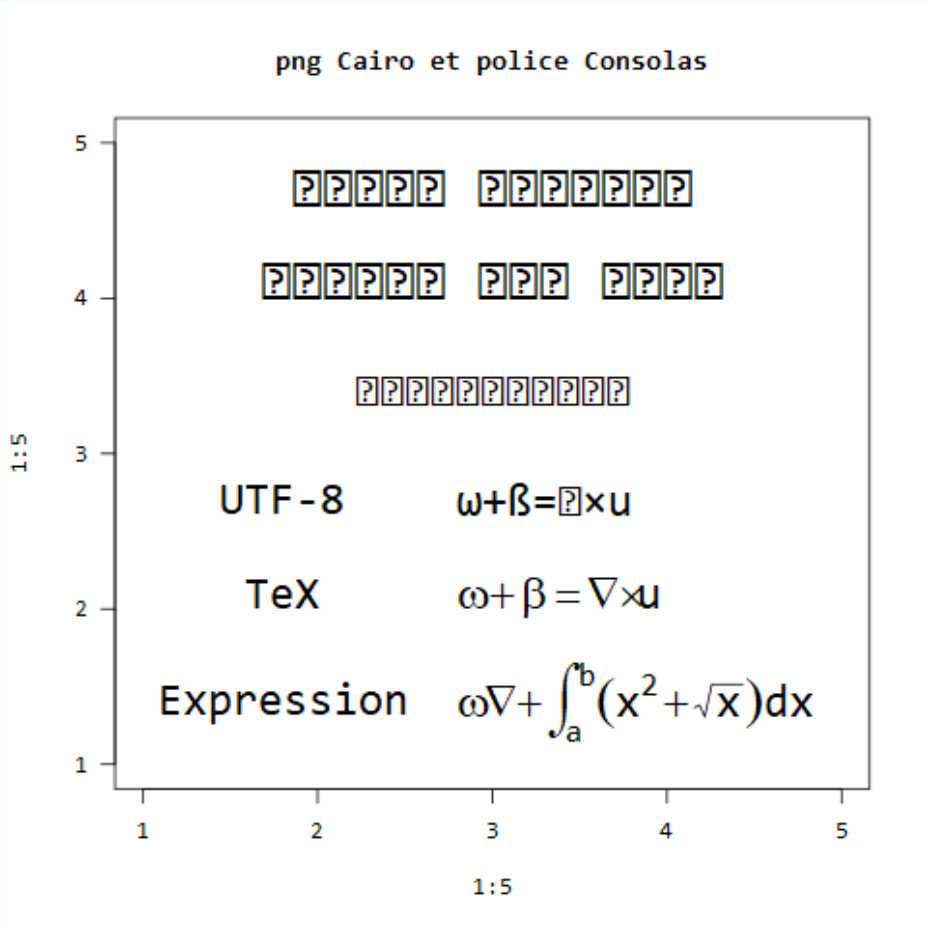
Fonction png(type = "cairo") et polices système

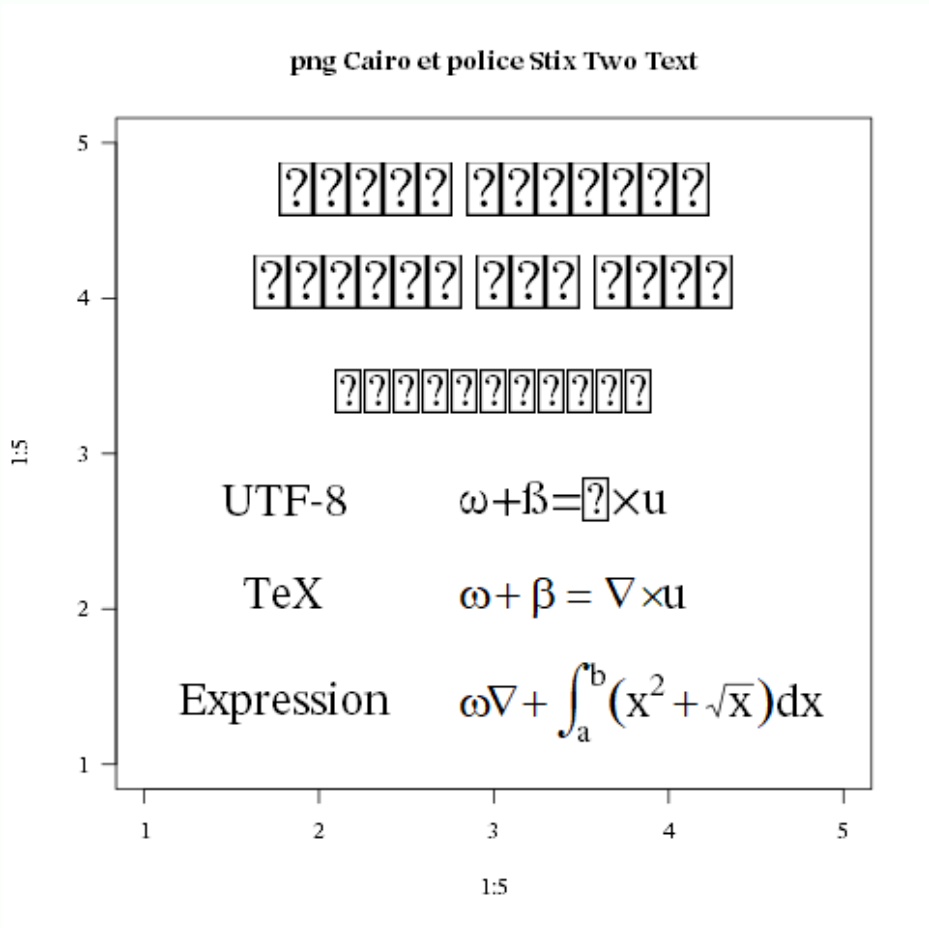
- Cairo est une librairie graphique indépendante (issue de GTK) qui propose une voie unifiée pour la création d'images bitmap ou vectorielles.
- Cairo n'utilise pas de cache intermédiaire et cherche les polices directement dans le répertoire /Fonts via leur nom officiel.

```
# famille2 <- "Consolas"
# famille2 <- "Stencil"
# famille2 <- "Book Antiqua"  # NOM OFFICIEL DANS LE RÉPERTOIRE /Fonts !!
famille2 <- "Stix Two Text"
png("pngcairo-StixTT.png", type = "cairo", family = famille2)
op <- par(family = famille2)
plot(1:5, 1:5,  col = 0, las = 1, main = paste("png Cairo et police", famille2))
...
...
par(op)
dev.off()
```

Fichier png Cairo et polices Consolas + Text TT

- Ces polices n'ont pas de version indienne, éthiopienne ou japonaise.
- `png(type="cairo")` n'a pas le mécanisme de substitution de `png()` !





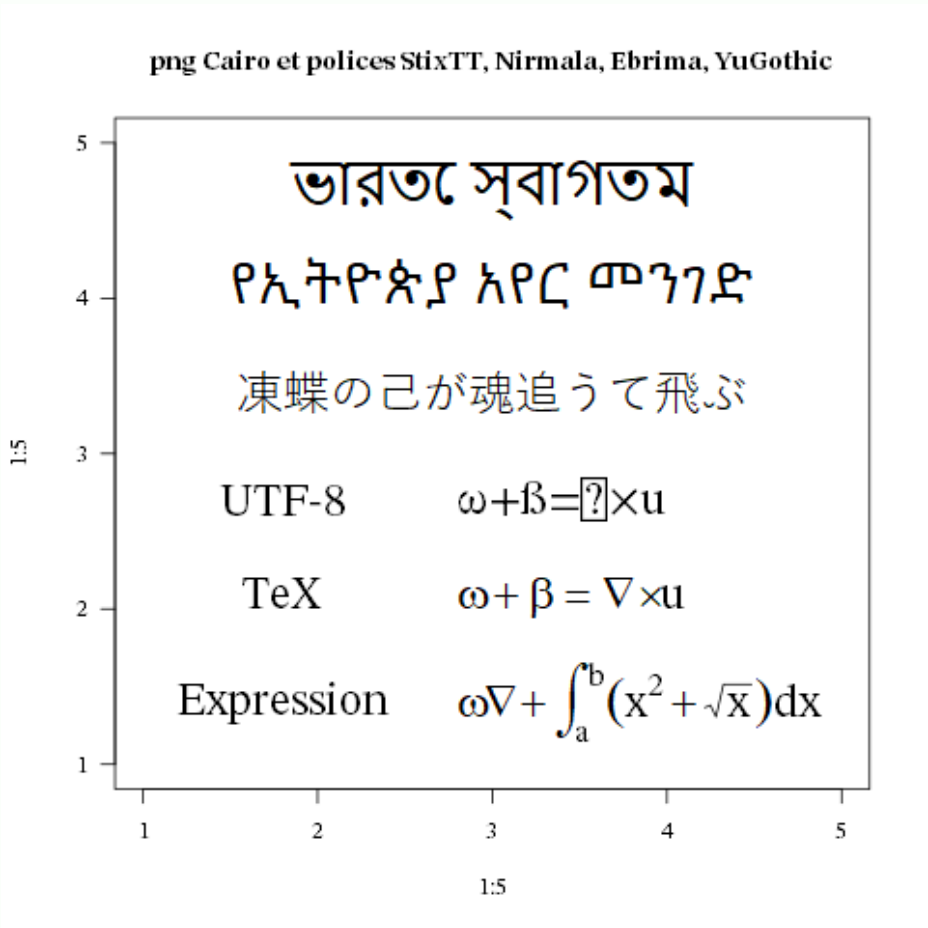
Fonction png(type = "cairo") et multiples polices

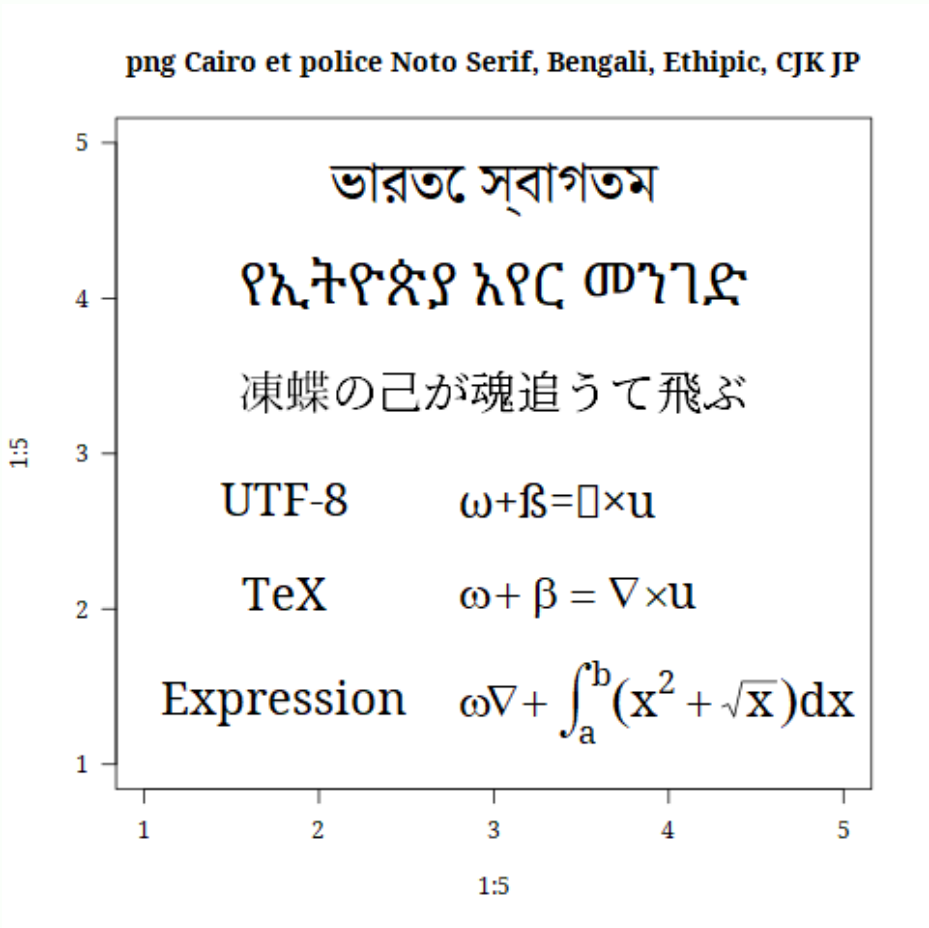
- Il faut combiner des polices UTF-8
- Windows : Stix Two Text, Nirmala UI, Ebrima, Yu Gothic
- Google : Noto Serif, Noto Serif Bengali, Noto Serif Ethiopic, Noto Serif CJK JP

```
famille2 <- "Noto Serif"
famille3 <- "Noto Serif, Bengali, Ethipic, CJK JP"
png("pngcairo-Noto.png", type = "cairo", family = famille2)
op <- par(family = famille2)
plot(1:5, 1:5,  col = 0, las = 1, main = paste("png Cairo et police", famille3))
text(3.0, 4.7, "ভারতে স্বাগতম", cex = 2.5,          family = "Noto Serif Bengali")
text(3.0, 4.1, "የኢትዮጵያ አየር መንገድ", cex = 2.5,     family = "Noto Serif Ethiopic")
text(3.0, 3.4, "凍蝶の己が魂追うて飛ぶ", cex = 2, family = "Noto Serif CJK JP")
...
...
par(op)
dev.off()
```

Fichier png Cairo et polices multiples + Noto

- Le rendu à l'affichage est maintenant excellent





Les polices pdfFonts pour pdf() et postscript()

postscript et pdf sont des méthodes de rendu vectoriel crées par Adobe

- R encapsule les 11 polices imposées par Adobe plus des polices CJK
- C'est un mécanisme entièrement indépendant de s+s+m+W+X+Q

```
> names(pdfFonts())
 [1] "serif"                "sans"                 "mono"
 [4] "AvantGarde"           "Bookman"              "Courier"
 [7] "Helvetica"            "Helvetica-Narrow"     "NewCenturySchoolbook"
[10] "Palatino"             "Times"                "URWGothic"
[13] "URWBookman"           "NimbusMon"            "NimbusSan"
[16] "URWHelvetica"         "NimbusSanCond"        "CenturySch"
[19] "URWPalladio"          "NimbusRom"            "URWTimes"
[22] "ArialMT"              "Japan1"               "Japan1HeiMin"
[25] "Japan1GothicBBB"      "Japan1Ryumin"         "Korea1"
[28] "Korea1deb"            "CNS1"                 "GB1"
>
>
> pdf.options(font = "Palatino")
> pdf.options(family = "Palatino")
> pdf.options(reset = TRUE)
```

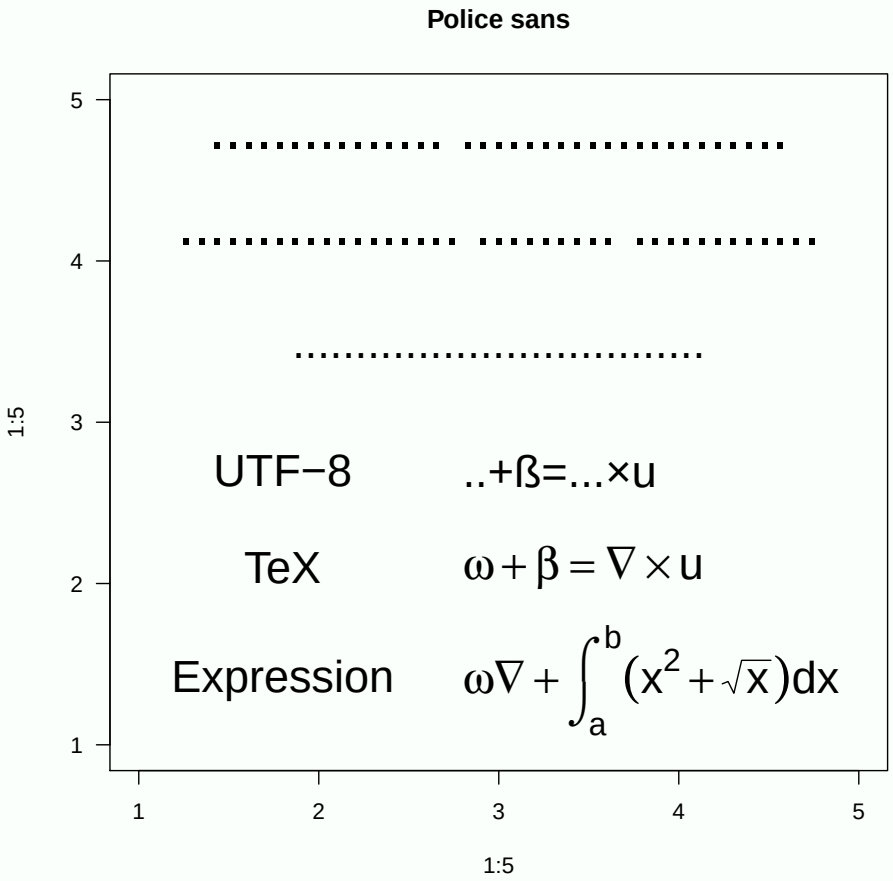
Fonction pdf() et polices pdfFonts

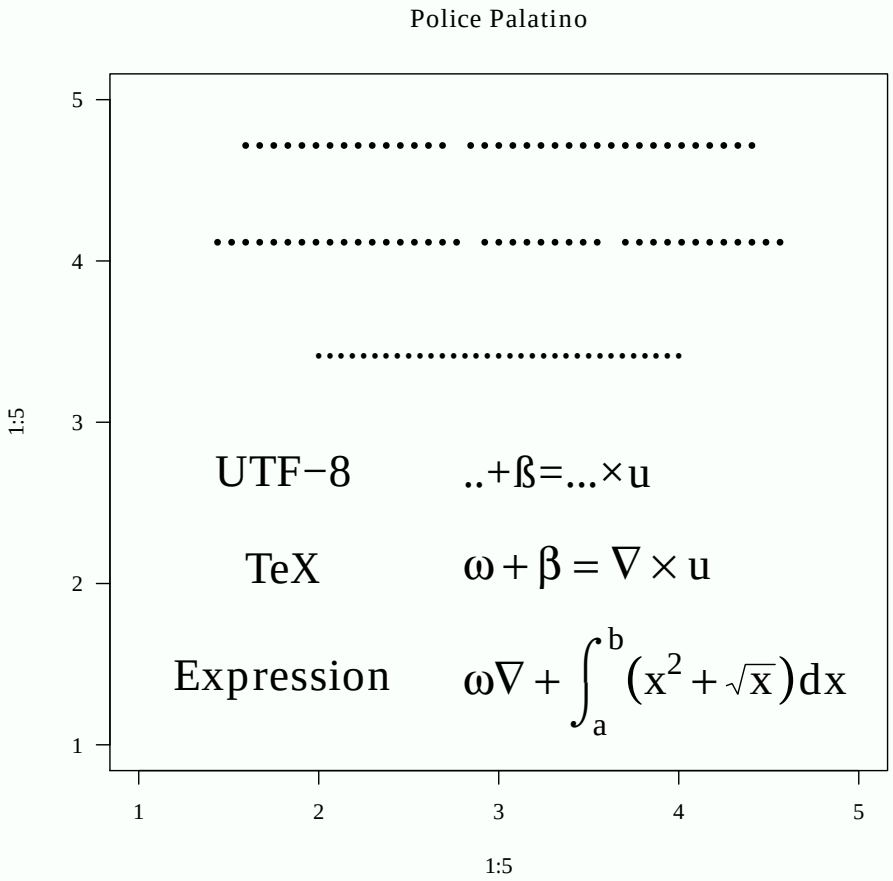
- Pour un fichier pdf ordinaire, on insère l'option `family=famille` dans pdf() et dans op()

```
famille <- "Palatino"
pdf("pdf-Palatino.pdf", width = 7, height = 7, family = famille)
op <- par(family = famille)
plot(1:5, 1:5,  col = 0, las = 1, main = paste("Police", famille))
text(3.0, 4.7, "ভারতে স্বাগতম", cex = 2.5)          # indien bengali
# There were 50 or more warnings (use warnings() to see the first 50)
text(3.0, 4.1, "የኢትዮጵያ እየር መንገድ", cex = 2.5)     # ethiopien
# There were 50 or more warnings (use warnings() to see the first 50)
text(3.0, 3.4, "凍蝶の己が魂追うて飛ぶ", cex = 2)  # japonais
# There were 50 or more warnings (use warnings() to see the first 50)
text(1.8, 2.7, "UTF-8", cex = 2)
text(2.8, 2.7, "ω+β=▽×u", cex = 2, adj = 0)
# There were 12 warnings (use warnings() to see them)
...
...
par(op)
dev.off()
```

Fichier pdf et polices sans + Palatino

- pdf() : le mécanisme et les polices d'Adobe ne gèrent pas l'UTF-8 !
- Les expressions sont plus ou moins bien rendues selon les logiciels





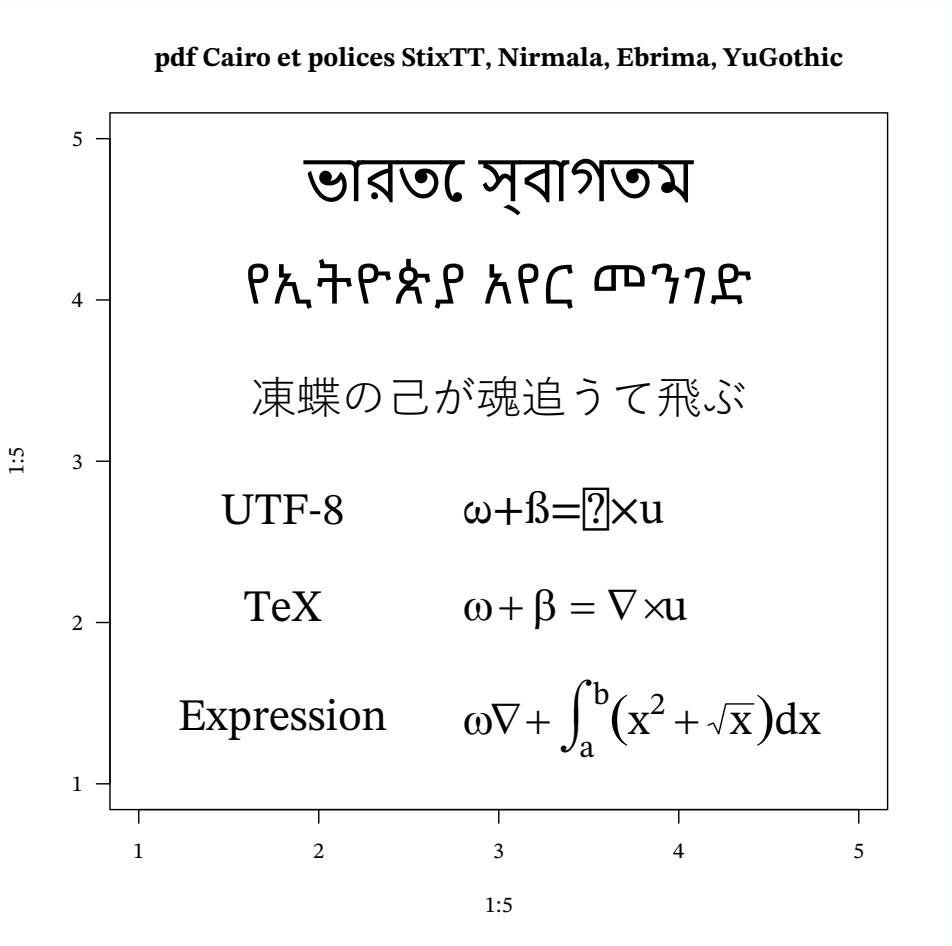
Fonction cairo_pdf() et polices système

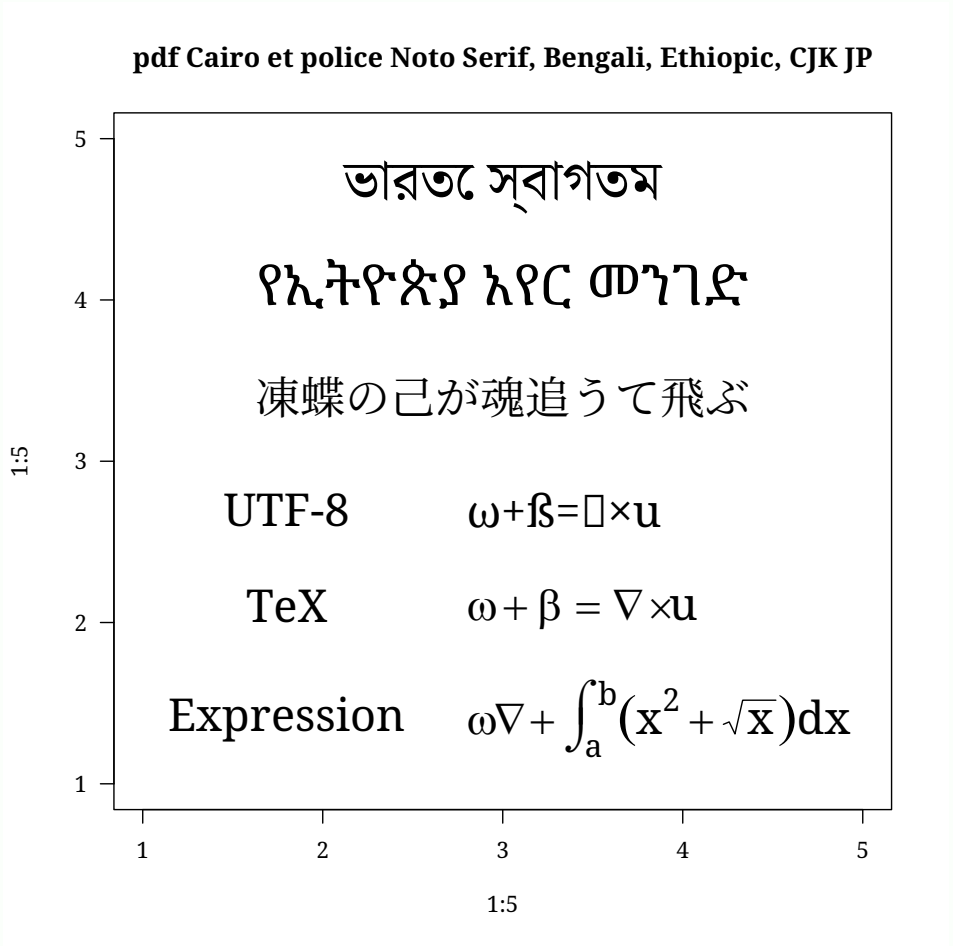
- Les logiciels récents de lecture de fichiers pdf ont évolué et lisent l'UTF-8
- cairo_pdf() utilise le même mécanisme que png(type = "cairo") avec des polices UTF-8 sans rapport avec les polices historiques d'Adobe
- cairo_pdf() enregistre les polices utilisées dans le fichier créé. Les fichiers sont plus volumineux mais sont lisibles partout.

```
famille2 <- "Noto Serif"
famille3 <- "Noto Serif, Bengali, Ethiopic, CJK JP"
cairo_pdf("pdfcairo-Noto.pdf", width = 7, height = 7, family = famille2)
op <- par(family = famille2)
plot(1:5, 1:5,  col = 0, las = 1, main = paste("pdf Cairo et police", famille3))
text(3.0, 4.7, "ভারতে স্বাগতম", cex = 2.5,          family = "Noto Serif Bengali")
text(3.0, 4.1, "የኢትዮጵያ አየር መንገድ", cex = 2.5,    family = "Noto Serif Ethiopic")
text(3.0, 3.4, "凍蝶の己が魂追うて飛ぶ", cex = 2, family = "Noto Serif CJK JP")
...
...
par(op)
dev.off()
```

Fichier pdf Cairo et polices multiples + Noto

- Le rendu à l'affichage est maintenant excellent
- Presque... Le code UTF-8 de Nabla n'est pas reconnu





Les polices pour les mathématiques

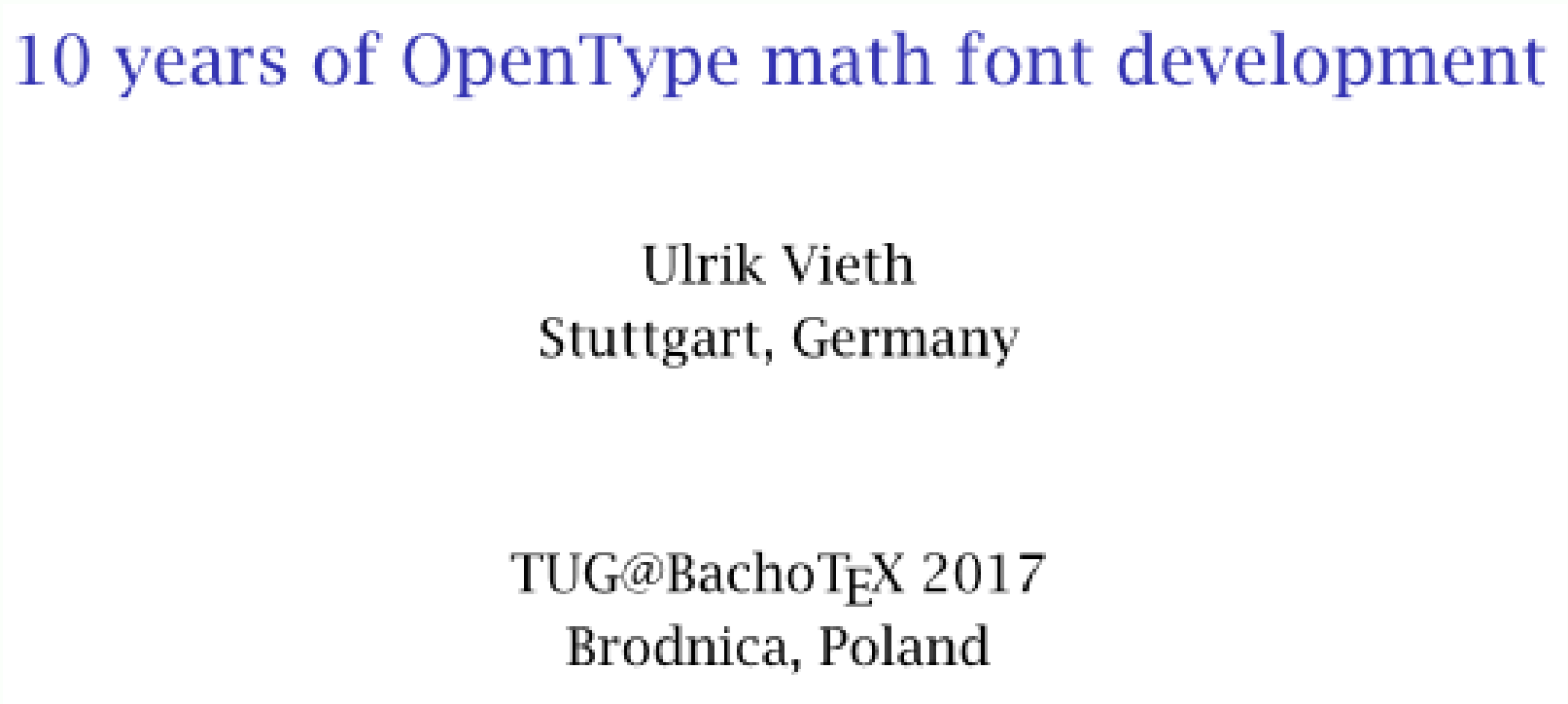
Les polices pour les mathématiques sont un monde à part.
R utilise les deux grands systèmes disponibles.

- Tex, Latex
  - Polices **T1**
  - Stockées dans l'arbre TeX de chaque distribution (Texlive, Miktex)
  - Téléchargeables à partir de ces distributions
  - Listées sur www.ctan.org
  - Environ une vingtaine de polices
- Word, Libre Office, Xelatex, Lualatex
  - Polices **OpenType** (TrueType + mathématiques)
  - Stockées dans le répertoire /Fonts
  - Téléchargeables depuis les sites des éditeurs
  - 15 polices en 2017

Conférence Bachotex

La conférence annuelle Bachotex réunit les meilleurs experts Tex + math

- http://www.gust.org.pl/bachotex/2018-pl
- La diapositive suivante provient de


10 years of OpenType math font development

Ulrik Vieth
Stuttgart, Germany

TUG@BachoTEX 2017
Brodnica, Poland

15 polices mathématiques Open Type en 2017

Choices of OpenType math fonts

- Choices of math fonts by release date
  - Cambria Math, since 2007
  - Asana Math, since 2008
  - Neo Euler, since 2009 (abandoned)
  - XITS Math, since 2010 (from STIX 1.0)
  - STIX Math, June 2013 (from STIX 1.1)
  - Latin Modern Math, May 2011
  - Minion Math, October 2011
  - Lucida Math, March 2012
  - TEX Gyre Pagella Math, May 2012
  - TEX Gyre Termes Math, Sept 2012
  - TEX Gyre Bonum Math, May 2013
  - TEX Gyre Schola Math, May 2014
  - (TEX Gyre) DejaVu Math, May 2015/16
  - Libertinus Math, since 2016
  - STIX Two Math, Dec 2016

Mes polices préférées pour les mathématiques

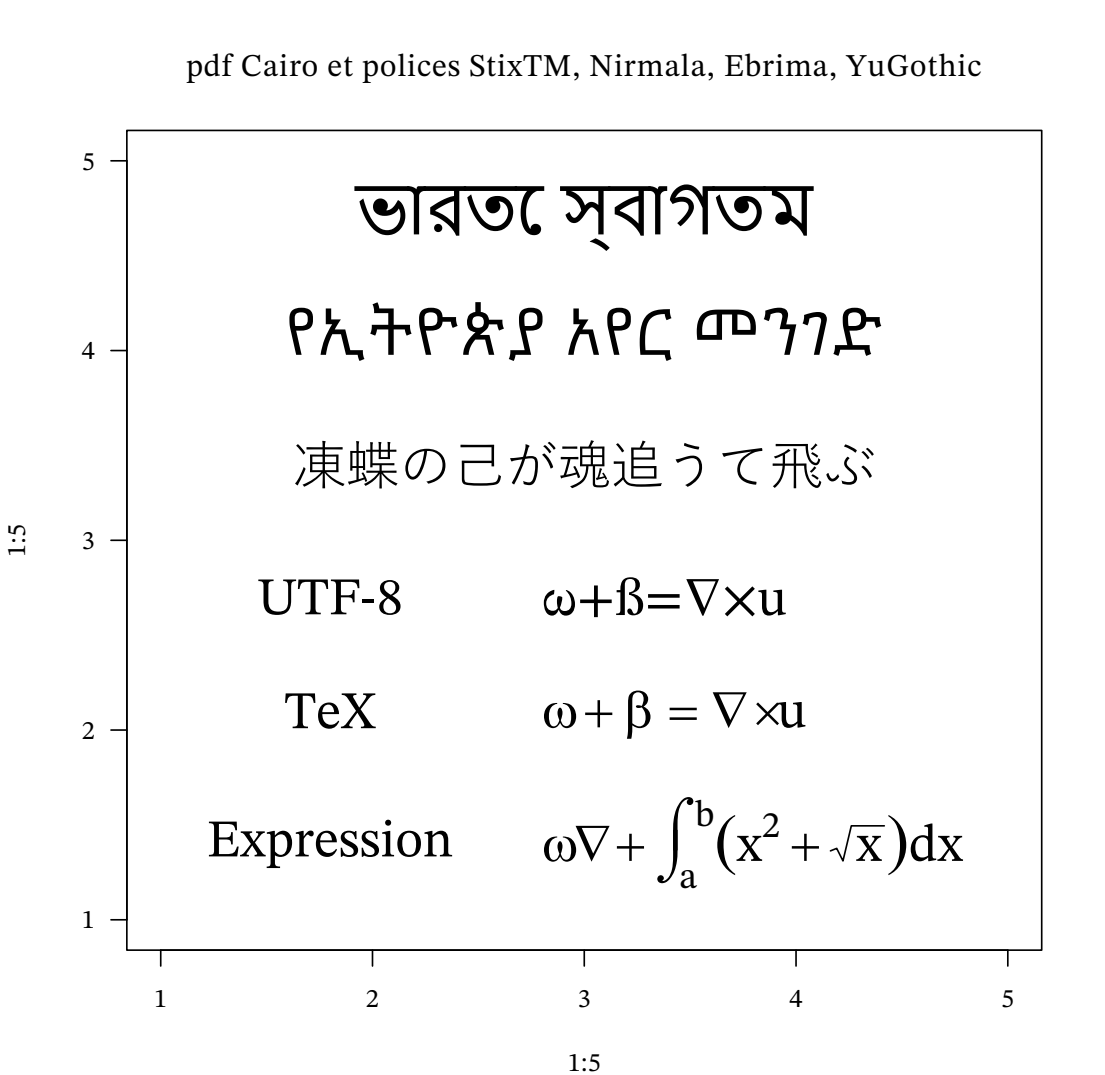
- Cambria Math est fournie avec Windows depuis Windows Vista
- Minion Pro est fournie dans Mac OSX
- Latin Modern Math est la police par défaut de Latex. On aime ou pas
- Tex Gyre Pagella Math améliore Palatino et rappelle les années 1990
- Tex Gyre DejaVu Math est la seule version sans-serif
- Libertinus Math est la version la plus aboutie de Times
- Stix Two Math est magnifique et la plus complète. Ma favorite !

Une solution approchée explicite, valide pour $3 < \kappa < \infty$, utilise les trois premiers termes du développement de Taylor de $S(\kappa, p, q)$. Soit $h = \frac{l}{u} = \frac{\text{logit}(p)}{\text{logit}(q)} > 1$

$$Q = S = h + \frac{h\left(h^2-1\right)u^2}{6\,\kappa^2} + \frac{h\left(h^2-1\right)\left(3h^2-7\right)u^4}{360\,\kappa^4} + o\left(\frac{1}{\kappa^6}\right)$$

Stix Two Math + les polices linguistiques

On a retrouvé Nabla !



Quelques liens

- http://www.gust.org.pl/bachotex/2018-pl
- https://www.facebook.com/BachoTeX-488489247924647/
- http://www.stixfonts.org/
- https://github.com/twardoch/schticks-fonts-ofl
- https://github.com/libertinus-fonts/libertinus
- http://www.gust.org.pl/projects/e-foundry/tex-gyre/whole
- http://www.gust.org.pl/projects/e-foundry/tg-math/download
- http://www.gust.org.pl/projects/e-foundry/latin-modern/download
- http://www.gust.org.pl/projects/e-foundry/lm-math/download

Merci pour votre attention !

patrice.kiener@inmodelia.com

Tél. : +33.9.53.45.07.38