# 1　環境とフォント

## 1　環境

このメモは,

```
> R.Version()$platform
[1] "x86_64-w64-mingw32"
```

で書いている．つまり，64 ビット Windows マシンである．R のバージョンは

```
> R.Version()$version.string
[1] "R version 3.4.1 (2017-06-30)"
```

であるが，R はしばしば更新されるので，www.okadajp.org/RWiki/?RjpWiki 等で最新バージョンを確認しよう．R はフリーなソフトウェアであるが，ライセンスの詳細は licence() で見ることができる．ただし英文で，またオンラインマニュアル等も英語で書かれているが，研究者や大学院生が遣うことを前提であれば問題ないであろう．たふぁし，学部生が使うことを考えると，少々困難な場合があるかもしれない．

「>」はコンソール画面に出てくるプロンプトで，入力を促す記号である．「>」の後に必要なコードを書くのが R の基本である．「>」が気に入らないならば，たとえば,

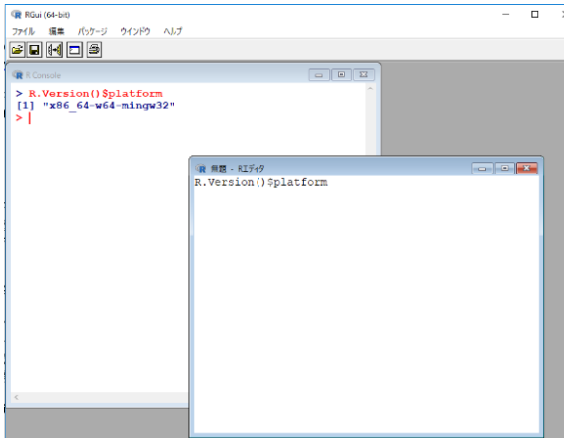
```
> options(prompt="R>")
R>
```

とすれば変更できる．プロンプトだけでなく各種のデフォルト設定（options() で分かる）を変更した後で，元のデフォルトに戻すためには,

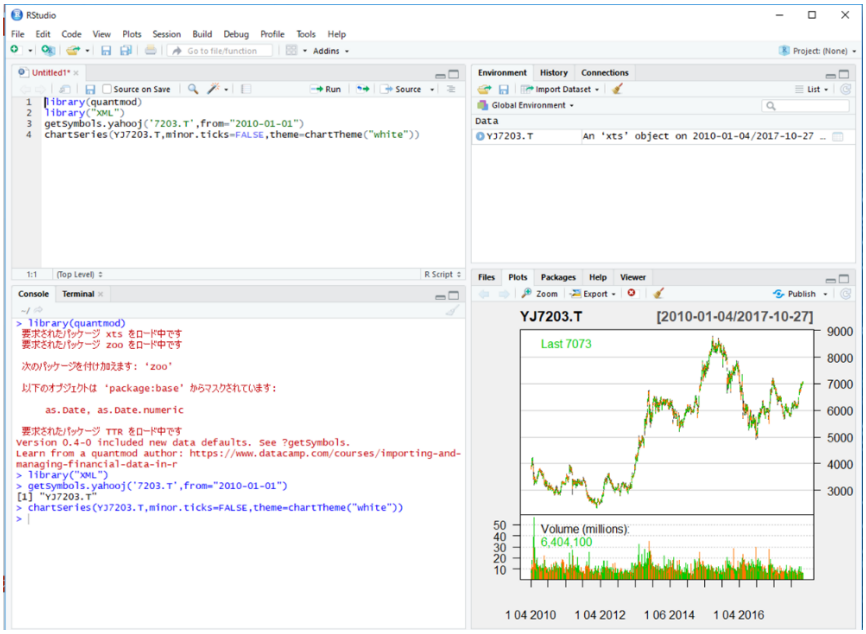
```
> options(op)
```

とすればよい．

R を起動すると，下記のような入力（コンソール）画面が現れる．エディター（スクリプト）を開いて，コードを記入して，Ctrl+R で実行すると，コンソールの「>」にそのコードが入力され実行される．コンソールに直接入力できるがエディター機能がないので，使いづらい．さらに，データ等を表示するためのグラフィックスは別の出力画面（デバイス）がある． 合計三つの画面があり，たいへん不便で，これだけで使う気になれない．もっとも，R には豊富な統計処理が装備されているので使っているが，商用のソフトとは違い，不便さは我慢するしかない．




最近では RStudio とう多少便利なインターフェースが利用できるようなった．www.rstudio.com/products/rstudio/download/からダウンロードできる．コンソール（左下の図），エディター（左上の図），出力画面（右下の図）が同じ画面上に現れるので便利であるが，基本的には R そのものである．なお，RStudio を R を前提にしているので，まず R をインストールしておかなければならない．




Rをインストールすると，始めから準備されているパッケージがあり，各種の基本的な関数が入っている．それ以外の特殊なパッケージは，必要に応じて別途インストールしなければならない．インストール済みの全てのパッケージは library() で確認できる．library()と入力すると，別画面「利用可能なパッケージ」にその一覧が現れる．

パッケージをインストールするとは CRAN（Japan ミラーサイトを選択）からダウンロードしてコンピュータの適当なデイレクトルに保存することである，そのパッケージを実際に使うためには library で実行可能な状態にしなけれ

ばならない．一度インストールしておけば，あとは実行可能な状態にするだけである．A というパッケージを初めて使うためには，

```
> install.packages("A")
> library(A)
```

とする．２回目以降は library(A)だけでよい．これと同じ動作は，起動したときの画面上でもできる．前者ではメニューの「パッケージ」→「パッケージのインストール」で CRAN から A をダウンロードし，後者では「パッケージ」→「パッケージの読み込み」で A を選択する．

最も基本的な Base パッケージに入っているものの一覧は，

```
> print(library(base),quote=FALSE)
[1] stats graphics grDevices utils datasets methods base
```

である．この中にある stats を調べるため，

```
> library(help="stats")
```

とすると，別画面「パッケージ stats のドキュメント」には各種の統計処理用のプログラムが表示される．たとえば，「AIC　Akaike's An Information Criterion」は赤池の情報量基準，「lm　Fitting Linear Models」は線形回帰モデルのように，どのような関数が入っているかはその略称あるいは名前からなんとなく分かるものが多い．通常使う統計プログラムはたいていこの中にある．

続いて，

```
> library(help="graphics")
```

とすると，別画面「パッケージ graphics のドキュメント」には各種のグラフが表示される．最も基本的なもの「plot　Generic X-Y Plotting」がである．しかし，ヒストグラムを描くプログラム histogram は見当たらないが，plot が(x,y)座標を与えて散布図等を描くこの plot は，ヒストグラムを描くためにもプログラムでもある．plot はよく用いるので，その使い方は十分に習得しておこう．

最後に，

```
> library(help="base")
```

を調べておこう，別画面「パッケージ base のドキュメント」には各種の基本的な関数が表示される．たてえば，date は現在の日付と時刻，eigen は固有値，sum はデータの和，solve は連立一次方程式の解法，などである．

search()は現在読み込まれている（ロード済み）パッケージの一覧を表示す

るコマンドである．Rを起動した時点では，読み込まれている（ロード済み），つまり実行可能なパッケージの一覧は，

```
> search()
[1] ".GlobalEnv"  "package:stats"  "package:graphics"
[4] "package:grDevices" "package:utils"   "package:datasets"
[7] "package:methods"  "Autoloads"  "package:base"
```

となっている．この時点で実行可能なパッケージはbaseの中の7個である．次に，いくつかのパッケージを読み込んでみよう．例えば，よく用いられるパッケージで，デフォルトのgraphicsよりも綺麗な，また複雑な図が描ける

```
> library(ggplot2)
```

を読み込んでみよう．すると，

```
> search()
 [1] ".GlobalEnv"  "package:ggplot2"  "package:stats"
 [4] "package:graphics"  "package:grDevices" "package:utils"
 [7] "package:datasets"  "package:methods"  "Autoloads"
[10] "package:base"
```

となって，ggplot2が追加されていることが確認できる．

なお，追加したパッケージggplot2に関して調べたいときは，読み込んだ後に，?ggplot2あるいはhelp(ggplot2)とすれば，英文のドキュメント(PDF)が現れる．また，このようなパーケージの開発者自身による解説書「Rグラフィックスクックブックーggplot2によるグラフ作成のレシピ集，オライリージャパン，2013年」もあるので参考になるだろう．

作業しているディレクトリは，特に指定しなければ，

```
> getwd()
[1] "C:/Users/###/Documents"
```

となっている．###はRの実行ファイル（Rのマークがあるファイル）があるホームディレクトリである．コンソール画面でメニューの「ファイル」から「ディレクトリの変更...」を選択しても分かる．また，そこでディレクトリを変更することもできる．プログラムやデータを保存するために別のディレクトリにまとめたい場合など，変更した方が分かりやすいこともあるだろう．その場合には作業ディレクトリを変更しよう．

同じ事は，コンソール画のコマンドで指定することができる．この場合は.、まずtcltkパッケージを読み込んで，

```
> setwd(paste(as.character(tkchooseDirectory(title="保存先を選択"),sep="",collapse="")))
```

とすると，開いたウィンドから選択して，フォルダーを指定する．

# ２　フォント

## ２．１　Windows のフォントを利用する

R で計算結果のグラフを作成しそれを word や Tex に貼り付けるのが一般的な使い方であろう．通常，R で文書まで作成することは前提にしていないと思う．従って，日本語フォントは word や Tex のようには使い勝手はよくなく，十分整備されていない．しかし，計算結果を資料とする場合，グラフの凡例などに日本語フォントを使いたいものである．

標準で出力画面用のフォントとして，

```
> str(windowsFonts())
List of 3
 $ serif: chr "TT Times New Roman"
 $ sans : chr "TT Arial"
 $ mono : chr "TT Courier New"
```

が定義されている．serif という名前で，英語フォントの Times New Roman が登録されている．TT は TrueType の略で，bold とか italic などの指定はない．英文ならこれで十分かもしれない．例えば，serif つまり Times フォントでは，

```
plot(0:1,0:1,type="n",axes=FALSE,xlab="",ylab="")
text(0.5,0.5,"matsuba 0123",family="serif",cex=2,pos=1)
```

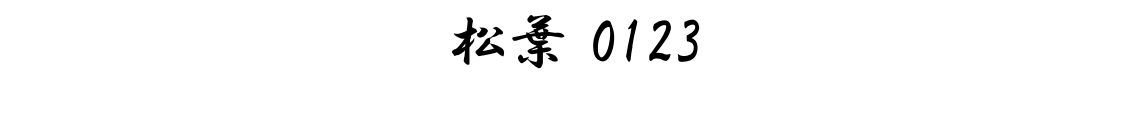

のようになる．family にはフォントを登録した名前"serif"を設定する．cex は文字の大きさの倍率．出力は，画面に出力された文字をメタファイルとしてコピーし，それを PowerPoint に貼って，適当な大きさにトリミングした結果を Word に貼り付けたものである．

なお，最初に text だけ入力しても何も表示されない．plot は一般的なグラフを描くための関数であるが，ここでは，何も描かない（type="n"で指定）白い画面（大きさは横縦ともに 1）だけを表示する．その後で text で文字をその上に書く．同じ位置に text で別の文字を描くと，重ね書きする．plot 以外の方法もあるので，それについては以下で説明する．

日本語の文章の場合は英文用のフォントだけでは不十分である．必要に応じて，使用したい日本語フォントを windowsFonts()を使って新たなフォントファミリーを適当な名前で登録し，その後で作図関数やグラフィックスパラメータの family で引用する．

例えば，

```
windowsFonts(myFont="HGS 行書体")
plot.new()
text(0.5,0.5,"松葉 0123",family="myFont",cex=2,pos=1)
```




である．HGS 行書体は Windows の Fonts ディレクトリに登録されている．最初に myFont（なんでも OK）という名前で HGS 行書体を登録しておいて，text では family=″myFont″で指定する．なお，ここでは，先の例より簡単な方法である plot.new()を用いた

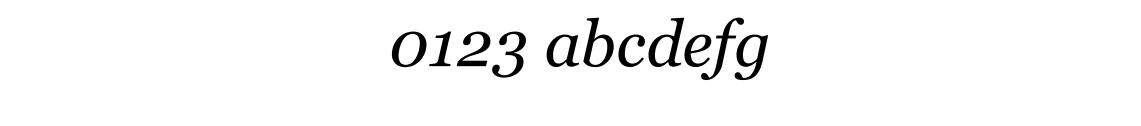
もう一例示しておく．

```
windowsFonts(myFont=″HG 平成角ｺﾞｼｯｸ体 W9″)
  ---前例のコードと同じ---
```


その他の windows のフォントをいろいろ試してみたが，すべてが正しく表示されないようである．

最後にもう一例

```
windowsFonts(myFont="Georgia Italic")
plot.new()
text(0.5,0.5,"0123 abcdefg",family="myFont",cex=2,pos=1)
```




この例は，物理の教科書などで重力の記号として $g$ が用いられるが，g のイタリック体は教科書のような $g$ にならなく迷うことがある．いくつかあるが，ここでは Georgia Italic を使った．

通常，報告書や論文で使うフォンは，MS 明朝体，MS ゴシック体の日本語フォントと，Times New Roman，数式でよく用いる symbol が多いであろう．そこで，

```
windowsFonts(JP1="MS Mincho",JP2="MS Gothic",
             JP3="Times New Roman",JP4="symbol")
```

と定義する．文字を書くために関数

```
jtext<-function(x,xp) {
  text(xp,0.8,x,cex=1,font=1,pos=3) #標準
  text(xp,0.7,x,cex=1,font=2,pos=3) #太字
```

```
  text(xp,0.6,x,cex=1,font=3,pos=3) #斜体
  text(xp,0.5,x,cex=1,font=4,pos=3) #太字斜体
  text(xp,0.4,"012Aa",font=1,cex=1,pos=3)
}
```

と定義しておこう．ここで，font の意味を確認すると，font=1 は標準体，font=3 は斜体などである．

```
jfont <- function(x) {
  plot.new()
  lines(c(0,1),c(1,1),col="black",lwd=2)
  lines(c(0,1),c(0.9,0.9),col="black",lwd=1)
  lines(c(0,1),c(0.4,0.4),col="black",lwd=2)
  par(family="JP1")
  text(0.1,0.9,"明朝",cex=1.5,font=1,pos=3)
  jtext(x,0.1)
  par(family="JP2")
  text(0.35,0.9,"ゴシック",cex=1.5,font=1,pos=3)
  jtext(x,0.35)
  par(family="JP3")
  text(0.6,0.9,"Times",cex=1.5,font=1,pos=3)
  jtext("012Aa",0.6)
  par(family="JP4")
  text(0.85,0.9,"Symbol",family="JP3",cex=1.5,font=1,pos=3)
  jtext("abcde",0.85)
}
jfont("松葉")
```

| 明朝 | ゴシック | Times | Symbol |
|---|---|---|---|
| 松葉 | 松葉 | 012Aa | abcde |
| **松葉** | **松葉** | **012Aa** | **abcde** |
| *松葉* | *松葉* | *012Aa* | *abcde* |
| ***松葉*** | ***松葉*** | ***012Aa*** | ***abcde*** |
| 012Aa | 012Aa | 012Aa | 012Aa |

次に数式の場合を考えよう．数式を書く場合，$\frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{(x-\mu)^2}{2\sigma^2}}$（Word の数式）のように表したいものである．また，Office に追加して用いることが多いアプリケーション MathType（有料）では，$\frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{(x-\mu)^2}{2\sigma^2}}$（デフォルト設定）となる．R と比べてみよう．

```
plot.new()
```

```
text(0.1,0.5,expression(paste(frac(1,sqrt(2*symbol(p)*symbol(s)^2))
,e^{-frac((x-symbol(m))^2,2*symbol(s)^2)})))
text(0.4,0.5,expression(paste(frac(1,sqrt(2*pi*sigma^2))," ",
italic(e)^{-frac((italic(x)-mu)^2,2*sigma^2)})),family="JP3")
```

$$\frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{(x-\mu)^2}{2\sigma^2}} \qquad \frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

①フォント family を指定しなければ，Times にならない（左の数式）.
②シンボル文字は例えば，symbol(p)でも sigma としても同じである.
③個別文字の斜体は italic(x)で指定する．italic(sigma)としも σ の斜体にはならない.
④文字間のスペースは 0 なので，見栄えよい数式にするためにはスペースを具体的に" "のように指定しなければならない.

もう一例示しておく.

```
plot.new()
text(0.5,0.5,expression(paste(frac(1,italic(n))," 
",sum(italic(x)[italic(i)],italic(i)==1,italic(n)))),family="JP3")
```

$$\frac{1}{n}\sum_{i=1}^{n} x_i$$

## ２．２　Tex 流のフォントを利用する

数式の記法としては Tex に慣れている方が多いと思う．その記法を上記 plotmath の記法に変換してくれる TeX()という関数がパッケージ latex2exp により提供されている.これはあくまで plotmath の枠内で数式を書くものであり，新しい機能を追加するものではないが，plotmath の独自記法を覚える必要がないという点で便利である.

```
install.packages("latex2exp")
library(latex2exp)
```

詳細は GitHub の stefano-meschiari/latex2exp に解説がある.
これを使えば，このページの最初の例は次のように書くことができる。¥¥（バックスラッシュ）は 2 重に書く必要がある.

```
plot.new()
text(0.5,0.5,TeX("$¥¥frac{1}{¥¥sqrt{2¥¥pi¥¥sigma^2}}¥¥exp^{
-¥¥frac{(¥¥mathit{x}-¥¥mu)^2}{2¥¥sigma^2}}$"),family="JP3")
```

$$\frac{1}{\sqrt{2\pi\sigma^2}}\exp^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

ただし，ここでも文字は自動的にはイタリック体にならないので，明示的に¥¥mathit{x}のように書く必要がある.

latex2exp でサポートされている例を以下に示す.

latex2exp_supported(plot=TRUE)

| | | | | | |
|---|---|---|---|---|---|
| $\%$ | % | $\pm$ | ± | $\bigcap_{x}^{y}$ | $\bigcap_{x}^{y}$ |
| $\,$ | | $\prime$ | ′ | | |
| $\;$ | | $\propto$ | ∝ | | |
| $\aleph$ | ℵ | $\rbrack$ | ] | $\bigcup_{x}^{y}$ | $\bigcup_{x}^{y}$ |
| $\approx$ | ≈ | $\Re$ | ℜ | | |
| $\ast$ | ∗ | $\rightarrow$ | → | | |
| $\bar{x}$ | x̄ | $\Rightarrow$ | ⇒ | $\frac{x}{y}$ | $\frac{x}{y}$ |
| $\bullet$ | • | $\sim$ | ∼ | | |
| $\cdot$ | · | $\small{x}$ | x | | |
| $\cdots$ | ⋯ | $\spadesuit$ | ♠ | $\int_{x}^{y}$ | $\int_{x}^{y}$ |
| $\clubsuit$ | ♣ | $\subset$ | ⊂ | | |
| $\cong$ | ≅ | $\subseteq$ | ⊆ | | |
| $\degree$ | ° | $\supset$ | ⊃ | $\lim_{x}$ | $\lim_{x}$ |
| $\diamondsuit$ | ♦ | $\supseteq$ | ⊇ | | |
| $\div$ | ÷ | $\surd$ | √ | $\overset{x}{y}$ | $\overset{x}{y}$ |
| $\dot{x}$ | ẋ | $\TeX$ | TEX | | |
| $\equiv$ | ≡ | $\textbf{x}$ | **x** | $\prod_{x}^{y}$ | $\prod_{x}^{y}$ |
| $\exists$ | ∃ | $\textit{x}$ | *x* | | |
| $\forall$ | ∀ | $\tilde{x}$ | x̃ | $\sqrt[2]{x}$ | $\sqrt[2]{x}$ |
| $\geq$ | ≥ | $\times$ | × | | |
| $\hat{x}$ | x̂ | $\tiny{x}$ | x | $\sum_{x}^{y}$ | $\sum_{x}^{y}$ |
| $\heartsuit$ | ♥ | $\underline{x}$ | x̲ | | |
| $\Im$ | ℑ | $\vee$ | ∨ | | |
| $\in$ | ∈ | $\wedge$ | ∧ | | |
| $\infty$ | ∞ | $\widehat{x}$ | x̂ | | |
| $\LaTeX$ | LATEX | $\widetilde{x}$ | x̃ | | |
| $\lbrack$ | [ | $\wp$ | ℘ | | |
| $\ldots$ | … | | | | |
| $\leftarrow$ | ← | | | | |
| $\Leftarrow$ | ⇐ | | | | |
| $\leq$ | ≤ | | | | |
| $\mathbf{x}$ | **x** | | | | |
| $\mathit{x}$ | *x* | | | | |
| $\mathrm{x}$ | x | | | | |
| $\neg$ | ¬ | | | | |
| $\neq$ | ≠ | | | | |
| $\normalsize{x}$ | x | | | | |
| $\notin$ | ∉ | | | | |
| $\nsubset$ | ⊄ | | | | |
| $\oplus$ | ⊕ | | | | |
| $\oslash$ | ⊘ | | | | |
| $\otimes$ | ⊗ | | | | |
| $\partial$ | ∂ | | | | |
| $\perp$ | ⊥ | | | | |

以下は数式の例である.

latex2exp_examples()

| | |
|---|---|
| `$\\alpha_{\\beta}^{\\gamma}$` | $\alpha_{\beta}^{\gamma}$ |
| `$\\frac{\\partial \\bar{x}}{\\partial t}$` | $\frac{\partial \bar{x}}{\partial t}$ |
| `$\\sum_{i=1}^{10} x_i \\beta^i$` | $\sum_{i=1}^{10} x_i \beta^i$ |
| `$\\prod_{i = 1}^{100} x^i$` | $\prod_{i=1}^{100} x^i$ |
| `$\\left(\\int_{0}^{1} \\sin(x) dx \\right)$` | $\left(\int_{0}^{1} \sin(x) dx\right)$ |
| `ine structure constant is $\\alpha \\approx \\frac{1}{137}$.` | The value of the fine structure constant is $\alpha \approx \frac{1}{137}$. |
| `$\\nabla \\times \\bar{x}$ and $\\nabla \\cdot \\bar{x}$` | $\nabla \times \bar{x}$ and $\nabla \cdot \bar{x}$ |
| `$\\sqrt[\\alpha\\beta]{x_i^2}$` | $\sqrt[\alpha\beta]{x_i^2}$ |
| `\\textbf{Bold} and \\textit{italic} text!` | **Bold** and *italic* text! |
| `$\\left{\\left(\\left[BRACES\\right]\\right)\\right}$` | $\left\{\left(\left[\mathrm{BRACES}\right]\right)\right\}$ |
| `Whitespace compliant: $x ^ 2 \\times \\sum_ 0 ^ 1 y _ i$` | Whitespace compliant: $x^2 \times \sum_{0}^{1} y_i$ |
| `Numbers: $0.05$, $0.03$, $0.005^{0.002}_{0.01}$` | Numbers: 0.05, 0.03, $0.005^{0.002}_{0.01}$ |

## ２．３　showtext でフォントを使う

Google が定義した fonts(http://www.google.com/fonts)も利用できる

```
install.packages("jsonlite")
library(showtext)
```

例えば，Google の Great Vibes 書体を用いると，

```
font_add_google("Great Vibes")
plot.new()
showtext_begin()
text(0.5,0.5,"matsuba 0123",family="Great Vibes",cex=3)
showtext_end()
```


となる.

## ２．４　PDF に出力する

PDF に出力する場合は，あらかじめ Japan1 などの名前で日本語フォントが登

録されています．使用方法は Using system fonts in R graphs や GitHub の yixuan/showtext を参考．

```
> 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"
```

日本語フォントが収録されているが，たとえば，Japan1Ryumin を調べると，

```
> str(pdfFonts()$Japan1Ryumin)
List of 5
 $ family     : chr "Ryumin-Light"
```

となっているので，標準的なモリサワのリュウミンライトである．また，Japan1GothicBBB は中ゴシック BBB で，よく使われているゴシック体である．"Times"は"serif"と同じようである．

PostScript プリンタによっては，Ryumin-Light フォントを内蔵していないことがあり，その場合に Japan1Ryumin を指定すると(画面表示はできるが)印刷はできないかもしれない．PDF で「ファイル」→「プロパティ」→「フォント」で確認すると，実際に使われているフォントは KozMinPr6N-Regular で，埋め込まれていないことがわかる．そのような場合にも，Ghostscript がインストールされていれば，Ryumin を使いたいのであれば，embedFonts(...) でフォントを埋め込むことで可能になる．help(embedFonts)を参照．

例えば，

```
pdf("pdf0.pdf", family="Times", width=10, height=10)
plot.new()
text(0.5,0.6,expression(paste(frac(1,sqrt(2*symbol(p)*symbol(s)^2)),e^
{-frac((x-symbol(m))^2,2*symbol(s)^2)})))
text(0.5,0.5,expression(paste(frac(1,sqrt(2*pi*sigma^2)),"          ",
italic(e)^{-frac((italic(x)-mu)^2,2*sigma^2)})))
dev.off()
```

$$\frac{1}{\sqrt{2\pi\sigma^2}}\mathrm{e}^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

$$\frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

のようになる．ここでも自動的にはイタリック体にならない．

ps.options( family="Japan1HeiMin", height=2.5, width=3.1, horizontal=FALSE, onefile=FALSE, paper="special", pointsize=8 )

http://ofmind.net/doc/r-intro-lecture

```
n <-2
TestData <- data.frame(Group = paste0("TEST", 1:n),
Data1 = sample(1:500, n, replace = TRUE),
Data2 = sample(200:300, n, replace = TRUE),
Data3 = sample(c("yes", "no"), n, replace = TRUE))
formattable(TestData)
```

| Group | Data1 | Data2 | Data3 |
|---|---|---|---|
| TEST1 | 450 | 277 | yes |
| TEST2 | 322 | 253 | no |