Part 2: 記述統計の可視化

1. Data

本パートでは，解析(Ｒ)の Part 2: 記述統計の可視化と同様に causaldata package に格納されている National Health and Nutrition Examination Survey Data I Epidemiologic Follow-up Study (NHEFS) の nhefs データセットを用います．同様のデータセットが SAS データセット形式で web 公開されているので，それを利用します．

What If の NHEFS data
https://cdn1.sph.harvard.edu/wp-content/uploads/sites/1268/2012/10/nhefs_sas.zip

ダウンロードし，解凍した nhefs.sas7bdat を，参照可能な適当な場所に保存し，

```
libname lib1 “保存したパス”;
```

これにより，ライブラリ lib1 の nhefs データセットとして認識可能です．
libname ステートメントで明示的に命名されたロケーションを SAS では永久ライブラリと呼び，なにも指定せずにデータ加工する一時的な領域を Work ライブラリと呼びます．
毎回ライブラリ名を指定するのも面倒であるのと，元データのあるライブラリとデータハンドリングを実施するライブラリとは分けるのが通常望ましいので，一旦 Work に移します．

```
data NHEFS;
 set lib1.NHEFS;
run;
```

これで永久ライブラリとして指定された lib1 の nhefs データセットと同様のデータッセットが Work ライブラリにも作成されました．

まずは，nhefs データセットは何人のデータで，いくつの変数があるか，contents プロシジャで確認します．varnum オプションは変数情報をデータセット内での格納位置順に表示します．

```
proc contents data=NHEFS varnum;
run;
```

CONTENTS プロシジャ

データセット名	WORK.NHEFS	オブザベーション数	1629
メンバータイプ	DATA	変数の数	64
エンジン	V9	インデックス数	0
作成日時	2023/01/06 13:04:05	オブザベーションのバッファ長	512
更新日時	2023/01/06 13:04:05	削除済みオブザベーション数	0
保護		圧縮済み	NO
データセットタイプ		ソート済み	NO
ラベル			
データ表現	WINDOWS_64		
エンコード	shift-jis Japanese (SJIS)		

=====(出力一部省略)＝＝＝

作成順の変数

#	変数	タイプ	長さ	ラベル
1	seqn	数値	8	unique personal identifier
2	qsmk	数値	8	quit smoking between baseline and 1982
3	death	数値	8	death between 1983 and 1992
4	yrdth	数値	8	year of death
5	modth	数値	8	month of death
6	dadth	数値	8	day of death
7	sbp	数値	8	systolic blood pressure in 1982
8	dbp	数値	8	diastolic blood pressure in 1982
9	sex	数値	8	sex
10	age	数値	8	age
11	race	数値	8	race

SAS では行のことをオブザベーションと表現するため，オブザベーション数から，1629 行(人数)で，64 列(変数)であることが確認できます.

先頭の 5 行を print プロシジャで確認してみましょう

```
proc print data = NHEFS (obs = 5);
run;
```

非常に変数が多いため，画像を途中で切ってきますが，スクロールバーで最後の変数までみることができます.

OBS	seqn	qsmk	death	yrdth	modth	dadth	sbp	dbp	sex	age	race	income	marital	school	education	ht	wt71	wt82	wt82_71	birthpla
1	233	0	0	.	.	.	175	96	0	42	1	19	2	7	1	174.188	79.04	68.9460	-10.0940	
2	235	0	0	.	.	.	123	80	0	36	0	18	2	9	2	159.375	58.63	61.2350	2.6050	
3	244	0	0	.	.	.	115	75	1	56	1	15	3	11	2	168.500	56.81	66.2245	9.4145	
4	245	0	1	85	2	14	148	78	0	68	1	15	3	5	1	170.188	59.42	64.4101	4.9901	
5	252	0	0	.	.	.	118	77	0	40	0	18	2	11	2	181.875	87.09	92.0793	4.9893	

Part 3 でも使用する累積的な喫煙の指標 Pack-Year の計算を行って，新しい変数（pack_year_n）をデータステップで作成したいと思います．SAS の場合，データセット名を set ステートメントで指定されたものと変えずにデータステップを行えば，上書き更新になります．ただし，実行 PC(サーバー)のデータ容量に余裕があるなら，トレースのし易さを考慮し，データセットを分けてデータセットステップを刻むことが推奨されます．

```
data NHEFS1;
 set NHEFS;
 pack_years_n = (smokeintensity / 20) * smokeyrs;
run;
```

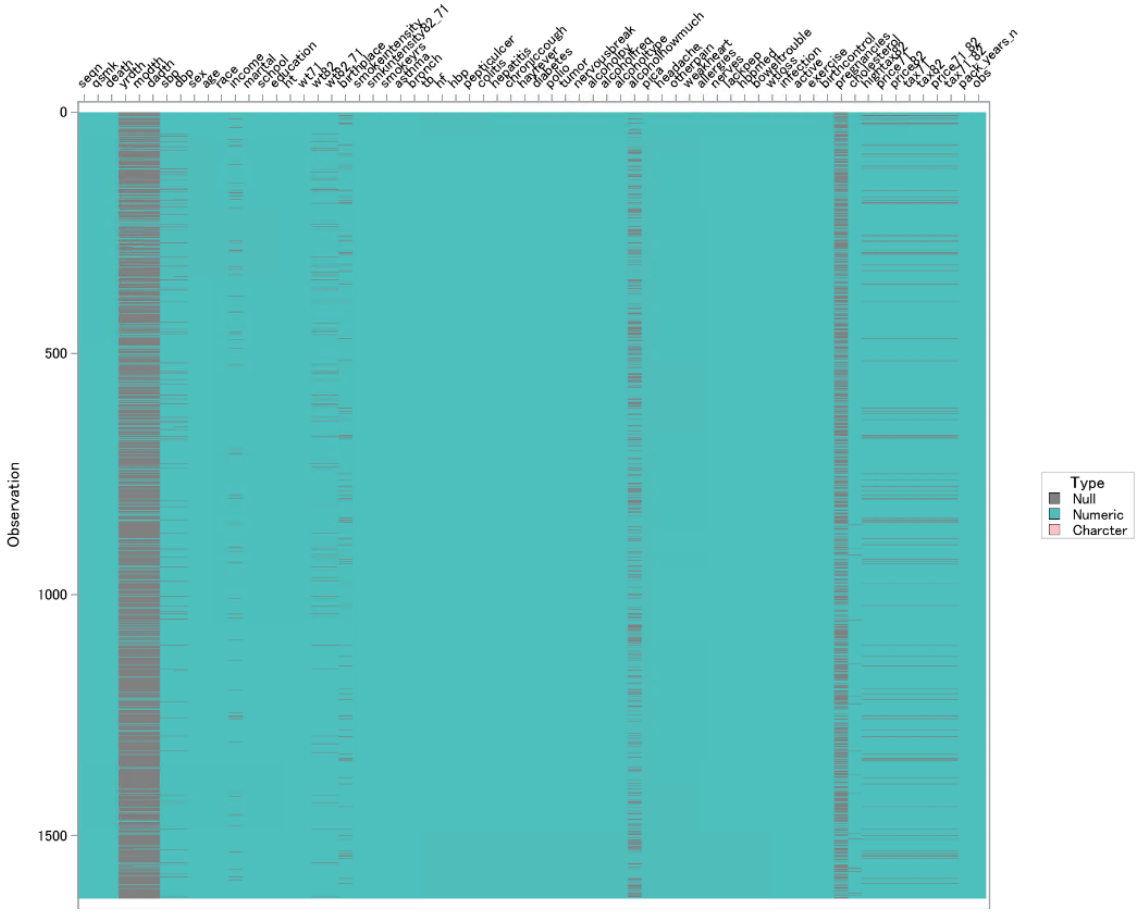
2. Overview & Missing values（欠測値）

まずは，データの概観を可視化してみます．どんなタイプの変数が多いのか,Null はどの程度あるのか,確認します．

SAS の Sgplot プロシジャのプロットステートメントの中には R の visdat パッケージのようにそのままデータを流して，データ構造を可視化できるプロットは（恐らく）ありません．sgplot のヒートマッププロット（heatmapparm）に合わせるために，データステップで描画用に欠損かどうかのカテゴリをいれたデータセットを作成する必要があります

```
proc format;
value catnf
        0     = "Null"
        1     = "Numeric"
        2     = "Charcter";
run;
data nullmap;
length varname $20.;
set NHEFS1;
obs=_N_;
array ar_num _numeric_;
do over ar_num;
 varname=vname(ar_num);
 varn=_i_;
 if missing(ar_num) then catn=0;
 else catn=1;
 output;
end;
array ar_character _char_;
do over ar_character;
 varname=vname(ar_character);
 varn=_i_;
 if missing(ar_character) then catn=0;
 else catn=2;
 if varname ne "varname" then output;
end;
format catn catnf.;
keep varname obs catn;
run;
data mapping;
length ID Show FillColor Value $20.;
```

```
ID='catn';
Show='AttrMap';
do Value ="Null","Numeric","Charcter";
 select(Value);
  when("Null") FillColor="gray";
  when("Numeric") FillColor="bibg";
  when("Charcter") FillColor="pink";
 end;
 output;
end;
run;
ods html path="グラフ出力先のパス  " file="nullmap.html" image_dpi=150;
ods graphics / nxybinsmax=9999999   antialiasmax =9999999 width=1200 height=1000;
proc sgplot data=nullmap dattrmap=mapping;
    format catn catnf.;
    heatmapparm x=varname   y=obs colorgroup=catn /   attrid=catn x2axis;
    x2axis;
    yaxis reverse label="Observation";
    keylegend / title="Type" location=outside position=right across=1;
run;
```



各変数でどの程度（割合）欠測値があるかをロリーポッププロット（点と線）で表現してみます.

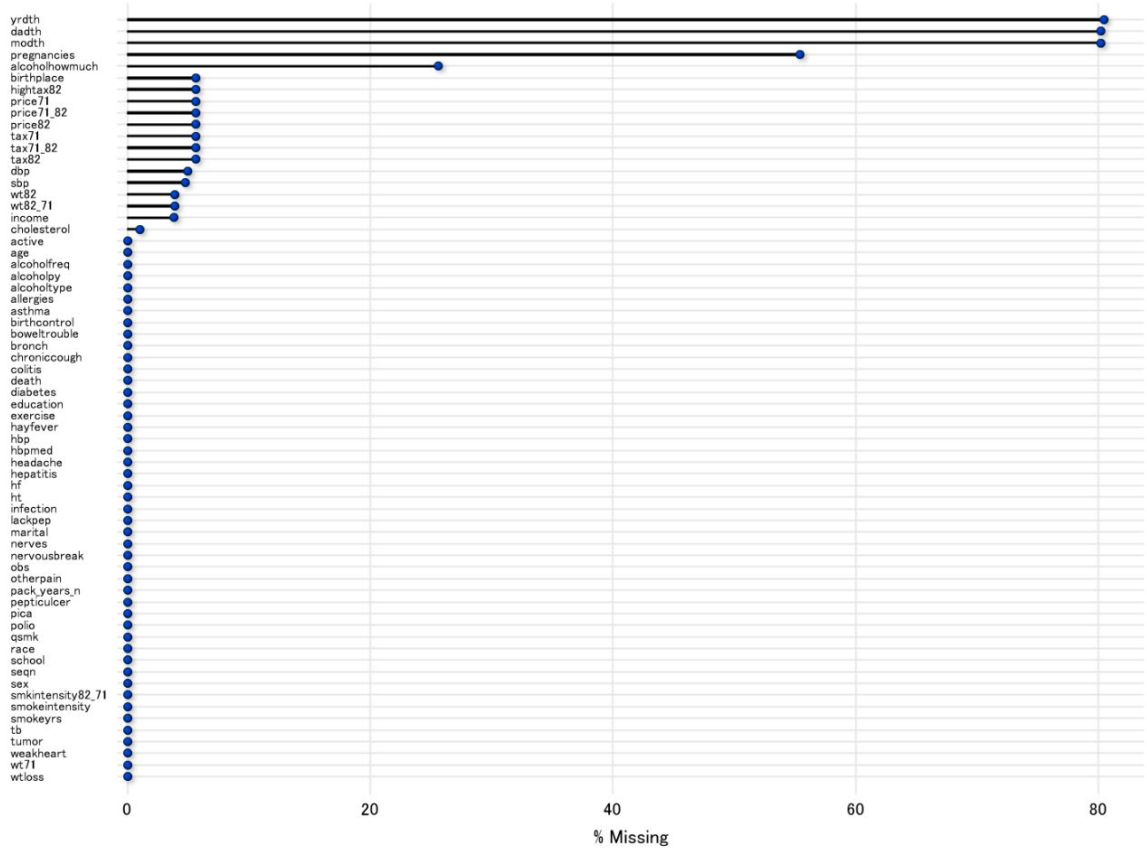
前のプロットにて，変数ごとオブザベーションごとの型判定のデータセットが作成できていれる前提で，カテゴリ集計を行う freq プロシジャで頻度集計をすれば，利用するデータ

セットは完成になります.
ロリーポッププロット自体の構成としては，横向き棒グラフと散布図を組み合わせて表現しています.

```
ods output CrossTabFreqs=NullFreqs(where=(catn=0));
proc freq data=nullmap ;
 table    varname*catn /nopercent nocol nocum;
run;
proc sort data=NullFreqs;
 by    descending RowPercent;
run;
data NullFreqs1;
 set NullFreqs;
 where ^missing(RowPercent);
 y=_N_;
run;
ods graphics / reset=all height=700;
proc sgplot data=NullFreqs1 noborder noautolegend;
   hbarparm category=y response=RowPercent /     barwidth=0.1 baselineattrs=(thickness=0);
   scatter y=y    x=RowPercent /    markerattrs=(symbol=circlefilled size=7)    dataskin=sheen;
   xaxis offsetmin=0.01 offsetmax=0.04 display=( noticks noline) grid label='% Missing';
   yaxis display=( noticks noline novalues nolabel) fitpolicy=split grid;
   yaxistable varname/y=y nolabel position=left;
run;
```

FREQ プロシジャ

度数 行のパーセント	表：varname * catn		
	catn		
varname	Null	Numeric	合計
active	0 0.00	1629 100.00	1629
age	0 0.00	1629 100.00	1629
alcoholfreq	0 0.00	1629 100.00	1629
alcoholhowmuch	417 25.60	1212 74.40	1629
alcoholpy	0 0.00	1629 100.00	1629
alcoholtype	0 0.00	1629 100.00	1629
allergies	0 0.00	1629 100.00	1629



ここまでは、ある一つの変数についての確認でしたが、解析で大きく問題になるのは、Null が変数間で独立していない場合（アウトカム間で、曝露変数の Null の分布に違いがあるなど）です.

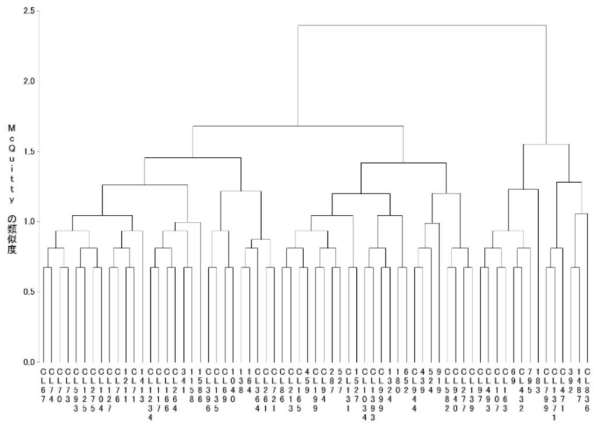
そこで、二変数やデータ全体での欠測値の共有状況についても確認する必要があります.

SAS の CLUSTER プロシジャを使い,R の visdat パッケージの vis_miss で Cluster を指定した際の McQuitty 法のクラスタリングと同様の処理を行いました.

```
proc format;
value preabf
        1     = "Missing"
        0     = "Present";
run;
data nullmap2;
    set NHEFS1;
    nullmap_index=_N_;
    array ar_num seqn--tax71_82;
    do over ar_num;
       if missing(ar_num) then ar_num=1;
     else ar_num=0;
    end;
run;
proc cluster data=nullmap2 outtree=cluster method=mcquitty ;
    id nullmap_index;
    var seqn--tax71_82;
run;
```

さらにその結果をデンドログラム化します.

```
ods output treelisting = tree_list1;
proc tree data = cluster out = tree1 level=0 list ;
run;
```

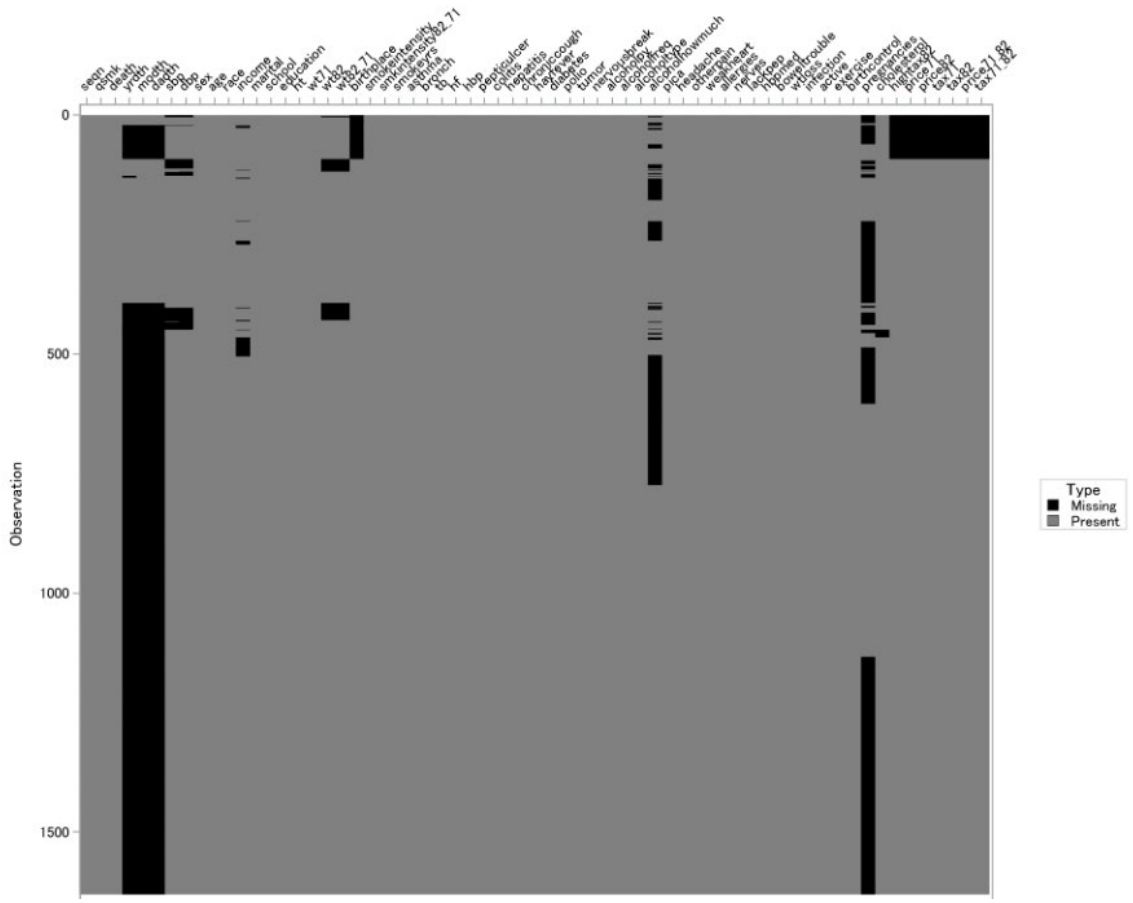


そこから元データに属するクラスタの位置情報を戻して，ビジュアル化した．

R での出力結果に類似の結果がでますが，細かい部分の処理が違い，完全には一致していない．

```
data tree_list2;
    set tree_list1;
    where type="d" and input(scan(batch,2," "),best.)=0;
    length CLUSNAME $12.;
    CLUSNAME = scan(batch,1," ");
    relative_node_pos = _n_;
    drop type batch;
run;
data tree2;
    set tree1;
    CLUSNAME = strip(CLUSNAME);
    nullmap_index = input(strip(_NAME_),best.);
    _NAME_ = strip(_NAME_);
run;
proc sort data = tree2;
by CLUSNAME;
run;
proc sort data = tree_list2;
by CLUSNAME;
run;
data tree3;
    merge tree2 tree_list2;
    by CLUSNAME;
run;
proc sort data = tree3;
by descending relative_node_pos nullmap_index;
run;
data tree4;
    set tree3;
    by descending relative_node_pos nullmap_index;
    retain _cnt;
    if first.relative_node_pos then _cnt = 1;
    else _cnt + 1;
```

```
run;
data tree5;
    set tree4;
    if _cnt = 2 then _cnt = 1;
run;
proc sort data = tree5;
 by descending relative_node_pos descending _cnt nullmap_index;
run;
data tree6;
    set tree5;
    disp_obs = _n_;
run;
proc sort data = tree6;
 by nullmap_index;
run;
proc sort data = nullmap2;
 by nullmap_index;
run;
data nullmap3;
    merge tree6 nullmap2;
    by nullmap_index;
run;
data nullmap4;
length varname $20.;
set nullmap3;
array ar_num seqn--tax71_82;
do over ar_num;
 varname=vname(ar_num);
 varn=_i_;
 catn=ar_num;
 output;
end;
format catn preabf.;
keep varname nullmap_index catn disp_obs;
run;
data mapping2;
length ID Show FillColor Value $20.;
ID='catn';
Show='AttrMap';
do Value ="Missing","Present";
 select(Value);
   when("Missing") FillColor="black";
   when("Present") FillColor="gray";
 end;
 output;
end;
run;
ods html path="&outpath." file="nullmap_cluster.html" image_dpi=150;
ods graphics /reset=all   nxybinsmax=9999999   ANTIALIASMAX=9999999 width=1200 height=1000;
proc sgplot data=nullmap4 dattrmap=mapping2;
    format catn preabf.;
    heatmapparm x=varname   y=disp_obs colorgroup=catn /   attrid=catn x2axis;
    x2axis;
    yaxis reverse label="Observation";
    keylegend / title="Type" location=outside position=right across=1;
run;
```

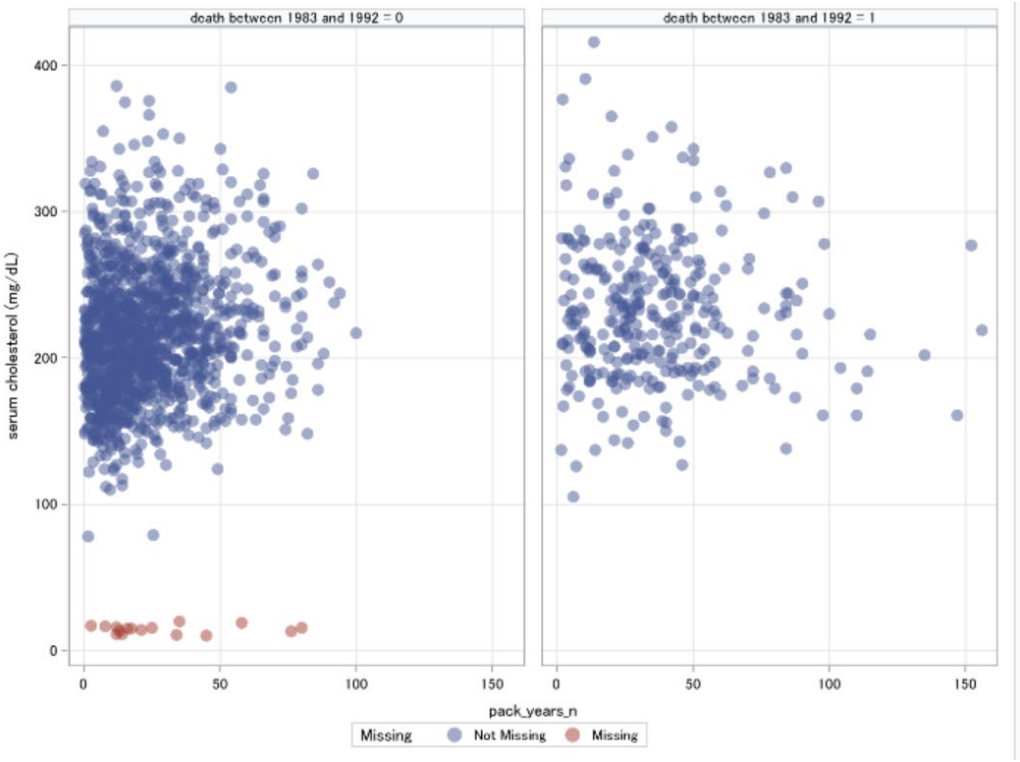


2 変量の関係性を層別変数でパネル分けして比較することが有用なケースがある.
またいずれかの変数に欠測がある場合，プロットから落とさずに色分けして表示することで，層別変数ごとの欠測の具合をみることができます.（その場合，欠測側の軸はジッタリングの要領で適当に乱数で散らしている）
SAS の場合,Sgpanel プロシジャの panelby ステートメントでパネル分割ができ，他の書き方は Sgplot とそれほど変わりません. 小技として transparency で透過度を調整しています.

```
data NHEFS2;
length cholesterol_FL $20.;
 set NHEFS1;
 if n(cholesterol , pack_years_n)=2 then cholesterol_FL="Not Missing";
 else do;
 cholesterol_FL="Missing";
 cholesterol=10+rand("uniform")*10;
end;
label cholesterol_FL="Missing";
run;

proc sgpanel data=NHEFS2;
   panelby    death/spacing=10;
   scatter x = pack_years_n y = cholesterol /
      transparency=0.5 group=cholesterol_FL markerattrs=(size=12 symbol=circlefilled);
```

```
   rowaxis grid;
   colaxis grid;
run;
```



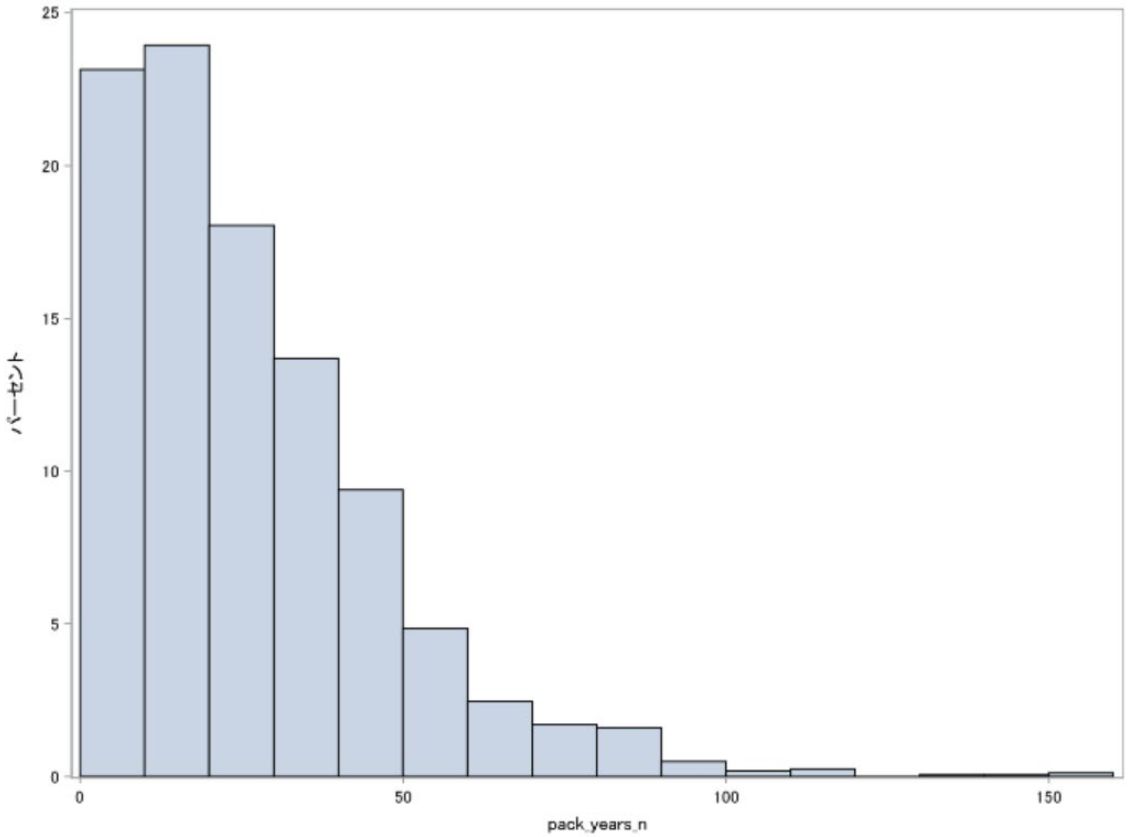
3. Continuous variable (連続変数)

まずは,Part 3 でも使用する Pack-Year を用いて,連続変数（一変量および二変量）の可視化方法を確認していきます.

3.1 ヒストグラム

ヒストグラムについてはプロットステートメントで histogram と対象変数を指定するのみとなります.

```
proc sgplot data=NHEFS1;
  histogram pack_years_n;
run;
```



3.2 密度関数プロット

SAS でカーネル密度推定を行う場合は kde プロシジャを使用します．層別因子で事前にソートして，by を使用することで層別に処理が可能です．

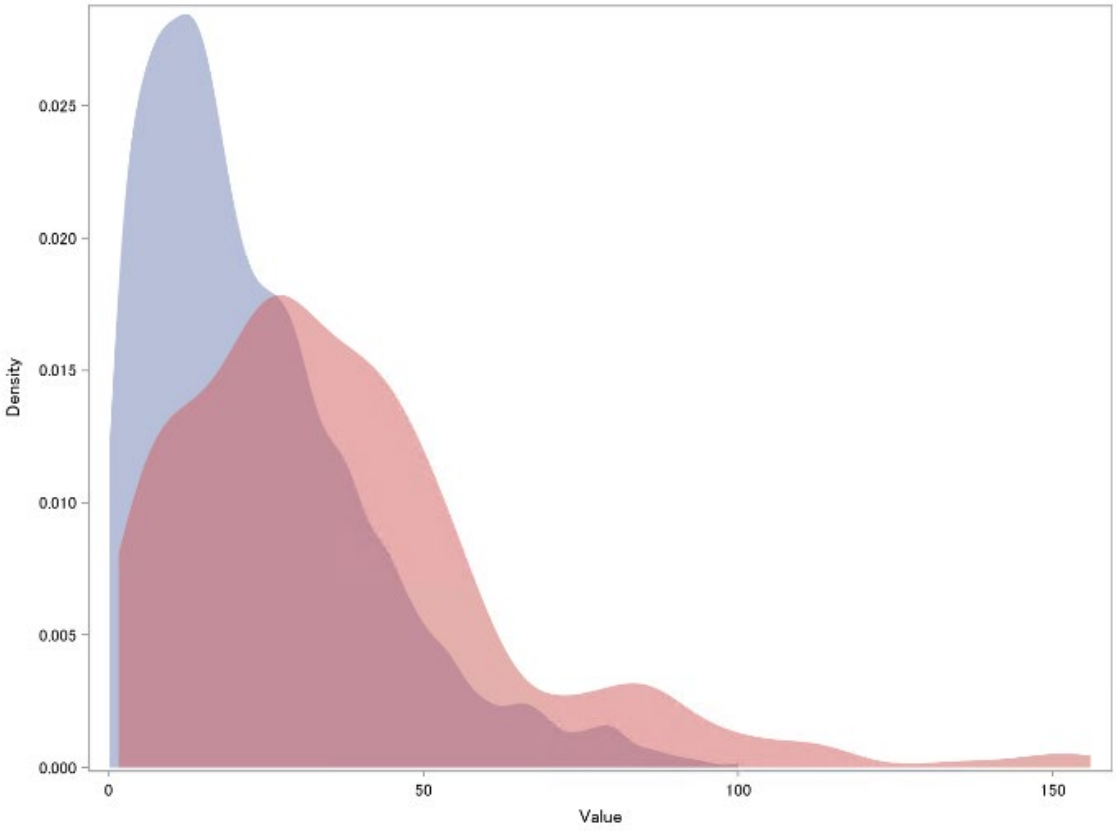
kde プロシジャに truncate オプションをつけることで，実測データの範囲を超える密度推定部分をカットすることができます

あとは結果を帯プロット作成用の band プロットプロシジャを使えば描画できます．

```
proc sort data=NHEFS1;
  by death;
run;
proc kde data = NHEFS1 ;
  univar pack_years_n / truncate out=kde ;
  by death;
run;

proc sgplot data=kde noautolegend;
band upper=density x=value lower=0 / transparency=0.5      group=death fill ;
```

```
xaxis offsetmin=0.02 offsetmax=0.02;
run;
```

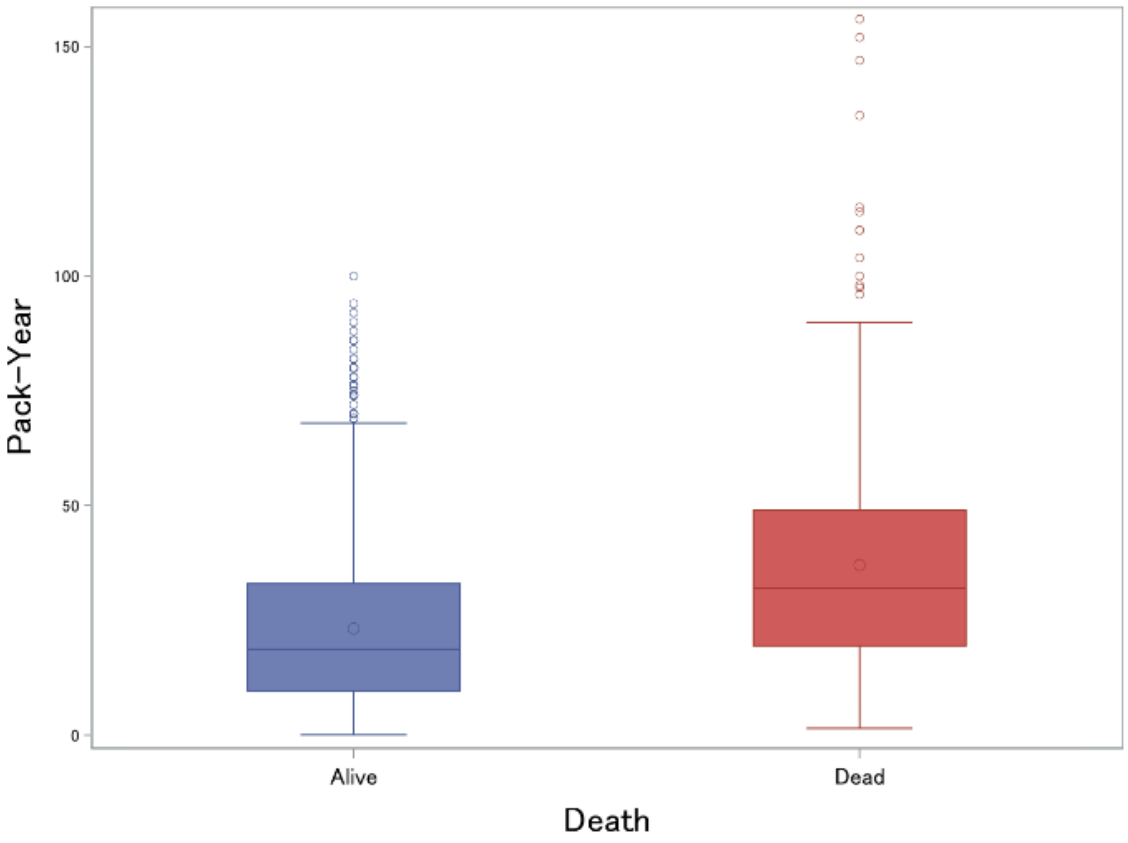


3.3 箱ひげ図

縦向きの箱ひげ図の場合，プロットステートメント vbox を使用します．group オプションと category オプションは似た概念ですが，category の方が上層の層分けとなっています．今回，両方使用しているのは category で分けられた x 軸が値をもつ性質が都合よく，group を使っているのは色分けをするためです．

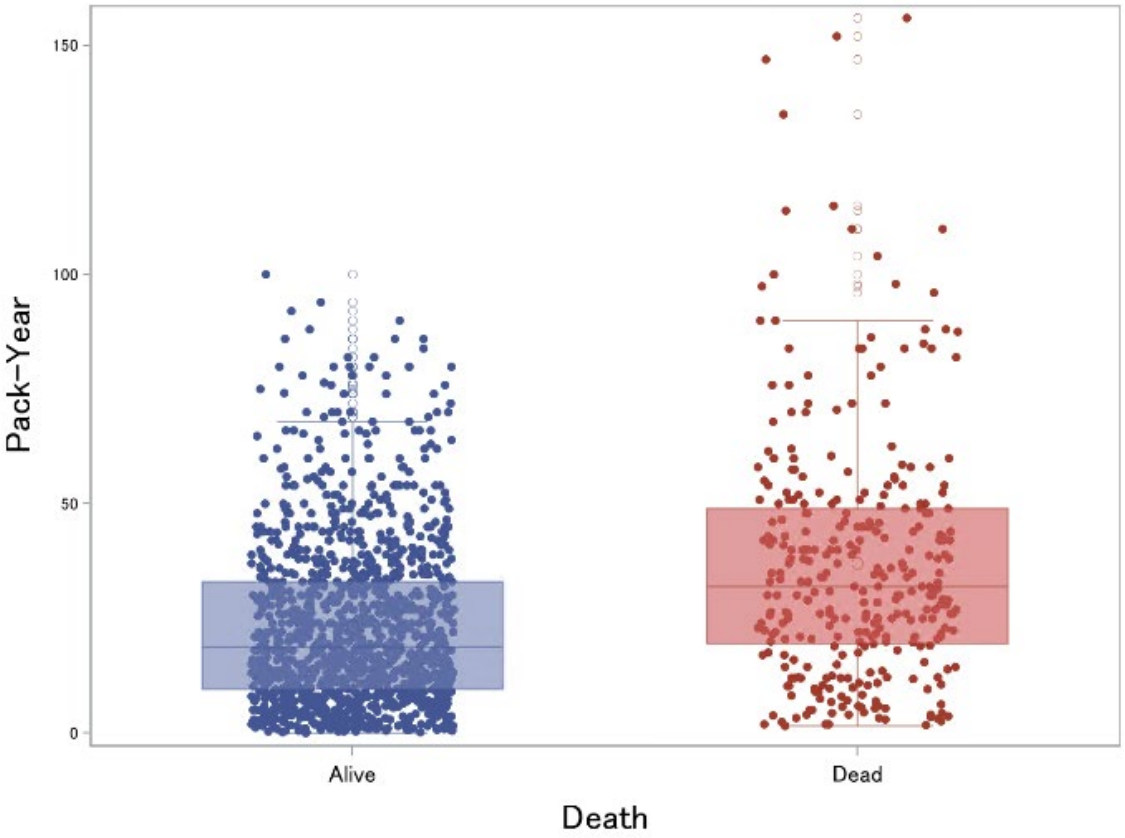
x 軸の death の 0 ,1 の値に対して format を当てて文字ラベルをつける方法も有効ですが，Valuesdisplay で直接指定することも可能です．軸値のラベルが凡例の意味となっているので noautolegend で凡例の生成を阻害しています．

```
proc sgplot data=NHEFS1 noautolegend;
 vbox pack_years_n / category=death group=death;
 xaxis values=(0 1) valuesdisplay=("Alive" "Dead") valueattrs=(size=12pt) label="Death"
labelattrs=(size=18pt);
 yaxis  label="Pack-Year" labelattrs=(size=18pt);
run;
```



scatter に jitter オプションを追加することで，データをずらすジッタリングが可能であり，箱ひげ図に重ねることもできます.

```
proc sgplot data=NHEFS1 noautolegend;
 scatter x=death y=pack_years_n / group=death
                    jitter markerattrs=(symbol=circlefilled);
 vbox pack_years_n / category=death transparency=0.4 group=death;
 xaxis values=(0 1) valuesdisplay=("Alive" "Dead") valueattrs=(size=12pt) label="Death"
labelattrs=(size=18pt);
 yaxis  label="Pack-Year" labelattrs=(size=18pt);
run ;
```



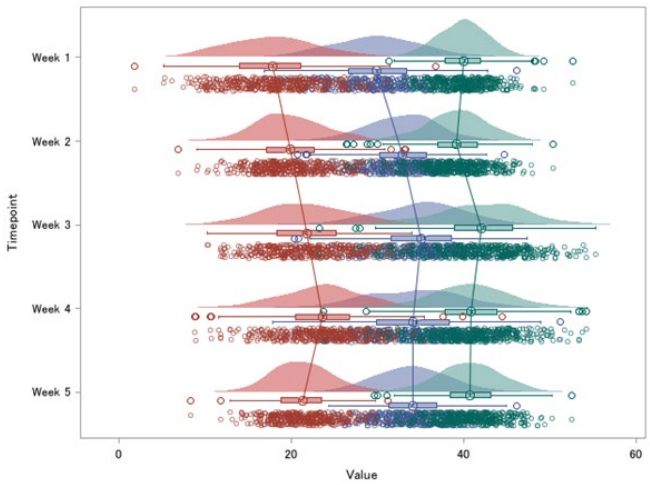
3.4 レインクラウドプロット

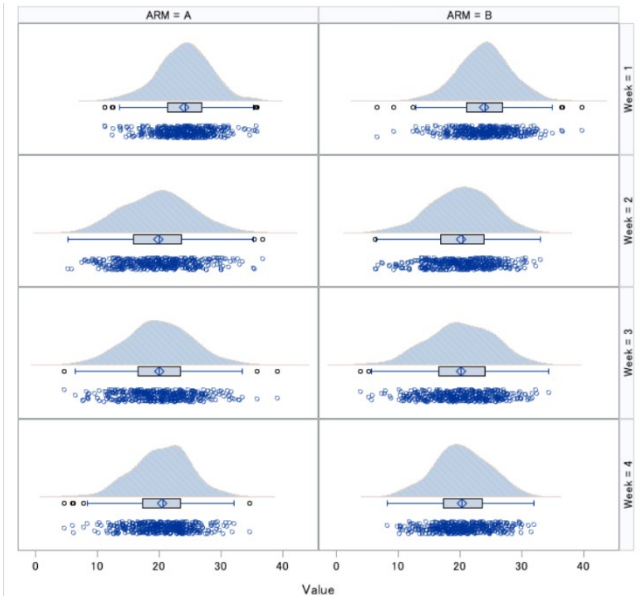
これまでみてきた連続量の可視化について，分布・要約統計量・各データの全てを網羅的に可視化する方法がレインクラウドプロットです.

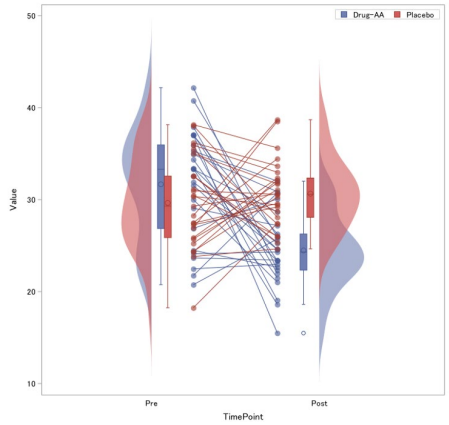
SAS で Rainclod plot を実装するのは Graph Template Language というグラフ設計文法と Sgrender プロシジャを使用するのがおすすめで，これまで紹介したプログラムを再編成するイメージなので，それほど難しくはないのですが，コード量はかなりボリュームがあるのと，Graph Template Language をここで一から解説すると，混乱を招きそうです.

SAS ユーザー総会という会にて「SAS による Raincloud Plot の実装」という演題で，論文・発表資料・プログラムなどをすべて無償公開しておりますので，SAS で Raincloud plot を描きたいんだ！という方はぜひ参考にしてください. 以下には，実際の作成イメージのみ提示してます.

https://www.sas.com/ja_jp/events/22/users-group-2022/sessions.html#m=a-1


Week 1
Week 2
Week 3
Week 4
Week 5
Timepoint
0
20
40
60
Value


ARM = A
ARM = B
Week = 1
Week = 2
Week = 3
Week = 4
0
10
20
30
40
Value


Drug-AA
Placebo
50
40
30
20
10
Value
Pre
Post
TimePoint

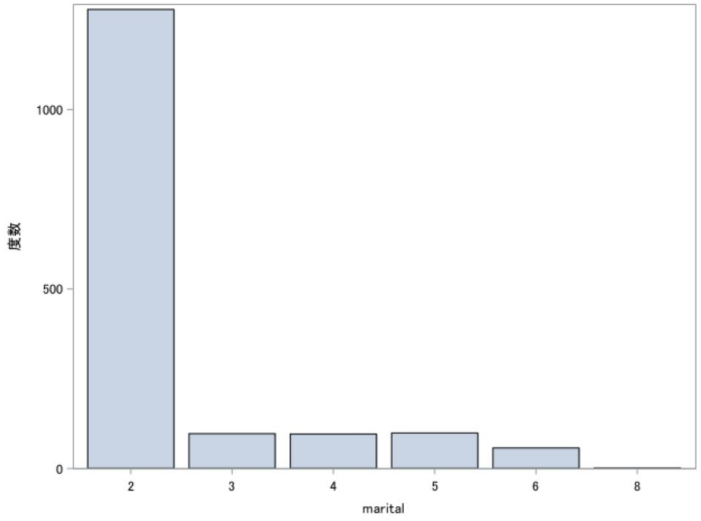
4. Nominal variable（名義変数）

ここでは，複数のカテゴリを持つ marital （婚姻状況）を用いて，名義変数（一変量および二変量）の可視化方法を確認していきます.

4.1 棒グラフ

もっとも単純な，頻度の棒グラフであれば，プロットステートメント vbar に変数を指定するのみで描画されます.

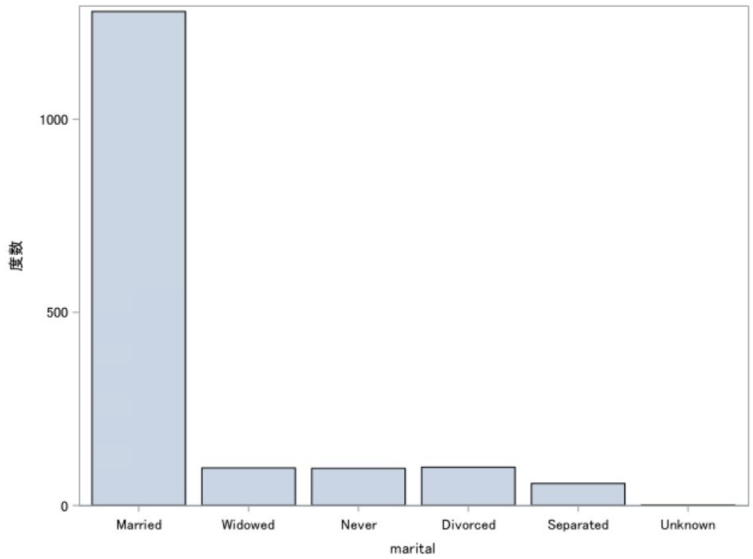
```
proc sgplot data=NHEFS1;
 vbar marital;
run;
```



値にラベルを付けたい場合は format プロシジャで出力フォーマットを作成して利用します.

```
proc format;
 value maritalf
 1="Under17"
 2="Married"
 3="Widowed"
 4="Never"
 5="Divorced"
 6="Separated"
 8="Unknown"
 ;
 run;
```

```
proc sgplot data=NHEFS1;
 vbar marital;
 format marital maritalf.;
run;
```



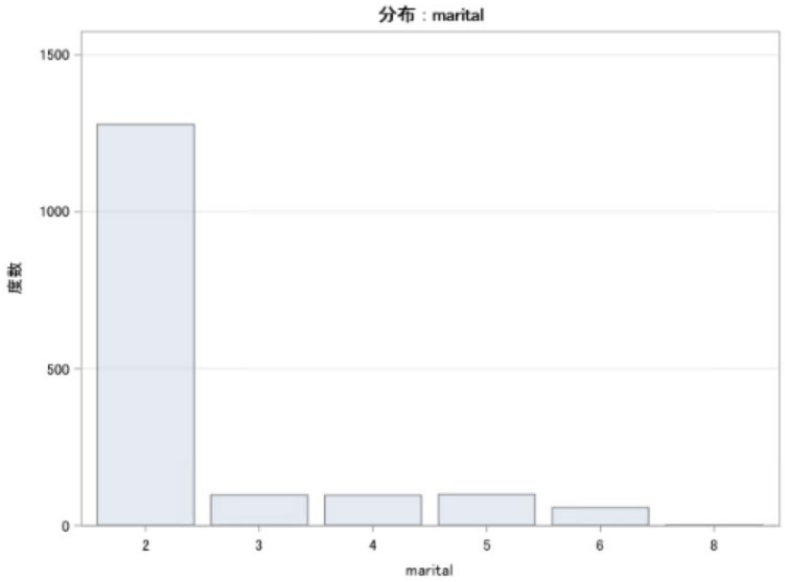
SAS には ODS 統計グラフという機能があります. これは統計解析プロシジャを実行した際に，その解析に関連の強い統計グラフを合わせて出力するという機能です.
カテゴリカルデータ解析の代表的なプロシジャである FREQ プロシジャの統計グラフ機能を使うと，頻度集計表と，棒グラフを同時に得ることができる．plot = に all を指定すると，その際の FREQ プロシジャの指定状況に応じて，作成可能なグラフがすべて描画されます.
例えば以下の場合，単純な頻度棒グラフと累積の頻度棒グラフが作図されます.

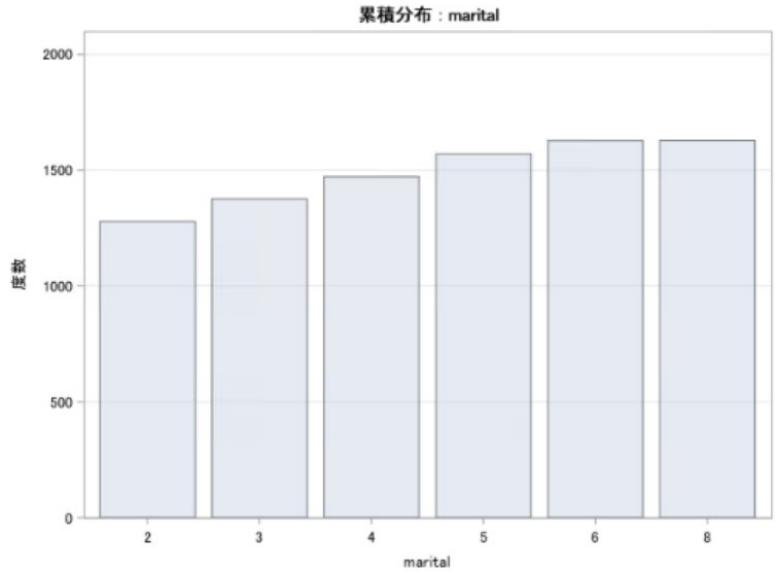
```
ods graphics on;
proc freq data=NHEFS1;
 tables marital / plot=all;
run;
```

FREQ プロシジャ

marital

marital	度数	パーセント	累積度数	累積パーセント
2	1279	78.51	1279	78.51
3	97	5.95	1376	84.47
4	96	5.89	1472	90.36
5	99	6.08	1571	96.44
6	57	3.50	1628	99.94
8	1	0.06	1629	100.00





4.2 横並び棒グラフ

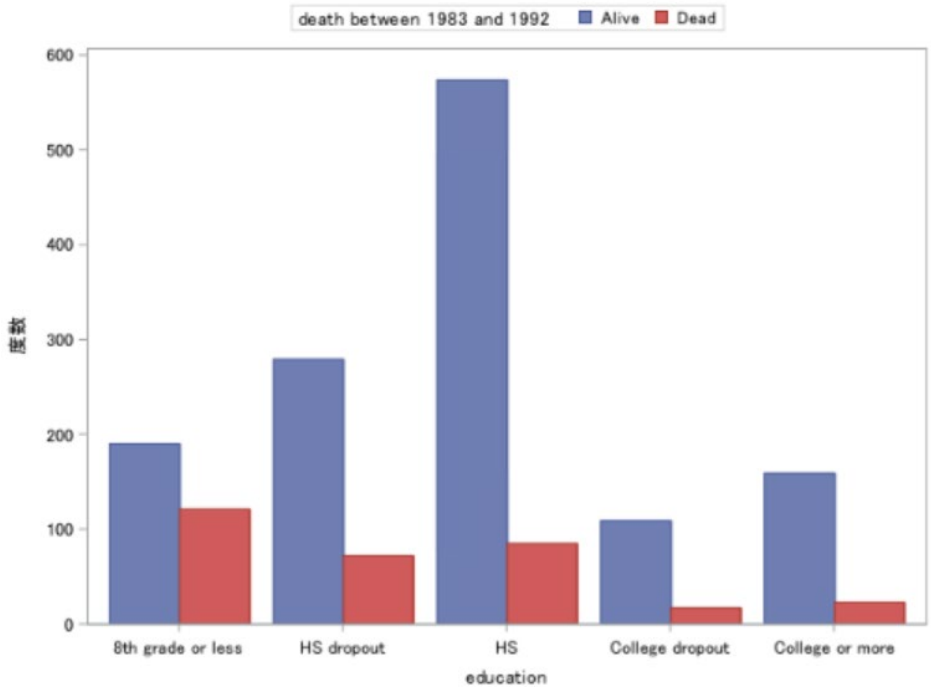
先ほどは，一変量の棒グラフでした．今度は Part 3 で使用する death のアウトカムで層別化した教育歴を確認します．

表示用のフォーマットを作ることと，group= に death を指定するとともに，groupdisplay=cluster にすることがポイントになります．cluster を指定することで横に並びますが，指定しないと stack がデフォルト設定になっているため，グループが積み上げ式で表現されます．

```
proc format;
 value eduf
 1="8th grade or less"
 2="HS dropout"
 3="HS"
```

```
 4="College dropout"
 5="College or more"
 ;
 value deathf
0="Alive"
1="Dead"
;
 run;

proc sgplot data=NHEFS1;
 vbar education /group=death groupdisplay=cluster;
 keylegend /location=outside position=top;
 format education eduf. death deathf.;
run;
```

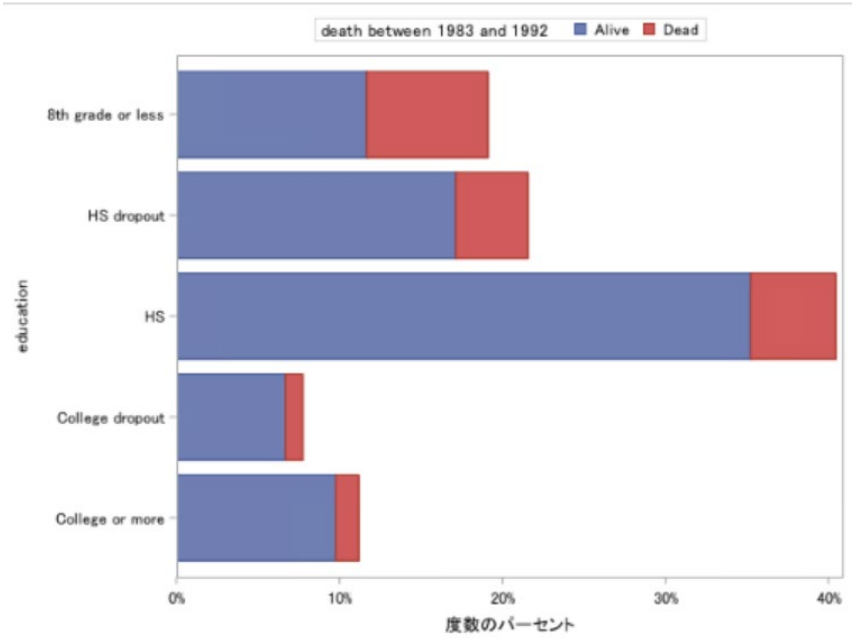


4.3 100%積み上げ棒グラフ

今度は絶対数ではなく，割合を確認してみたいので，100%積み上げグラフを使用します．
stat=オプションを percent（デフォルトは freq）にすることにより割合で計算されることになります．
groupdisplay=stack はデフォルトなので明示指定はなくても OK です．
ただし，ここで計算される割合は，全体数を分母にした割合であることに注意してください．
各水準の中で death の割合を見たい場合は次で説明します．

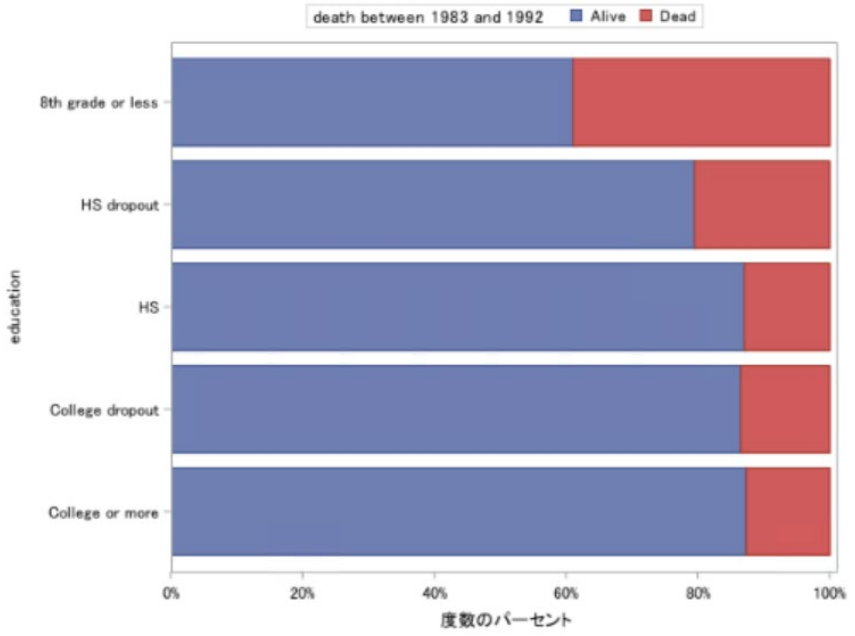
```
proc sgplot data=NHEFS1 ;
 hbar education /stat=percent    group=death groupdisplay=stack;
 keylegend /location=outside position=top;
 format education eduf. death deathf.;
```

```
run;
```



先ほどのコードに pctlevel=group というのを sgplot 自体のオプションとして使用することで，stat=percent の場合の分母が各水準の総数に変更されます.

```
proc sgplot data=NHEFS1 pctlevel=group;
 hbar education /stat=percent    group=death groupdisplay=stack;
 keylegend /location=outside position=top;
 format education eduf. death deathf.;
run;
```

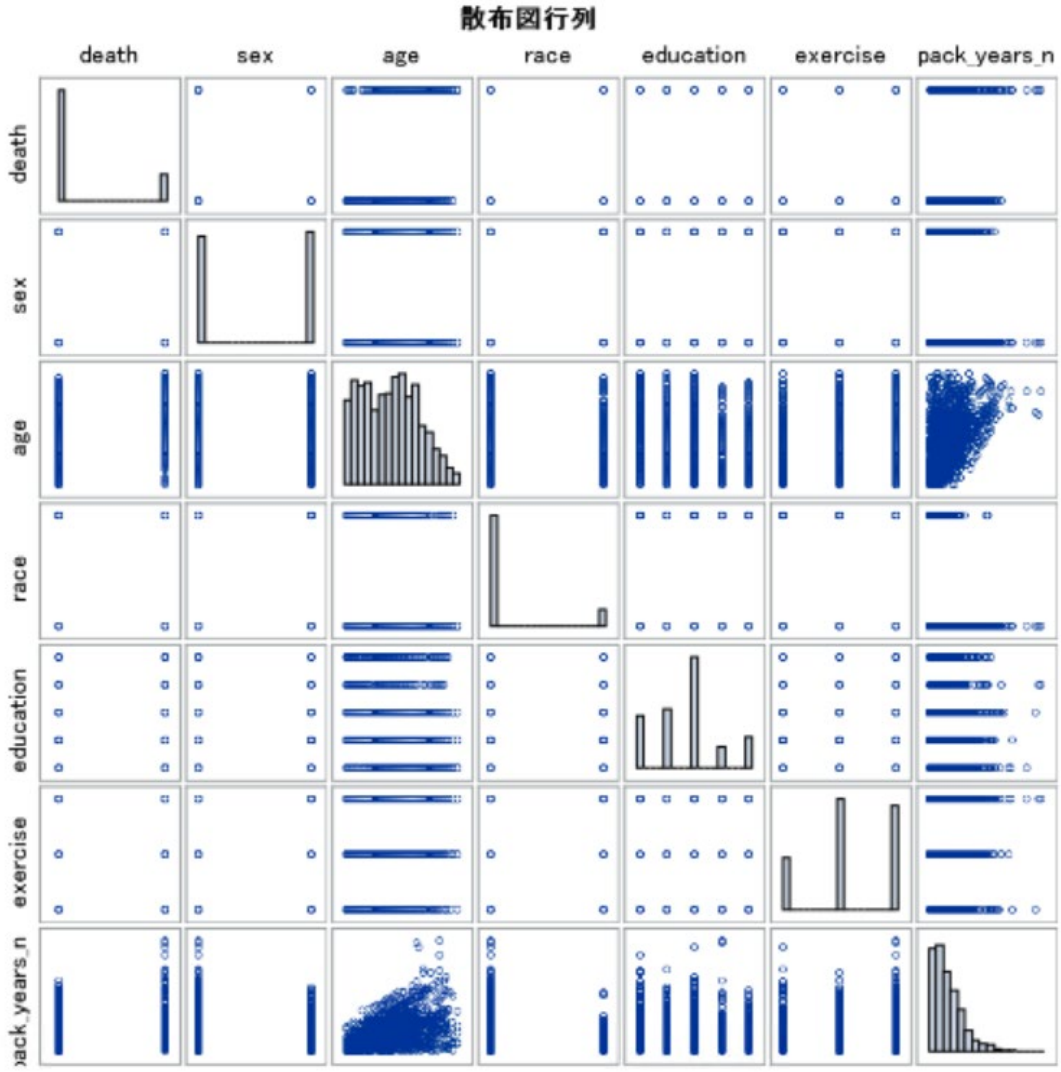


5. Multivariable（多変量）

次のパートで利用する変数どうしの関係を可視化したいと思います．曝露変数（Pack-Year）と結果変数（Death）の列に注目して確認するようにしましょう

今回は，指定が簡便なので，相関係数を算出する corr プロシジャの統計グラフ機能を使います．不用意に巨大要領の描画が起きないようにリミットがかかっている(5000 データ，5 変数)ので，確認したうえで maxpoint オプションや nvar オプションで解除しておきます

```
proc corr data=NHEFS1 plots(maxpoints=999999)=matrix(histogram nvar=10);
var death sex age race education exercise pack_years_n;
run;
```



作成者：森岡　裕(イーピーエス株式会社)
Mail: morioka.yutaka038@eps.co.jp