

KETpic v4.3.1 ltd コマンド一覧 for R

PD

プロットデータ

1 R についての注意

1. KETpic の読み込みには次を実行する.

```
load("C:/work/ketpic.Rdata") (C:/work/は作業フォルダ名)
```

※ ディレクトリの変更は `setwd("c:/work")`

2. 文字列は " (ダブル) で囲む

※ 文字列の中に文字列を入れるときは ' と " を入れ子に使う.

3. 関数などを引数とするときは, 文字列とする.

例) `G<- Plotdata("x^2", "x=c(0,1)")`

4. \ (バックスラッシュ) を出力するには 2 つ並べてかく.

5. 注釈は #

6. 数と文字列の変換

```
as.character(数)
```

```
as.numeric(文字列)
```

```
eval(parse(text=文字列)))
```

```
sprintf(書式, 数) 書式付き文字列
```

7. 異なる型のデータからリストを作るには list を用いる.

作成 `L<- list(... , ...);`

取り出し `A<- L[[i]]`

部分リスト `L[V]` (V はベクトル)

置き換え `L[[i]]<- ...`

長さ `length(L)`

追加 `L<- c(L1, L2)`

結合 `c(L1,L2,...)`

空リスト `list()`

タイプを見る `is.list(L)` (論理値) または `mode(L)`

2 R のための追加コマンド

`Member(D, L)` D が L の要素であれば true, そうでなければ false を返す.

※ L はベクトルまたは list

`Flattenlist(L)` L を平準化して単層のリストを作る

`Mixdisp(list)` list の要素を画面に簡易表示

`Op(N, Data)` Data の N 番目の要素 (Data は文字列, ベクトル, list)

`Assign(式, 変数名, 値, ...)`

変数名 (文字列) に値を割り当てた文字列を返す

※ 値は, 数, 文字列, 数行列, Scilablist

例) `A<- 0.4; B<- c(2,1)`

`Fn<- Assign("A*x^2+B(1)*y^2","A", A, "B", B)`

Assignset(変数名, 値, ...) 割り当て変数テーブルをセットする
 例) Assignset("A", 0.4, "B", c(2,1), "C", list(...), "D", "string")

Assignset("?"+変数名) 変数名の値を返す
 例) Assignset("?A")

Assignset() 割り当て変数テーブルを初期化

Assignadd(変数名, 値, ...) 割り当て変数テーブルに追加する
 例) Assignadd("C", 0.4, "D", c(2,1))

Assignrep(変数名, 値, ...) 割り当て変数テーブルを置き換える
 例) Assignrep("C", 0.8)

Prime(文字列) 文字列の最後に " をつける
 例) Prime("A")
 例) Prime() (" だけを出力)
 ※ Assign("A ") としてもよい.

XMIN, XMAX, YMIN, YMAX ウィンドウ範囲 (デフォルト $-5 \leq x \leq 5, -5 \leq y \leq 5$)

Ptne(), Ptnw(), Ptsw(), Ptse() フレーム枠の各点

ThisVersion Ketpic のバージョン

Fracform(x{, 分母の最大値 }) x に近い分数 (文字列) を返す
 ※ 分母の最大値のデフォルトは 100000
 例) Fracform(c(2.36))

Dotprod(v1,v2) 内積

Crossprod(v1,v2) 外積

Derivative(関数文字列, 変数名, { 値ベクトル }) 関数の微分係数を求める.
 例) Derivative("x^2+y", c("x","y"), c(2, 3))

Integrate(関数文字列, 変数文字列, 積分区間 (数リスト)) 関数の定積分を求める.
 例) Integrate("sin(x)","x",c(0,pi))
 ※ 区間 (リスト) は積分を分けて計算するときに指定
 Integrate("Fn(x)","x",c(-2,0,3))

3 設定コマンド

3.1 基本

Setwindow(c(xmin, xmax), c(ymin, ymax)) ウィンドウ範囲を設定
 例) Setwindow(c(-pi, pi), c(-1.5, 1.5))
 ※ XMIN, XMAX, YMIN, YMAX で値を得られる.

Setscaling(ratio) 縦の横に対する比を ratio に設定 (デフォルトは 1)
 例) Setscaling(2)
 ※ ウィンドウも連動

Setax(線種, 横軸名, 位置, 縦軸名, 位置, 原点名, 位置)	座標軸を設定 (引数 7 個)
	例) Setax("a", "t", "s", "u", "w", "O", "nw")
	※ 線種は d : line, a : arrow
	※ arrow のとき "a0.5" のように, 矢印の大きさを指定できる.
	※ "" とすると, 現在の設定を変更しない.
	※ 以降が "" のとき省略できる. また途中からも指定できる
	例) Setax("a")
	例) Setax(6, "O", "se") (6 番目から指定)
	※ 位置は "n", "s", "e", "w", "ne", "nw", "se", "sw"
	※ "s2w3" のように微小移動量を付加してよい.
	※ 空引数のとき, 現在の設定値を表示
Setorigin(点)	座標軸の原点を指定 (デフォルトは (0, 0))
	※ 空引数のとき, 現在の設定値を表示
Setpen(倍率)	線の太さを指定 (標準からの倍率で)
	例) Setpen(1.5)
	※ 空引数のとき, 現在の設定値を表示
Setpt(倍率)	Drwpt の点の大きさを指定 (標準からの倍率で)
	※ 空引数のとき, 現在の設定値を表示
Setmarklen(倍率)	目盛りの大きさを指定 (標準からの倍率で)
	※ 目盛りの大きさは微小移動量の単位としても用いられる.
	※ 空引数のとき, 現在の設定値を表示
Setunitlen("単位長")	単位長を指定する
	※ Beginpicture("") とすると指定された単位長が使われる.
	※ 空引数のとき, 現在の設定値を表示
Setarrow(鋸の大きさ {, 開き角 {, 鋸位置 {, 太さ }} {, 形と位置 })	矢印の形状を指定する
	例) Setarrow(0.5, 1, 1, 0.7, "tf")
	※ デフォルト 大きさ 1, 角度 18°, 位置は終点
	※ 5 以下の開き角を指定したときは, 18° からの倍率とする
	※ 形状 "l" : ライン "f" : 塗り (デフォルト)
	※ 位置微調整 "c" : 中央 "b" : 下 "t" : トップ (〃)
	※ 空引数のとき, 現在の設定値を表示
Ketinit()	定数をデフォルトに初期化

3.2 空間 (平行投影)

Setangle(θ, φ)	角度 (°) を指定
	※ デフォルト値は $\theta_i-60, \varphi_i-30$
	※ 空引数のとき, 現在の設定値を表示
Initangle()	デフォルト値に戻す

3.3 空間 (一点投影)

Setpers(注視点, 視点)	一点投影の FocusPoint, EyePoint を指定
	※ デフォルト値は FocusPoint<- c(0,0,0), EyePoint<- c(5,5,5)
Setpers()	現在の FocusPoint, EyePoint を表示
	※ 空引数のとき, 現在の設定値を表示
SetstereoL($R, \theta, \varphi, \Delta$)	原点を注視点として, 空間極座標により定まる左目の位置を 視点にセット (Δ は目の間隔)

SetstereoR($R, \theta, \varphi, \Delta$) 原点を注視点として、空間極座標により定まる右目の位置を
視点にセット (Δ は目の間隔)

4 プロットデータの作成

4.1 平面図形

Plotdata(関数, 範囲, オプション)

関数のグラフの PD を作成

例) `G1<- Plotdata("sin(x)", "x= c(-2*pi, 2*pi)")`

※ 範囲を "x" とすると, XMIN から XMAX にとる.

※ x 以外の変数を使うときは関数に使われていないかを注意.

※ オプション

"N=..." 点の個数

"E=c(...)" 除外点のリスト

"E=関数" 関数の 0 点は除外

"D=..." 連続限界値 (これ以上離れたら結ばない)

※ デフォルトは N=50, D=Inf

例) `G1<- Plotdata("1/x", "x", "N=200", "E=c(0)", "D=1")`

例) `G2<- Plotdata("1/((x-1)*(x+2))", "x", "E=(x-1)*(x+2)")`

Listplot(点のベクトルまたは列または list)

折れ線の PD を作成. ただし、点は線分で結ぶ.

例) `G2<- Listplot(c(c(3,2),c(5,4)))`

例) `G2<- Listplot(c(3,2),c(5,4))`

Lineplot(点 A, 点 B{ , 長さ, 半直線 })

線分 AB を延長した線分の PD を作成

例) `G3<- Lineplot(c(3,2),c(5,4))`

例) `G4<- Lineplot(A, B, "+")`

半直線 AB (B 側に延長)

※ 長さのデフォルトは片側 100

Paramplot(パラメトリック関数, 範囲, オプション)

パラメトリック関数のグラフの PD を作成

※ t 以外の変数を使うときは関数に使われていないかを注意

例) `G3<- Paramplot("c(cos(t), sin(t))", "t=c(0, 2*pi)")`

Rotatedata(PD または点, 角度 {, 中心 })

平面の PD を回転した PD を作成

例) `G4<- Rotatedata(G1, pi/4)`

Translatedata(PD, x 方向 y 方向)

PD を平行移動した PD を作成

例) `G5<- Translatedata(G1, 3, -1)`

Scaledata(PD, x 方向, y 方向 {, 中心 })

PD を拡大 (縮小) した PD を作成

例) `G6<- Scaledata(G1, 2, 1/3)`

Reflectdata(PD, 点) 点対称移動した PD を作成

Reflectdata(PD, c(点 1, 点 2))

線対称移動した PD を作成

例) `G7<- Reflectdata(G1, c(0,0))`

例) `G8<- Reflectdata(G1, c(c(0,0), c(0,1)))`

Pointdata(PD, ...) PD の節点の list を作成
 例) G9=Pointdata(G1)
 ※ Drwpt(G9) などで、点のプロットができる。

Circledata(中心, 半径 {, オプション }) 円の PD を作成
 例) G10<- Circledata(c(3,1), 2)
 ※ オプション
 "R=..." θ の範囲
 "N=..." 点の個数
 例) G10a<- Circledata(c(3,1), 2, "R=c(0, pi/2)")
 例) G10b<- Circledata(c(3,1), 2, "N=100")

Framedata(P, dx{, dy}) 点 P を中心に $\pm dx, \pm dy$ の矩形 (dy を省略すると dy<- dx)
 例) G3<- Framedata(c(3, 1), 0.5)

Framedata(c(x_1, x_2), c(y_1, y_2))
 $x_1 \leq x \leq x_2, y_1 \leq y \leq y_2$ の矩形の PD を作成 (右上から反時計)
 ※ 引数が空のとき, Setwindow で指定した枠
 例) G1<- Framedata(c(-2, 3), c(1, 4))
 例) G2<- Framedata()

Hatchdata(パターン文字 (list){, 開始点 }, (閉) 曲線の列 {,kaku{,haba} })
 パターンと一致する領域を斜線塗りする PD を作成
 例) G1<- Hatchdata(list("io"), list(g1,g2), list(g3))

(i : 内部, o: 外部)

※ 開始点が指定されたとき
 (仮想的に) その点を通る斜線から描き始める

※ kaku は斜線の傾き (def=45), haba は間隔 (def=1)
 例) G2<- Hatchdata(list("ii"), O, list(G1), -45, 1.5)

※ 曲線リスト内の PD は隣接の順に指定
 ※ 閉じていないとき
 (1) 方向 "s", "n", "w", "e" を指定する
 (2) 窓枠とちょうど 2 点で交わる場合、領域の点を指定
 (3) 指定しなければ端点を直線で結ぶ。
 例) G3<- Hatchdata(list("ii"), list(g1,"s"), list(g2, c(3,0)))

Hatchdata(領域の点) {, 開始点 }, (閉) 曲線 list の列 {,kaku{,haba} })
 点 (のどれか) が含まれる領域を斜線塗り
 例) F4<- Hatchdata(list(A,B,C), list(G1), list(G2,G3))
 ※ 包含パターンが点 A, B, C のどれかと一致する領域
 を斜線塗り (領域は隣接するものとする)

Enclosing(PD リスト {, 始点の近くの点 })
 PD 列の直近の交点を結んで閉曲線を作成
 例) G2<- Enclosing(list(G1, invert(G2), G3), c(2,1))
 G1 と (最後の)G3 の交点のうち, c(2,1) に近い点から始める
 ※ 交点が 1 個の場合は, 点を省略してよい。

Dotfilldata(パターン文字列 (リスト) {, 開始点 }, (閉) 曲線 PD リストの列 {, 濃さ })
 パターンと一致する領域を点描する PD を作成
 例) Fd<- Dotfilldata("ii",list(G1),list(G2),0.7)
 ※ 濃さ d は $0 < d \leq 1$ (デフォルトは 0.5)
 ※ 書き出しは, Drwpt を用いる。

Arrowdata
 矢印の PD を作成 (Arrowline 参照)
 ※ やじりは塗りつぶさず, 線データのみ

Arrowheaddata
 やじりだけの PD を作成 (Arrowhead 参照)

Bowdata(点 A, 点 B{, 曲がり {, 切り } })

弓形の PD を作成

- ※ 曲がり：弧の曲がり（デフォルトは 1）
- ※ 切り：中央に入れる切りの長さ（デフォルトは 0）
- ※ 点 A から B に反時計まわりに弧をかく

例) G<- Bowdata(c(2, 1), c(3, 4), 0.8, 0.5)

Bowmiddle(弧データ) 弓形の midpoint を返す

Splinedata(点データ {, オプション })

spline 曲線の PD を作成

- ※ 点データはリストまたは PD
- ※ オプション：
 - "N<- 点の個数"（デフォルトは 50）
 - "C"（閉曲線でスムーズにつなぐ）

例) Fs<- Splinedata(PL, "N<- 200", "C")（PL は点データ）

Anglemark(A, B, C {, サイズ })

∠ABC の間の角度記号を作成

- ※ BA から BC へ反時計回りに描く
- ※ サイズのデフォルトは 0.5

Paramark(A, B, C {, サイズ })

∠ABC の間の角度記号（平行四辺形）を作成

4.2 空間図形

Spaceline(空間点のベクトルまたは list)

空間点を結ぶ線分の PD3d を作成

例) G1<- Spaceline(c(c(3, 2, 1), c(5, 6, 6)))

Spacecurve(関数, 範囲, オプション)

空間曲線の PD3d を作成

例) G2<- Spacecurve("c(cos(t), sin(t), t)", "t<- c(0, 2*pi)")

Rotate3data(PD3, v1, v2 {, 中心 })

PD3 を v1 が v2 に重なるように回転した PD3d を作成

例) G2<- Rotate3data(G1, c(1, 0, 0), c(1, 2, 3))

※ PD3 は list でもよい（この場合は list を返す）

Rotate3datac(PD3, 回転軸, 角度 {, 中心 })

PD3 を回転軸のまわりに回転した PD3d を作成

例) G2<- Rotate3datac(G1, c(0, 0, 1), %pi/4)

※ PD3 は list でもよい（この場合は list を返す）

Translata3data(PD3, 移動ベクトル v)

PD3 を v だけ移動した PD3d を作成

例) G2<- Translate3datac(G1, c(3, 2, 1))

例) G2<- Translate3datac(G1, 3, 2, 1)

※ PD3 は list でもよい（この場合は list を返す）

Xyzax3data(x 範囲, y 範囲, z 範囲)

座標軸の PD3d の list を作成

Projpara(PD3 列または list)

空間曲線の平行投影による射影 PD(2d) を作成

Projpers(PD3 列または list)

空間曲線の一点投影による射影 PD(2d) を作成

Skeletonparadata(空間曲線 list1, 空間曲線 list2{, 大きさ {, 遠近の閾値 }})
 平行投影で list1 から list2 により隠される部分を除いた残りの平面
 PD 列 (スケルトンデータ) を作成

Skeletonpara3data(空間曲線 list1, 空間曲線 list2{, 大きさ {, 遠近の閾値 }})
 平行投影で list1 から list2 により隠される部分を除いた残りの空間
 PD 列 (スケルトンデータ) を作成

Skeletonpersdata(空間曲線 list1, 空間曲線 list2{, 大きさ {, 遠近の閾値 }})
 一点投影で list2 による list1 のスケルトンデータ (2D) を作成

Skeletonpers3data(空間曲線 list1, 空間曲線 list2{, 大きさ {, 遠近の閾値 }})
 一点投影で list2 による list1 のスケルトンデータ (3D) を作成

Embed(平面曲線 (リスト), 埋め込み関数)
 埋め込み関数により空間曲線を作成

```
例) deff("Out$<$- Fun(x,y)", "Out$<$- c(x,y,0)")
      G1$<$- Listplot(c(0,0),c(3,2))
      G1_3d$<$- Embed(G1, Fun)
```

4.3 多面体の描画

Phcutdata(頂点リスト VL, 面添字リスト FL, 平面データ PlaneD)
 多面体を平面で切ったときの多面体と切断面の 3d リストを作成
 ※ PlaneD (平面) の形式

```
"a*x+b*y+c*z-d", "a*x+b*y+c*z<- d"
```

 (x, y, z をクリアしておく)
 または

```
"c(a, b, c, d)"
```



```
"list(a, b, c, P)"
```

 (点 P を通る)
 ※ 切断面はリストの最後の要素
 例)

```
VL<- list(c(0, 0, 0), c(1, 0, 0), c(0, 1, 0), c(0, 0, 1))
      FL<- list(c(1, 2, 3), c(1, 2, 4), c(1, 3, 4), c(2, 3, 4))
      PL<- Phcutdata(VL, FL, "c(1, 1, 1, 3)")
      Windisp(PL)
```

Phcutoffdata(VL, FL, PlaneD, 符号)
 PlaneD で切断された部分多面体の 3d データリストを作成
 ※ 符号は "+" または "-"
 例)

```
PL<- Phcutoffdata(VL, FL, "x+y+(z-1/2)", "+")
```


 ※ PhVertexL(), PhFaceL() で頂点, 面リストを取り出せる.

Phparadata(VL, FL) 陰線処理をした多面体の PD3d (平行投影) を作成

Phpersdata(VL, FL) 陰線処理をした多面体の PD3d (一点投影) を作成
 ※ PhHiddenData() で陰線の PD を取り出せる.

Phsparadata(面 datalist) 複数の多面体の PD3d (平行投影) を作成 (陰線処理)

Phspersdata(面 datalists) 複数多面体の PD3d (一点投影) を作成 (陰線処理)
 ※ 面 datalist は list(VL, FL), または, その list
 ※ 面を点で直接指定するときには VL<- list() とする.
 例)

```
Fd<- list(list(),list(c(3,2,1),c(0,0,0),c(c(1,2,4)))
```


 ※ PhHiddenData() で陰線の PD を取り出せる.

Phsrawparadata(面 datalist), Phsrawpersdata(面 datalist)
 複数の多面体の PD3d を作成 (陰線処理をしない)

Facesdata(面 datalist {, 追加曲線 PDlist }, 射影のタイプ)
 面の辺 (と追加曲線) を面により陰線処理
 ※ 射影のタイプは "para", "pers", "rawpara", "rawpers"

Faceremovaldata(面 datalist, 曲線 PDlist, 射影のタイプ)

曲線を面により陰線処理

5 データの書き出し

5.1 基本コマンド

Windisp(PD 列または list)

画面を開き, PD 列を表示 (確認のため)

例) Windisp(G1, G2)

例) Windisp(list(G1,G2))

WindispT(PD 列または list {, オプション })

画面を開き, PD 列を表示 (確認のため) 図を重ねて表示する. 事前に WindispT()

例) WindispT(G1, G2 color="red",width=1,new=TRUE)

例) WindispT(list(col="blue",border="white",G1),new=TRUE)

(閉曲線 G1 を塗る)

例) WindispT(list(col="blue",border="white",density=200,G1,G2),new=TRUE)

(閉曲線 G1,G2 を塗る.density は内側を線分で塗りつぶす場合のパラメータ)

Openfile("ファイル名",{,"単位長","SF=ソースファイル名"})

書き出し用ファイルを開く (デフォルトは画面)

例) cd('C:/TeXF/');

Openfile('fig.tex');

Openfile('fig.tex','fig.r');

※ 単位長を指定すると Beginpicture("単位長") まで書き出す

※ ソースが同一フォルダにあるときは, SF の指定は不要

Beginpicture("単位長") picture 環境を始める.

例) Beginpicture("1cm")

例) Beginpicture("2*10/12cm")

Endpicture(1) picture 環境を終える (座標軸をかく)

Endpicture(0) picture 環境を終える (座標軸をかかない)

Closefile({"1"または"0"})書き出し用ファイルを閉じる (デフォルト=画面に戻す)

※ "1"または"0"の文字列を指定

Endpicture(1 または 0) を書き出す

5.2 プロットデータ

Drwline(PD 列または list{, 太さ })

PD 列または list を実線で書き出す

例) Drwline(G1, G2)

例) Drwline(G3, 0.5)

Dashline(PD 列または list{, len {, gap} })

PD 列または list を破線で書き出す (実線部から始まる)

例) Dashline(G1, G2)

例) Dashline(G1, 1.5) (実線部, ギャップとも 1.5 倍)

例) Dashline(G1, G2, 1.5, 0.5)

(実線部 1.5 倍, ギャップ 0.5 倍)

Invdashline(PD 列または list{, len{, gap} })

破線を書き出す (ギャップから始まる)

Dottedline(PD 列または list{, len {, size} })

点線を書き出す

※ 「接線方向」, 「法線方向」 はそれぞれの微小移動量

※ graphicx パッケージが必要

Texletter(点 (list 形式), 方向, 文字列, ...)

点の位置の「方向」に文字列をかく (複数可)

例) Texletter(list(4, "#1"), "n", "文字")

※ 位置は "n", "s", "e", "w", "ne", "nw", "se", "sw", "c"

※ 点の位置はリスト形式で, T_EX の文字列を渡すことができる.

Openphr(ユーザーコマンド名), Closephr()

\def のコマンド定義

例) Openphr("\p")

Texcom("\begin{array}{cc}")

Texcom("5 & 3\\")

Texcom("8 & 7")

Texcom("\end{array}\$")

Closephr()

Openpar(ユーザーコマンド名, 幅 { 縦方向 }), Closepar()

minipage 環境を含む \def コマンド定義

例) Openpar("\s", "5cm", "t")

Texcom("\input{rei}")

Closepar()

Letter(c(2, 3), "se", "\s")

※ 縦方向のデフォルトは c

FontSize("記号")

文字サイズの変更コマンドを書き出す

"n", "s", "f", "ss", "t",

"la", "La", "LA", "h", "H" (" " のとき "n")

例) Fontsize("s")

Texcom("コマンド")

T_EX コマンドのコードを書き出す

例) Texcom("\newcounter{tmpct}")

※ "newline" のとき, 空白行を挿入

Bowname(弓形, 数式 {, 方向 })

弓形 PD の「方向」に式を書き入れる

※ 方向のデフォルトは "c"

例) Gb=Bowdata(A, B, 1, 0.5);

Bowname(Gb, 'd');

Bownamerot(弓形 {, 接線方向 {, 法線方向 }}, 数式 {, 向き })

弓形 PD の中央に式を傾けて書き入れる

※ graphicx パッケージが必要

※ 向きに -1 を指定すると向きが反対になる

Xyzaxparaname(軸データ {, 各軸のラベル名 } {, 離れ })

平行投影で, 各軸のラベルを書き入れる

例) Gax<- Xyzax3data("x<- c(0,1)", "y<- c(0,1)", "z<- c(0,2)")

Xyzaxparaname(Gax)

※ "¥sin x" など文字列で指定することもできる

Xyzaxpersname(軸データ {, 各軸のラベル名 } {, 離れ })

一点投影で, 各軸のラベルを書き入れる

例) Xyzaxpersname(Gax, "", "", "w")

6 プロットデータの操作

6.1 平面

Joingraphics(PD1,PD2, ... { , "L" })	複数の PD を 1 つの PD に合併 G1<- Joingraphics(F9, G10) ※ "L"を指定したときは、結果をリストで返す
Dividegraphics(PD)	PD を要素に分けた list を作成 例) FL<- Dividegraphics(G1) 例) G1<- Op(1, FL)
Joincrvs(PD 列)	複数の曲線をつなげた PD を作成 (2D, 3D 共通) 例) G3<- Joincrvs(G1, Invert(G2)) ※ 曲線は隣接する順番で指定する
Invert(PD)	PD の点列を逆順にした PD を作成 (2D, 3D 共通)
Partcrv(s1, s2, PD)	曲線 PD 上のパラメータ値 s1, s2 を両端とする PD を作成 ※ s1 > s2 の場合 s2 から終点, 始点から s1 までの PD のリストを出力 PD が閉曲線のときは上の 2 つの PD をつなげる.
Partcrv(A, B, PD)	曲線 PD 上の点 A, B の間の部分曲線の PD を作成 ※ A, B の順序が逆転しているとき, B から終点, 始点から A までの PD のリスト (閉のとき接続) を出力 例) G1<- Plotdata("x^2", "x=c(XMIN, XMAX)") G2<- Partcrv(c(0,0), c(1,1), G1) G3<- Partcrv(c(1,1), c(0,0), G1)
Intersectcrvs(PD1, PD2)	2 曲線 PD1, PD2 の交点リストを作成 例) G1<- Paramplot("c(cos(t), sin(t))", "t=c(0, 2*pi)") G2<- Plotdata("x+1/2", "x") PL<- Intersectcrvs(G1,G2)
IntersectcrvsPp(PD1, PD2)	2 曲線 PD1, PD2 の交点とパラメータのリストを作成
Intersectlines(L1, L2)	2 直線の交点を返す 例) L1<- Lineplot(A, B) L2<- Lineplot(C, D) P<- Intersectcrvs(L1,L2)
Nearestpt(P, PD)	点 P に最も近い曲線 PD 上の点とパラメータ値のリストを返す 例) Pp<- Nearestpt(c(0, 1), G1) A<- Op(1,Pp)
Nearestpt(PD1, PD2)	PD1 の節点のうち、PD2 に最も近い点データのリストを返す 例) Pp<- Nearestpt(G1, G2) A<- Op(1,Pp)
Ptstart(PD), Ptend(PD)	曲線 PD の始点 (終点) を返す
Numptcrv(PD)	曲線 PD の節点データの個数を返す
Ptcrv(n, PD)	曲線 PD の n 番目の節点を返す
Pointoncrv(s, PD)	PD 上の点でパラメータ値 s をもつ点を返す 例) Pointoncrv(5.3, G1) (5 番目の線分上で 0.3 の位置にある点)
Paramoncrv(P {, n }, PD)	PD (の n 番目の線分) 上にある点 P のパラメータを返す 例) Paramoncrv(c(3, 2), G1) 例) Paramoncrv(c(2, 4), 5, G1)

6.2 空間

Partcrv3(S1, S2, PD) 曲線 PD 上のパラメータ値 S1,S2 を両端とする PD を作成
Rotate3pt(点, V1, V2{, C})
回転移動した点を返す (Rotate3data 参照)
Rotate3ptc(点, 軸, 角度 {, C})
回転移動した点を返す (Rotate3data 参照)
Parapt(点), Perspt(点)
空間の点を投影した点を返す
Zparapt(点), Zperspt(点)
投影した平面を X, Y としたときの Z 座標
Invparapt(P, PD3d), Invperspt(P, PD3d)
PD3d を投影した PD 上の点 P に対応する PD3d 上の点
※ Pd3d が線分のときは, 延長線上の点でもよい.
Invperspt(s, PD2d, PD3d), Invperspt(s, PD2d, PD3d)
PD2d 上のパラメータ値 s の点に対応する PD3d 上の点
Cancoordpers(投影座標) 一点投影で「投影座標」で表される点の標準座標
Viewfrom(Vec, 曲線 3D {, 非表示オプション})
一時的に Vec 方向からみた射影データを返す
例) Out1<- Viewfrom(c(0,0,1), G1) (表示してデータを作成)
例) Out1<- Viewfrom(c(0,0,1), G1, 0) (データのみを作成)

7 その他

Readtextdata(ファイル名, { 開始位置 {, オプション }})
ファイルからコンマ, スペース, タブ区切りのテキストを読み込み,
データ行列を返す
※ オプション:
"R=読み込み行数" (デフォルトはすべて)
"C=読み込み列数" (デフォルトはすべて)
"Cna=論理値" 1 行を列名にするか (デフォルト TRUE)
"Rna=論理値" 1 列を行名にするか (デフォルト FALSE)
例) DL<- Readtextdata("dt.csv", c(2, 1), "R=1000", "C=2")
Writetextdata(データフレーム, ファイル名)
データフレームを .csv ファイルに書き出す
※ 列名は 1 行目におき, NA は blank にする
例) Writetextdata(Df, "ex.csv")
Tonumeric(文字列からなるデータ行列 {, 開始位置 {, 終了位置 }})
行列の成分を数値に変換 (変換できる行と列からなる部分行列)
例) Dn<- Tonumeric(DL)

7.1 作表

Tabledata({ 大きさ, } 縦線相対幅, 横線相対高さ)
表のデータ list を返す
戻り値: PD, 縦線添字, 横線添字, 枠縦 PD, 枠横 PD, 外枠 PD
大きさは次のベクトル
横, 縦 (, 左 margin, 右 margin (, 上 margin, 下 margin))
※ 横 (縦) を -1 としたときは, 縦 (横) 線のデータから
自動的に計算される (デフォルト)

縦線相対位置は左の罫線からの幅 list (縦方向の始点, 終点)

横線相対位置は上の罫線からの幅 list (横方向の始点, 終点)

※ 描画領域は自動的に設定される

例) Tmp1<- list(20, 30,list(30,0,10), list(0,15,20), 40)

Tmp2<- list(15)

Out<- Tabledata(c(150,20),Tmp1,Tmp)

Tb<- Tabledata(Tmp1,Tmp2)

Dividetable(表データ) 枠, 縦罫線, 横罫線を成分とするリストを返す

例) G<- Dividetable(Tb) (G[[1]],G[[2]],G[[3]] が枠, 縦, 横)

Partframe(表データ, 開始位置, 終了位置)

枠の一部の PD

※ 位置はそれぞれ, c(列番号, 行番号)

※ 開始位置から終了位置までの反時計回りの PD

例) G<- Parframe(Tb, c(4,1),c(1,2))

Findcell(表データ, 列番号, 行番号)

セルの情報 list (中心, 横幅/2, 縦幅/2) を返す

※ 番号は左上の位置

例) Out<- Findcell(Out,2,1)

※ 番号がベクトルのときは, その範囲のセル

例) Out<- Findcell(Out,c(2,4),1)

※ 番号がベクトルのときは, その範囲のセル

例) Out<- Findcell(Out,c(2,4),1)

Findcell(表データ, 左セル, { 右セル })

例) Out<- Findcell(Out,"A2")

Diagcelldata(表データ, 列番号, 行番号)

セルの対角線 PD のリストを返す

Putcell(表データ, 列番号, 行番号, 位置, 文字データ)

セルに文字列を入れるコードを出力

※ 位置は"c", "r", "l", "u", "d", "b"

※ u : up, d : down, b : baseline (微小移動量を付加できる)

例) Putcell(Out,2,1,"c","221")

例) Putcell(Out,"B3","l","\$ab\$")

Putrow(表データ, 行番号, 文字データの列)

1 行に順に文字を書き入れる

例) Putrow(TbL, 2, "a", "b", "c")

※ デフォルト位置は "c" それ以外のときは list 内で指定

※ 複数列にわたるときは, 列数を list 内で指定

例) Putrow(TbL, 2, list("r","a"), list(2, "b"), "c")

(r の位置に a, 2 列とって b をおく)

Putrowexpr(表データ, 行番号, 文字位置, 文字データの列)

1 行に順に数式を書き入れる

PutcoL(表データ, 列番号 (名前) (, 文字データの列)

1 列に順に文字を書き入れる

例) PutcoL(TbL, "C", "a", "b", "c")

PutcoLexpr(表データ, 列番号 (名前), 文字位置, 文字データの列)

1 列に順に数式を書き入れる

Putrowstr(表データ, 行番号, 文字位置, 文字列)

1 行に文字列の文字を 1 つずつ書き入れる

例) Putrowstr(TbL, 1, "c", "xyz")

PutcoL(表データ, 列番号 (名前), 文字位置, 文字列)

7.2 T_EX のコマンド書き出し (メタコマンド)

Texcom(" コマンド") T_EX コマンドのコードを書き出す
 例) Texcom("\begin{minipage}{3cm}")
 ※ "newline" のとき, 空白行を挿入

Openphr(ユーザーコマンド名), Closephr()
 \def のコマンド定義を始める
 例) Openphr("\p")
 Texcom("\begin{array}{cc}")
 Texcom("5 & 3\\")
 Texcom("8 & 7")
 Texcom("\end{array}\$")
 Closephr()

Openpar(ユーザーコマンド名, 幅 {, 位置 }), Closepar()
 minipage 環境を含む \def コマンド定義を始める
 ※ 位置のデフォルトは "c"
 例) Openpar("\s", "5cm")
 Texcom("\input{rei}")
 Closepar()
 Letter(c([2,3], "se", "\s")

Texletter(点 (list 形式), 方向, 文字列, ...)
 点の位置の「方向」に文字列をかく (複数可)
 例) Texletter(list(4, "#1"), "n", "文字")
 ※ 位置は "n", "s", "e", "w", "ne", "nw", "se", "sw", "c"
 ※ 点の位置はリスト形式で, T_EX の形式で渡すことができる.

Texnewctr(番号または番号のベクトル)
 K_ETpic で使うカウンタ (ketpicctr, ...) を定義

Texctr(番号またはカウンタ名)
 番号のカウンタ名またはカウンタ名を返す

Texthctr(番号) \the+カウンタ名の文字列を返す

Texvalctr(番号) \value{ カウンタ名 } の文字列を返す

Texsetctr(番号, 文字列) カウンタに値をセットする T_EX コマンド列を出力
 例) Texsetctr(2, "1*2/3");
 例) Texsetctr(2, "(-#1)+2");

Texletter(位置 (list), 方向, 文字列)
 位置 list で表される点に文字列をかく T_EX コマンド列を出力
 例) Texletter(list(10, paste("-", Texvctr(2), sep="")), "ne", "\content");
 例) Texletter(list(0, "#1"), "c", "A");

Texnewcmd(コマンド名, 引数の個数, オプション値)
 \newcommand を始める T_EX コマンドを出力

Texrenewcmd(コマンド名, 引数の個数, オプション値)
 \renewcommand を始める T_EX コマンドを出力

Texend() T_EX のコマンド定義を終わる T_EX コマンドを出力

Texfor(カウンタ番号, 初期値, 終了値)
 T_EX のループ構造を始める.
 ※ 初期値, 終了値は文字列で与える.
 例) Texfor(1, "1", "#1");

Texendfor(カウンタ番号) \TeX のループ構造を終える.
 例) Texendfor(1);

Texforinit() \TeX のループ構造を初期化

Texif(数値条件 { ,1 }) \TeX の if 構造を始める. (ifnum または ifdim)
 ※ 条件は文字列で与える.
 ※ 1 を追加したときは ifdim
 例) Texif("Texctr(1)<#2");

Texelse() \TeX の else ブロック.

Texendif() \TeX の if 構造を終える.

7.3 カラー設定

Setcolor(色 {, 濃さ }) 色を設定
 ※ color パッケージ必要

色は, 次の文字列または [c,m,y,k] のベクトル

"greenyellow" [0.15,0,0.69,0], "yellow" [0,0,1,0], "goldenrod" [0,0.1,0.84,0], "dandelion" [0,0.29,0.84,0]
 "apricot" [0,0.32,0.52,0], "peach" [0,0.5,0.7,0], "melon" [0,0.46,0.5,0], "yelloworange" [0,0.42,1,0]
 "orange" [0,0.61,0.87,0], "burntorange" [0,0.51,1,0], "bittersweet" [0,0.75,1,0.24],
 "redorange" [0,0.77,0.87,0]
 "mahogany" [0,0.85,0.87,0.35], "maroon" [0,0.87,0.68,0.32], "brickred" [0,0.89,0.94,0.28], "red" [0,1,1,0]
 "orangered" [0,1,0.5,0], "rubinered" [0,1,0.13,0], "wildstrawberry" [0,0.96,0.39,0],
 "salmon" [0,0.53,0.38,0]
 "carnationpink" [0,0.63,0,0], "magenta" [0,1,0,0], "violetred" [0,0.81,0,0], "rhodamine" [0,0.82,0,0]
 "mulberry" [0.34,0.9,0,0.02], "redviolet" [0.07,0.9,0,0.34], "fuchsia" [0.47,0.91,0,0.08],
 "lavender" [0,0.48,0,0]
 "thistle" [0.12,0.59,0,0], "orchid" [0.32,0.64,0,0], "darkorchid" [0.4,0.8,0.2,0], "purple" [0.45,0.86,0,0]
 "plum" [0.5,1,0,0], "violet" [0.79,0.88,0,0], "royalpurple" [0.75,0.9,0,0], "blueviolet" [0.86,0.91,0,0.04]
 "periwinkle" [0.57,0.55,0,0], "cadetblue" [0.62,0.57,0.23,0], "cornflowerblue" [0.65,0.13,0,0],
 "midnightblue" [0.98,0.13,0,0.43]
 "navyblue" [0.94,0.54,0,0], "royalblue" [1,0.5,0,0], "blue" [1,1,0,0], "cerulean" [0.94,0.11,0,0]
 "cyan" [1,0,0,0], "processblue" [0.96,0,0,0], "skyblue" [0.62,0,0.12,0], "turquoise" [0.85,0,0.2,0]
 "tealblue" [0.86,0,0.34,0.02], "aquamarine" [0.82,0,0.3,0], "bluegreen" [0.85,0,0.33,0],
 "emerald" [1,0,0.5,0]
 "junglegreen" [0.99,0,0.52,0], "seagreen" [0.69,0,0.5,0], "green" [1,0,1,0], "forestgreen" [0.91,0,0.88,0.12]
 "pinegreen" [0.92,0,0.59,0.25], "limegreen" [0.5,0,1,0], "yellowgreen" [0.44,0,0.74,0],
 "springgreen" [0.26,0,0.76,0]
 "olivegreen" [0.64,0,0.95,0.4], "rawsienna" [0,0.72,1,0.45], "sepia" [0,0.83,1,0.7], "brown" [0,0.81,1,0.6]
 "tan" [0.14,0.42,0.56,0], "gray" [0,0,0,0.5], "black" [0,0,0,1], "white" [0,0,0,0]