

ツール改善（2025年9月以降）によって操作などが変わった部分を、書籍の図でソフト操作を説明したものについて新旧ソフトを対比した。

195ページ図 1 0. 1

213ページ図 1 0. 1 0

214ページ図 1 0. 1 1

215ページ図 1 0. 1 2

240ページ図 1 1. 3

241ページ図 1 1. 5

244ページ図 1 1. 8

253ページ図 1 1. 1 6

255ページ図 1 1. 1 8

258ページ図 1 1. 2 0

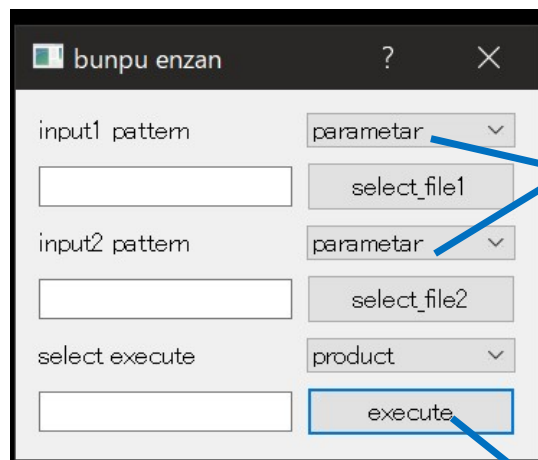
266ページ図 1 1. 2 7

268ページ図 1 1. 2 9

195ページ

図 10. 1

<旧ソフト>

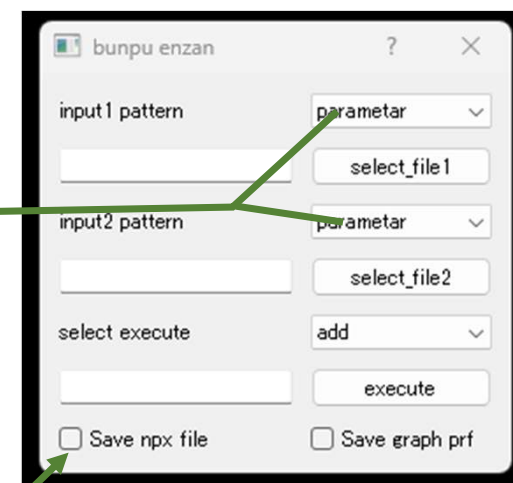


parameter :分布諸元指定
distribution_file : npz 形式指定
data_file :データから分布生成
vector(2のみ) :ベクトル指定
weibull (2のみ) :ワイブル分布
data file2 :データからGUI調整
hist :頻度値データ指定
parameter2 :諸元をGUI調整
existing dist :既存の分布名指定

生成された分布は、npz出力しなくても分布名指定で再利用できるので、出力可否チェックを追加

add :加算
sub :減算
product :積算
division :商算
percent :パーセント
percent2 :分布の比較
simulation_x :距離積分
simulation_t :時系列演算

<新ソフト>



percent4 :パーセントGUI表示
compare :分布の比較GUI表示
contact :分布の接近
balance :分布のバランス (交点)
balance2 :分布のバランス (指定点)
graph2 :複数分布GUI表示
multiple :分布の所定倍

GUIと書いたメニューは表示ツールで、
グラフ保存や視点変換などが可能

メニューの追加
青枠が旧ソフト
緑枠が新ソフト

(GUI調整、GUI表示とは、
select_fileをクリックすると
グラフ表示画面が立上る)

213ページ図10.10

演算対象のデータ間に
相関関係がある場合

<旧ソフト>

bunpu enzan

input1 pattern: parameter
[0],[175],[158],[6],[100] select_file1

input2 pattern: parameter
[5],[128],[122],[2],[100] select_file2

select execute: add
add_pos,0.5 execute

出力ファイル名、
相関係数相当

<新ソフト>

bunpu enzan

input1 pattern: parameter
[3],[5],[4.2],[0.3],[100] select_file1

input2 pattern: parameter
[6],[11],[7.8],[0.8],[100] select_file2

select execute: add
add_pos,corel=[0.5] execute

☒ Save npz file ☒ Save graph prf

新旧で入力
項目は同じ、
(一部項目
はラベルで
指定)

出力ファイル名指定して
も、しなくてもよい。

**相関係数はcorel=[],分割
数はdivt0=[]として指定**

出力ファイル名は指定すればそ
のファイル名、しなければ自動
割当、チェックでファイルが生
成、npzファイル無くても変数
名として使いまわせる

214ページ図10.11
データ間の相関がある
場合の和算

<旧ソフト>

bunpu enzan

input1 pattern parameter
[0],[175],[158],[6],[100] select_file1

input2 pattern parameter
[5],[128],[122],[2],[100] select_file2

select execute add
add execute

bunpu enzan

input1 pattern parameter
[0],[175],[158],[6],[100] select_file1

input2 pattern parameter
[5],[128],[122],[2],[100] select_file2

select execute add
add_pos,0.5 execute

bunpu enzan

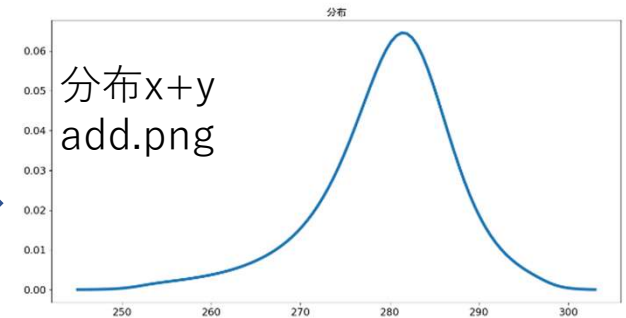
input1 pattern parameter
[0],[175],[158],[6],[100] select_file1

input2 pattern parameter
[5],[128],[122],[2],[100] select_file2

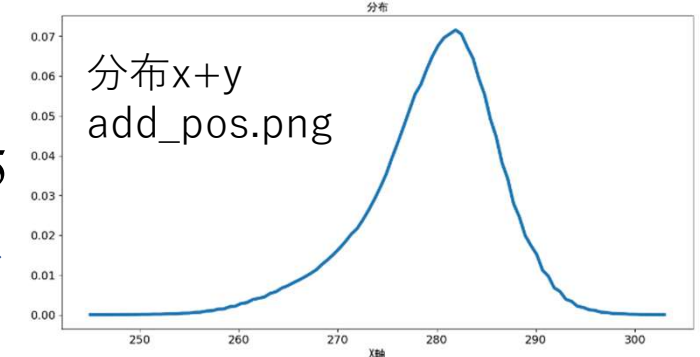
select execute add
add_neg,-0.5 execute

和算 X+Yの分布

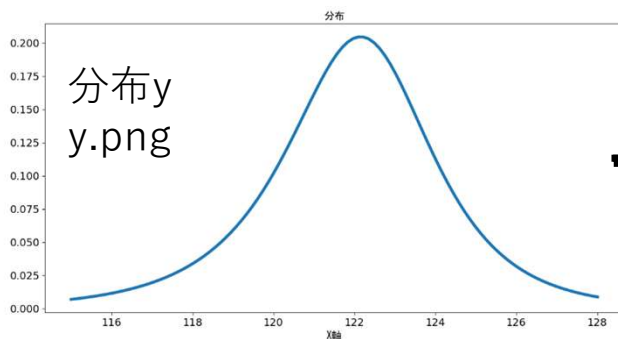
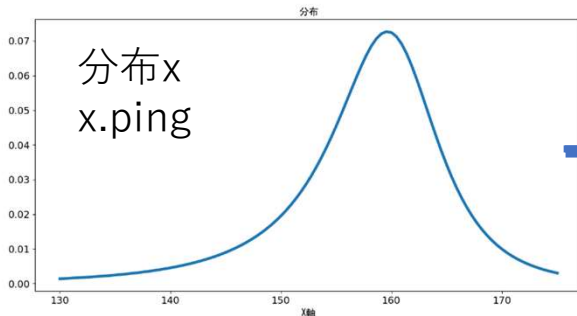
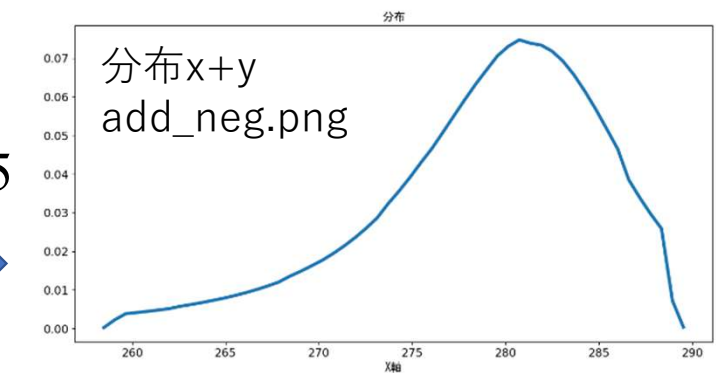
独立
 $r = 0$



正相関
 $r = +0.5$



逆相関
 $r = -0.5$

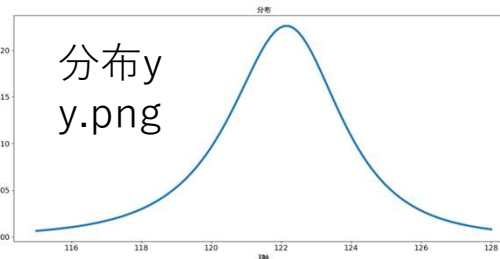
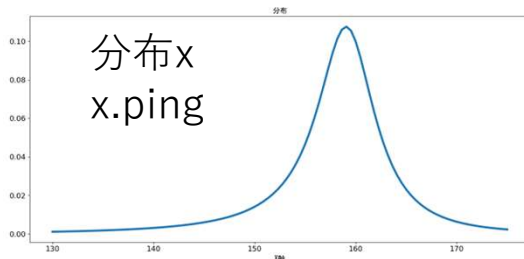


214ページ

図 1 0 . 1 1

データ間の相関がある場合の和算

<新ソフト>



変化点

相関係数の指定はcorel=[]
とコマンドと配列で指定
相関係数が1、-1の場合
の演算を行う機能を追加
チェックでファイル
出力の有無を指定

bunpu enzan

input1 pattern parameter
[130],[175],[158],[6],[100] select_file1

input2 pattern parameter
[115],[128],[122],[2],[100] select_file2

select execute add

add_0 execute

☐ Save npx file ☒ Save graph prf

bunpu enzan

input1 pattern parameter
[130],[175],[158],[6],[100] select_file1

input2 pattern parameter
[115],[128],[122],[2],[100] select_file2

select execute add

add_05,corel=[0.5] execute

☐ Save npx file ☒ Save graph prf

bunpu enzan

input1 pattern parameter
[130],[175],[158],[6],[100] select_file1

input2 pattern parameter
[115],[128],[122],[2],[100] select_file2

select execute add

add_n05,corel=[-0.5] execute

☐ Save npx file ☒ Save graph prf

bunpu enzan

input1 pattern parameter
[130],[175],[158],[6],[100] select_file1

input2 pattern parameter
[115],[128],[122],[2],[100] select_file2

select execute add

add_1,corel=[1] execute

☐ Save npx file ☒ Save graph prf

bunpu enzan

input1 pattern parameter
[130],[175],[158],[6],[100] select_file1

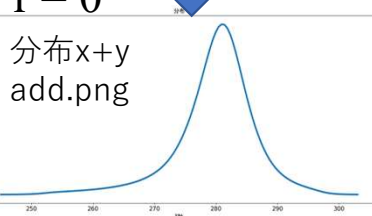
input2 pattern parameter
[115],[128],[122],[2],[100] select_file2

select execute add

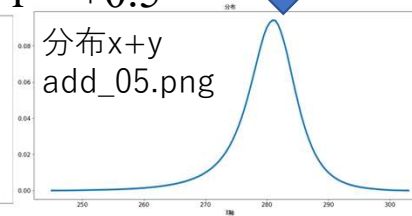
add_n1,corel=[-1] execute

☐ Save npx file ☒ Save graph prf

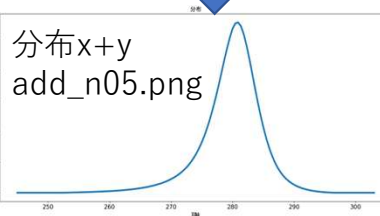
独立
 $r = 0$



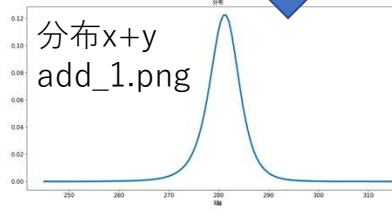
正相関
 $r = +0.5$



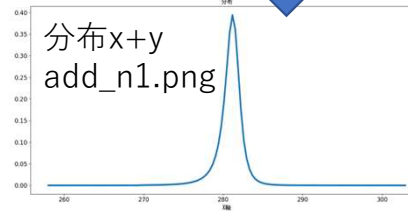
逆相関
 $r = -0.5$



正相関
 $r = +1$



逆相関
 $r = -1$



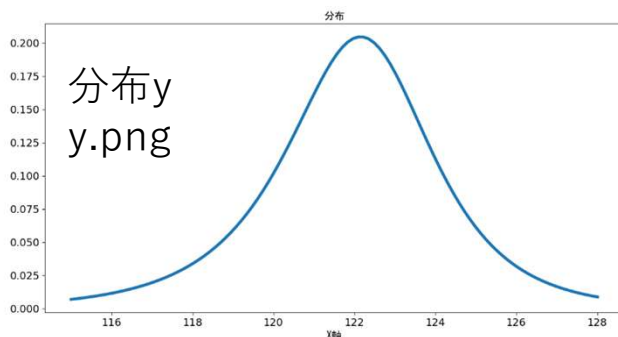
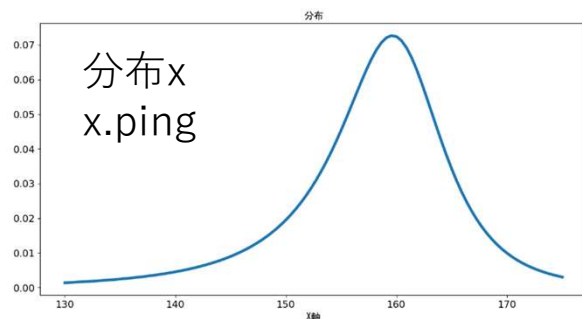
和算
X+Y
の分布

215ページ

<旧ソフト>

図10.12

データ間の相関がある
場合の積算



bunpu enzan

input1 pattern: parameter
[0].[175].[158].[6].[100] select_file1

input2 pattern: parameter
[5].[128].[122].[2].[100] select_file2

select execute: product
pro execute

bunpu enzan

input1 pattern: parameter
[0].[175].[158].[6].[100] select_file1

input2 pattern: parameter
[5].[128].[122].[2].[100] select_file2

select execute: product
pro_pos,0.5 execute

bunpu enzan

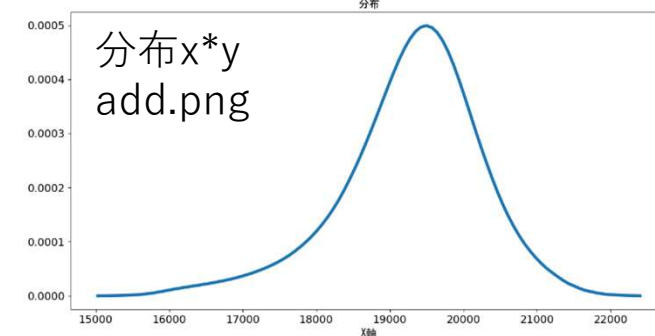
input1 pattern: parameter
[0].[175].[158].[6].[100] select_file1

input2 pattern: parameter
[5].[128].[122].[2].[100] select_file2

select execute: product
pro_neg,-0.5 execute

和算 $X * Y$ の分布

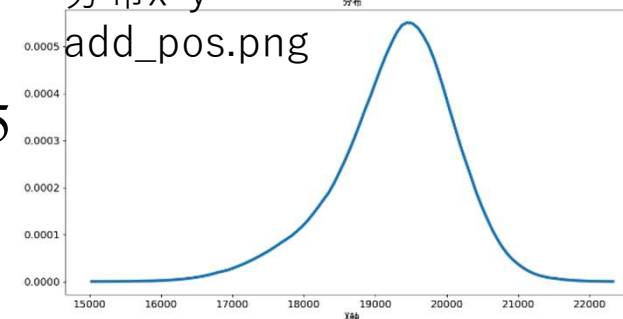
独立
 $r = 0$



正相関
 $r = +0.5$



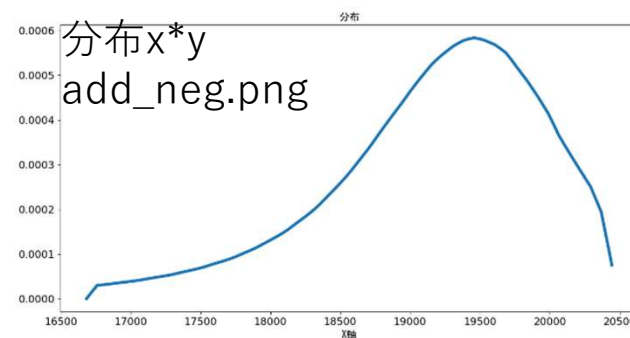
分布x*y
add_pos.png



逆相関
 $r = -0.5$



分布x*y
add_neg.png

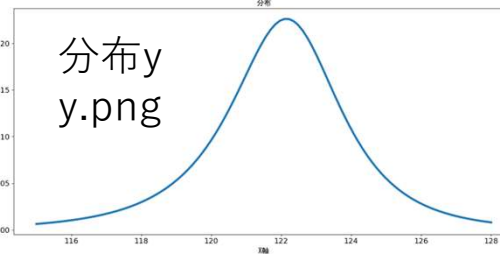
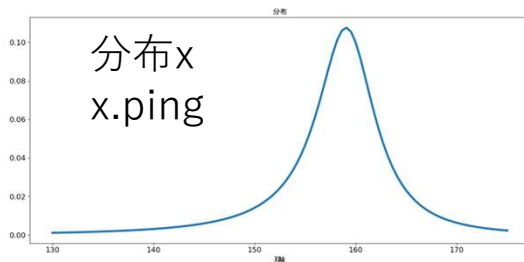


215ページ

図 10. 12

データ間の相関がある場合の積算

<新ソフト>



変化点

相関係数の指定はcorel=[]
とコマンドと配列で指定
相関係数が1、-1の場合
の演算を行う機能を追加
チェックでファイル
出力の有無を指定

bunpu enzan

input1 pattern: [130],[175],[158],[6],[100] select_file1

input2 pattern: [115],[128],[122],[2],[100] select_file2

select execute: product

prd_0

execute

☐ Save npx file ☒ Save graph prf

独立
 $r = 0$

bunpu enzan

input1 pattern: [130],[175],[158],[6],[100] select_file1

input2 pattern: [115],[128],[122],[2],[100] select_file2

select execute: product

prd_05,corel=[0.5]

execute

☐ Save npx file ☒ Save graph prf

正相関
 $r = +0.5$

bunpu enzan

input1 pattern: [130],[175],[158],[6],[100] select_file1

input2 pattern: [115],[128],[122],[2],[100] select_file2

select execute: product

prd_n05,corel=[-0.5]

execute

☐ Save npx file ☒ Save graph prf

逆相関
 $r = -0.5$

bunpu enzan

input1 pattern: [130],[175],[158],[6],[100] select_file1

input2 pattern: [115],[128],[122],[2],[100] select_file2

select execute: product

prd_1,corel=[1]

execute

☐ Save npx file ☒ Save graph prf

正相関
 $r = +1$

bunpu enzan

input1 pattern: [130],[175],[158],[6],[100] select_file1

input2 pattern: [115],[128],[122],[2],[100] select_file2

select execute: product

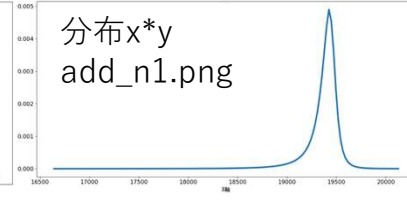
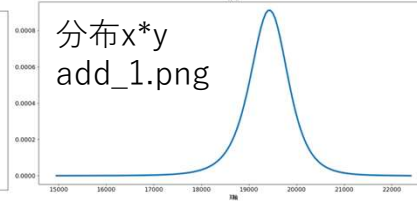
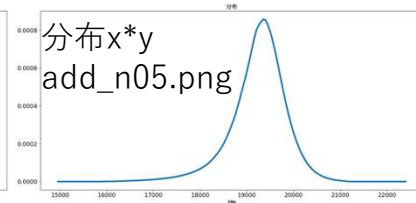
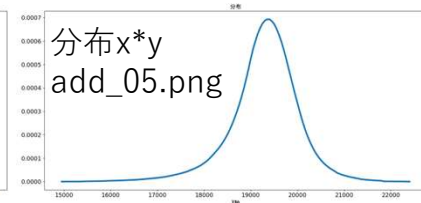
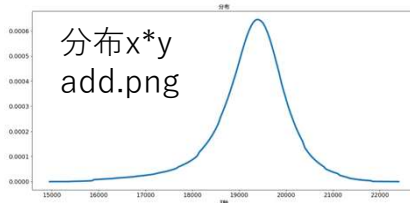
prd_n1,corel=[-1]

execute

☐ Save npx file ☒ Save graph prf

逆相関
 $r = -1$

積算
 $X*Y$
の分布



①生涯作動回数の演算

生涯作動回数 = 作動頻度 × 生涯走行距離（分布の掛算）

<旧ソフト>

Input1
トリップ当たりブレーキ頻度
距離あたりの作動回数
/hindo.csv,1,[1],[100],0,0

Input2
生涯走行距離
/lifetime.csv,1,[1],[100],0,0

出力ファイル名

演算メニューは product（積算）を選択

bunpu enzan

input1 pattern data_file
k/hindo.csv,1,[1],[100],0,0 select_file1

input2 pattern data_file
lifetime.csv,1,[1],[100],0,0 select_file2

select execute product
lifecnt execute

ファイル名、無視する行、データを抽出する列、分布の分割数、データフィルタを設定、カーネル分布のバンド幅

<新ソフト>

新旧で入力項目は同じ

bunpu enzan

input1 pattern data_file
k/hindo.csv,1,[1],[100],0,0 select_file1

input2 pattern data_file
lifetime.csv,1,[1],[100],0,0 select_file2

select execute product
lifecnt execute

☒ Save max file ☒ Save graph prf

lifecnt=x*y

割当られた分布名
後の演算で利用可能。

演算は出力分布名を入力しても、しなくてもよい
入力しなければ自動的に割当、npzファイルを作成しなくても、分布名で後の演算で利用可能。

241ページ図1 1. 5

②故障率の演算

生涯作動回数分布がワイブル分布
を上回る確率の演算

<旧ソフト>

ワイブル分布の指定

Input1
先ほど生成した
生涯作動回数分布
lifecnt.npz

Input2
ワイブル分布
[200,50000,6]

input1 pattern: distribution_file
unpu_pyinst/lifecnt.npz
input2 pattern: weibull
[200,50000,6]
select execute: percent2
[time,time],[1]
execute

新旧で入
力項目は
同じ

<新ソフト>

input1 pattern: distribution_file
_202509/book/lifecnt.npz
input2 pattern: weibull
[00,50000,6]
select execute: percent2
[time,time],[1]
execute

```
python/bunpu_202509/book/hindo.csv,1,[1  
python/bunpu_202509/book/lifetime.csv,1  
56  
...npz 分布ファイルを選択  
open1  
C:/Users/syn/python/bunpu_202509/book/lifecnt.npz  
distribution_file  
weibull  
percent2  
x1.bunpu_load C:/Users/syn/python/bunpu_202509/book/lifecnt.npz  
[200, 50000, 6]  
z1=x.percent2(y)
```

前回演算結果を使う場合、従来通りnpzを使っても
よいし、引当てられた分布名を使ってもよい。
(既存の分布名はメニューのexisting_distを使う)

[出力ファイル名、単位名]、[累積方向]

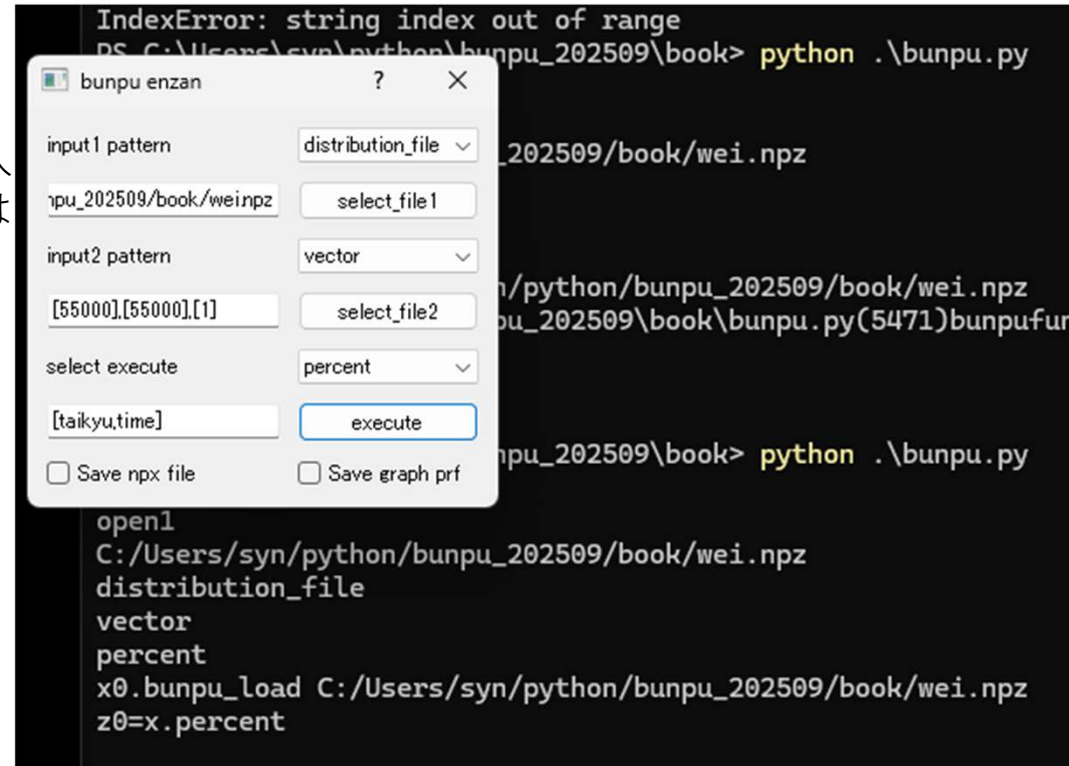
244ページ図 1 1. 8

③評価条件の演算
パーセンタイルを求める

<旧ソフト>
vectorを選択

新旧で入
力項目は
同じ

<新ソフト>



Input1
先ほど生成した
ワイブル分布
wei.npz

Input2
パーセンタイル
[55000],[55000],[1]

[出力ファイル名,単位名] percentを選択

[境界パラメータ 1],[境界パラメータ 2],[累積方向]

境界パラメータ 1 と 2 が異なる場合

⇒ 累積方向 + はその間、- はその外の確率

境界パラメータ 1 と 2 が同じ場合

⇒ 累積方向 + はそれ以上、- はそれ以下の確率

253ページ図1 1. 1 6

①記録日数（日/回）の演算

メモリ 1 回記録される日数（日/回）
＝メモリ 1 回記録される距離（km/回）
／日当たり走行距離（km/日）

<旧ソフト>

新旧で入力
項目は同じ

Input1

減速データの記録頻度、トリップ毎の平均記録距離（km/回）
/kmbtime.csv,1,[1],[100],0,0

Input2

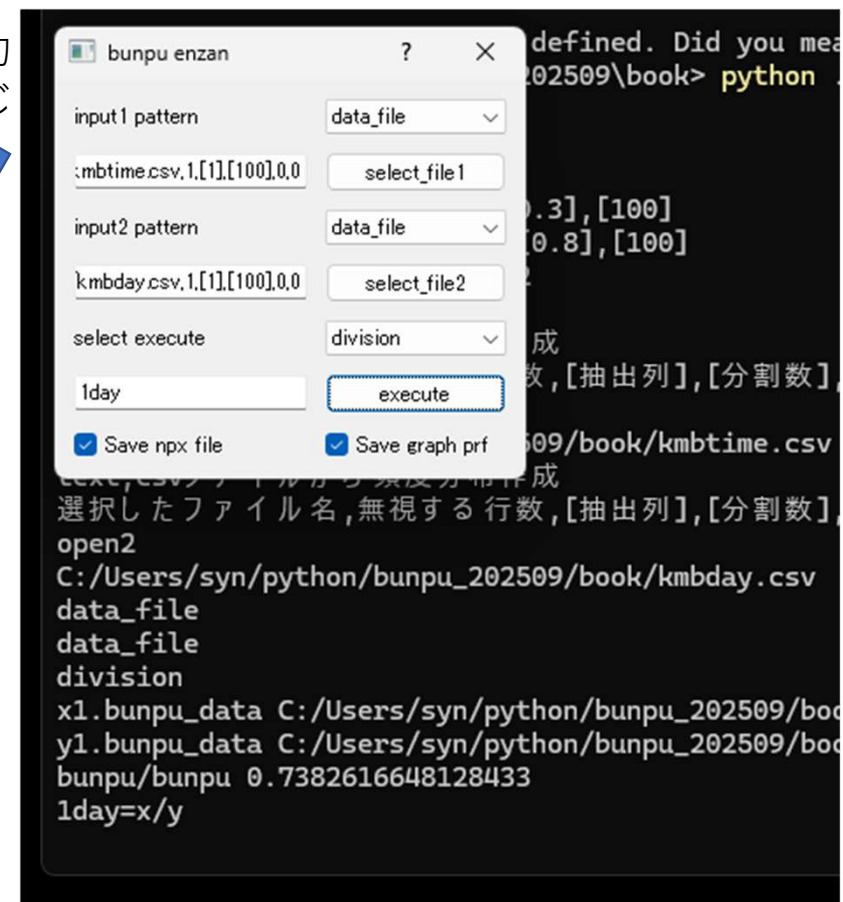
日当たり走行距離（km/日）
/kmbday.csv,1,[1],[100],0,0

出力ファイル名

ファイル名、無視する行、データを抽出する列、
分布の分割数、データフィルタを設定、カーネル
分布のバンド幅

演算メニューは
division（割算）を選択

<新ソフト>



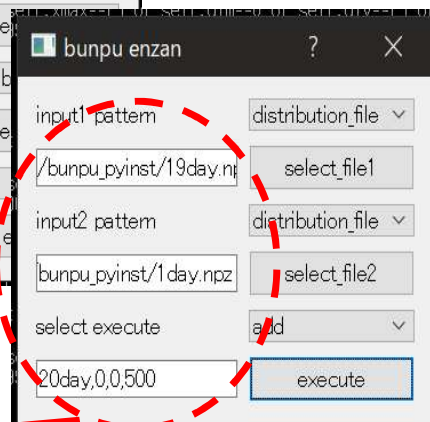
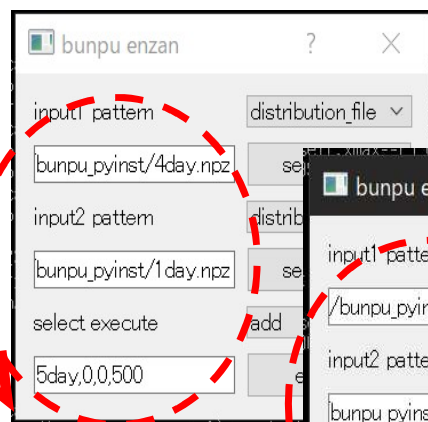
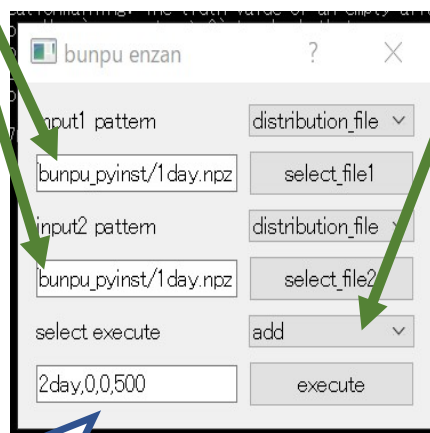
255ページ図1 1. 1 8

②メモリー上書き迄の日数演算

<旧ソフト>

メモリ 1 回記録される
日数 (日/回) /1day.npz

演算メニューは
add (加算) を選択



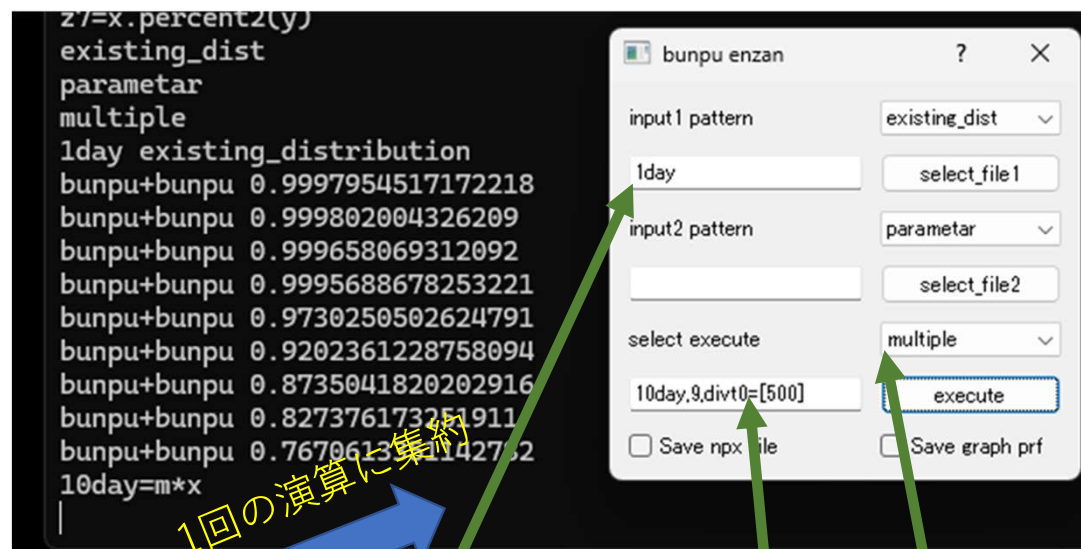
出力ファイル名,0,0,
分布の分割数

メモリー 4 個埋まる日
数と 1 個埋まる日数
を加算して 5 個埋まる日
数分布生成

メモリー 20 個の日数分布

<新ソフト>

加算を10回繰返していたものを倍数計算で1回に集約



前回演算時生成し
た分布名で指定

演算時の分布分割数は、
ラベルを付けて指定
divt0=[分割数]

演算メニューmultipleで任意回数
add (加算) を繰返す。
小数点倍の場合は分布積分で演算

258ページ図 1 1. 2 0

③データ吸上げ前に上書きされる確率

<旧ソフト>

新旧で入力
項目は同じ

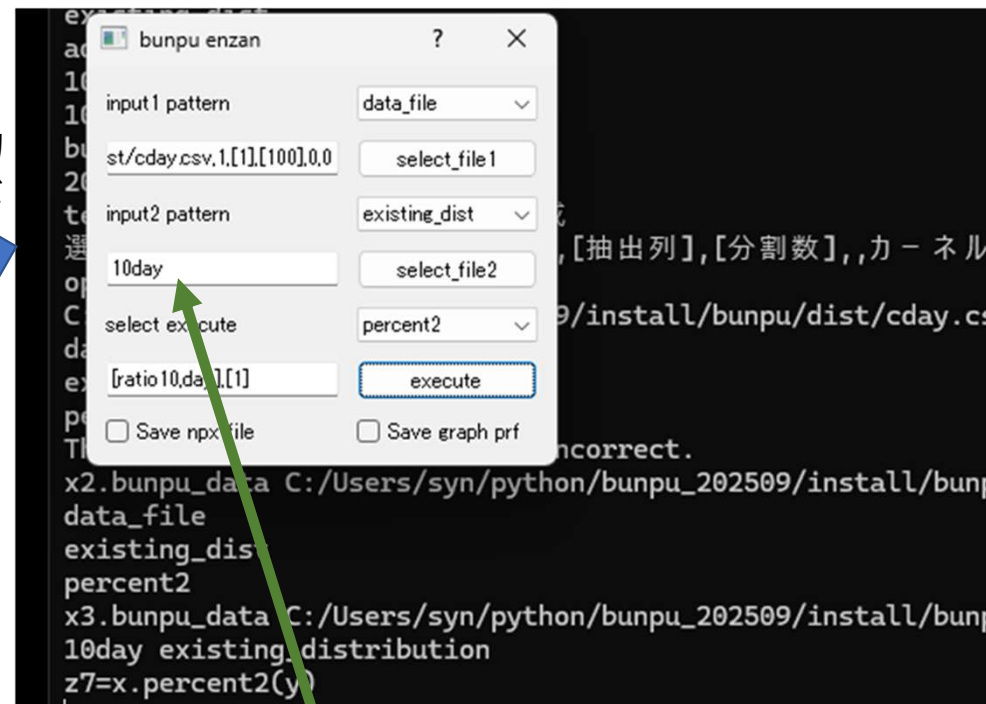
Input1
データ吸上げまでの日数（日）
/cdlay.csv,1,[1],[100],0,0

Input2
メモリー20個埋まる日数分布
/20day.npz

[出力ファイル名、単位名]、[累積方向]

演算メニューはpercent2を選択

<新ソフト>



前回演算時生成した分布名で指定
前回npzファイルを生成すれば、
それを指定してもよい。

266ページ図 1 1. 2 7

①後続車 $n \cdot dt$ 秒後の存在領域演算

<新ソフト>

シミュレーションは変更なし、但しリポジトリ
bunpu3にて新しいシミュレーションを含む
pythonライブラリを公開

<旧ソフト>

simulation_tを選択

Input1

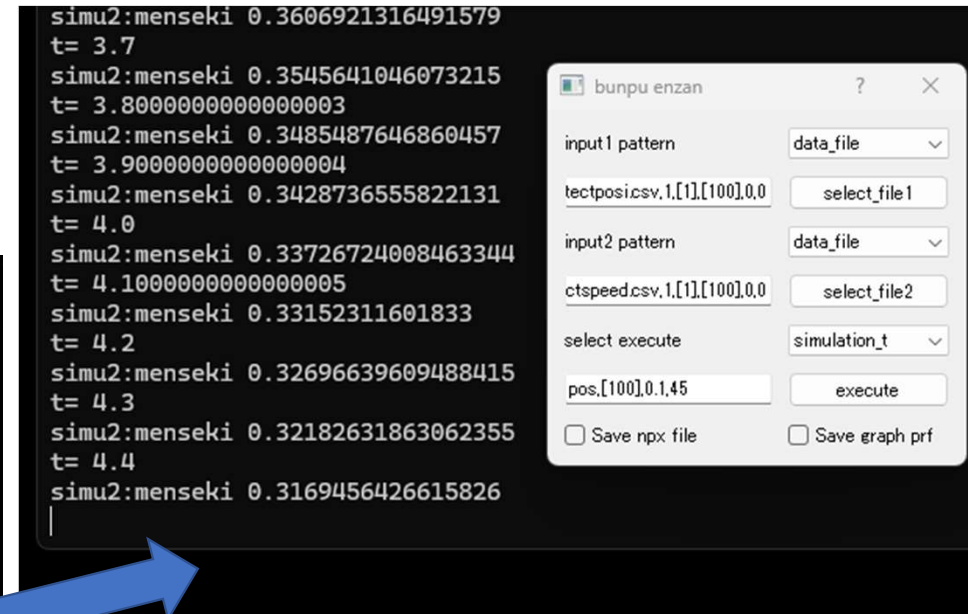
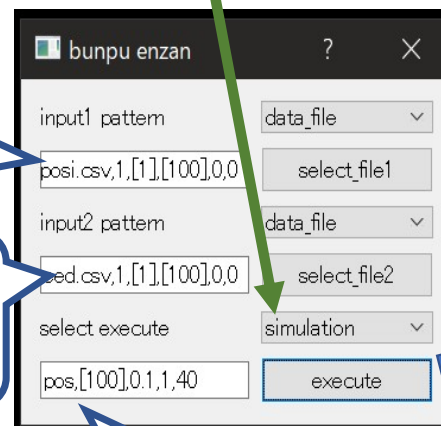
センサーが接近車両を最初に
検出する位置 (m)

/detectposi.csv,1,[1],[100],0,0

Input2

接近車両の速度 (m/s)

/detectspeed.csv,1,[1],[100],0,0



新旧で入力
項目は同じ

出力ファイル名,[分割数],
dtの値,繰返し回数
例 pos,[100],0.1,45

ファイル名、無視する行、データを抽出
する列、分布の分割数、データフィルタ
を設定、カーネル分布のバンド幅

268ページ図 1 1. 2 9

②後続車接近確率演算

<旧ソフト>

Input1
後続車45秒後の存在領域 (m)
/pos44.npz

Input2
車線変更制御後の自車位置分布
/lanechange.csv,1,[1],[100],0,0

input1 pattern: distribution_file
input2 pattern: data_file
select execute: percent2
execute

演算メニューはpercent2を選択

[出力ファイル名、単位名]、[累積方向]

<新ソフト>

```
simu2:menseki 0.3428736555822131
t= 4.0
simu2:menseki 0.33726724008463344
t= 4.1000000000000005
simu2:menseki 0.33152311601833
t= 4.2
simu2:menseki 0.32696639609488415
t= 4.3
simu2:menseki 0.32182631863062355
t= 4.4
simu2:menseki 0.3169456426615826
--.npz 分布ファイルを選択
open1
C:/Users/syn/python/bunpu_202509/install/bunpu/dist/p
distribution_file
data_file
percent2
x5.bunpu_load C:/Users/syn/python/bunpu_202509/install/bunpu/dist/p
y3.bunpu_data lanechange.csv,1,[1],[100],0,0
z9=x.percent2(y)
```

input1 pattern: distribution_file
input2 pattern: data_file
select execute: percent2
execute

新旧で入力
項目は同じ