Part 2: 記述統計の可視化

2 Data

データの概要を確認するためのコマンドです（いずれも Stata1.0 から存在する古参コマンドです)。

part2.do

```
des
sum
list in 1/10
```

3 Overview & Missing values（欠測値）

3.1 Overview

Stata で欠損値の概要を示すためには **misstable** コマンドを利用します。コマンドの詳細は **help misstable** にて確認できます。R での実行結果とは異なり、欠損値の状況は表として示されます。

part2.do

```
misstable sum
misstable pattern
misstable tree
misstable nested
```

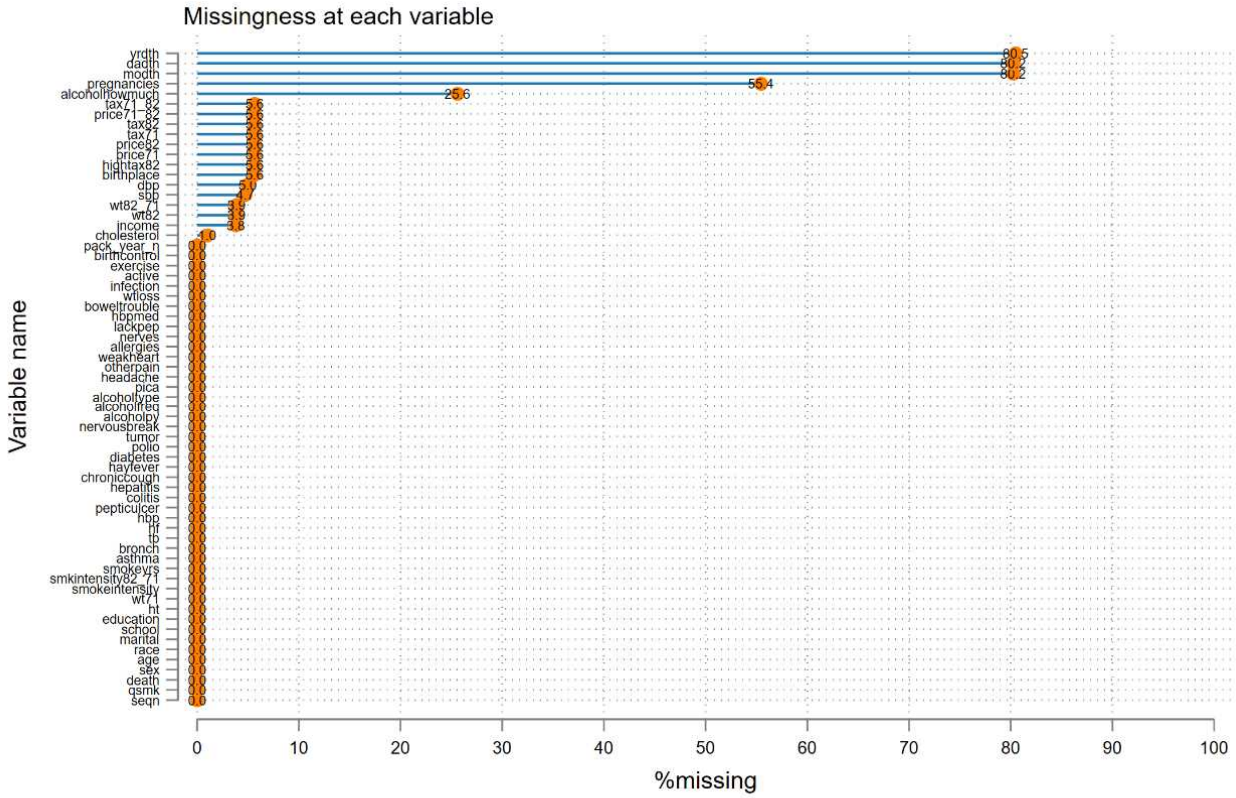
3.2 Missingness at each variable

Stata では標準で lollipop plot を行なうためのコマンドが存在しませんし、調べた限りでは外部コマンドでも lollipop plot をサポートしているコマンドがありませんでした。そこで、欠損値割合を lollipop plot として表示するプログラムを作成しました（**missing_lollipop_plot.do**)。

近日中に、外部コマンド（ado file）として作成公開しようと考えています。中身は 100 行くらいになるので、ここでは全体を記載していません。

part2.do

```
qui: do missing_lollipop_plot // program 定義の読込み
missing_lollipop_plot
```



3.3 Missingness relationship between variables

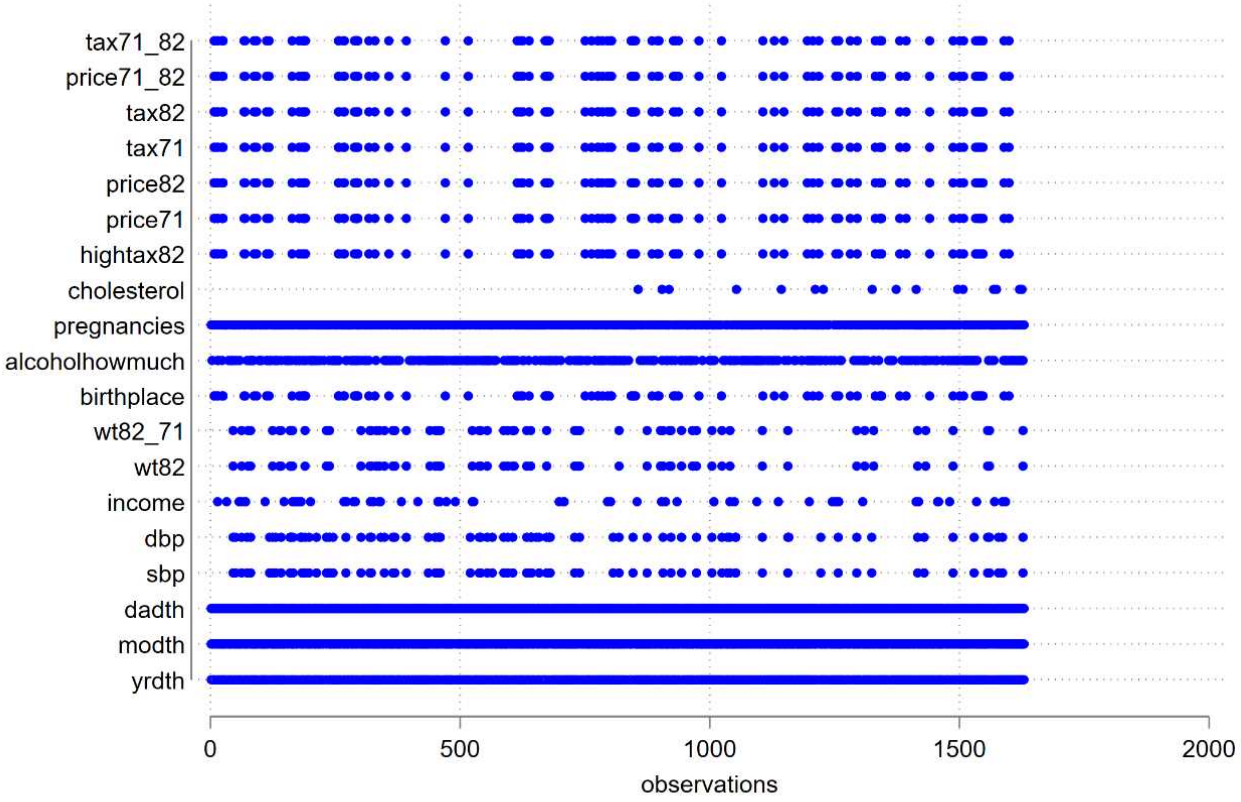
3.3.1 全ての変数

R における **vis_miss** 関数に相当する Stata 標準コマンドはありませんでしたが、外部コマンドとして **missingplot** がありました。これは Nicolas J Cox 先生が作成されたものですが、若干古いので、プロットのビジュアルが **vis_miss** 関数よりも見栄えしないことが欠点です。

missingplot コマンドは標準では、欠損値がある変数だけをプロットします。

part2.do

```
missingplot, scheme(white_tableau) msymbol(oh)
```



missingplot コマンドでは、残念ながらプロットサイズを小さくしたとしても、欠損値が多い変数（**yrdth**）などでは、全欠損しているように見えてしまいます。

次に、死亡の有無（**death**）で層別化した場合に、特定の変数（**alcoholhowmuch**）の欠損の割合に違いがあるかを確認します。

3.3.2 連続変数 × 名義変数

Stata では、**misstable sum** コマンドを用いて、欠損値数を表示させることができます。ここで if を繋げる事で、死亡の有無で層別します。

part2.do

```
misstable sum alcoholhowmuch if death==0
misstable sum alcoholhowmuch if death==1
```

生存では欠損値が 308 件、死亡では欠損値が 109 件であることが示されます。**misstable** コマンドでは割合は計算されませんので、手計算で行なう必要があります。

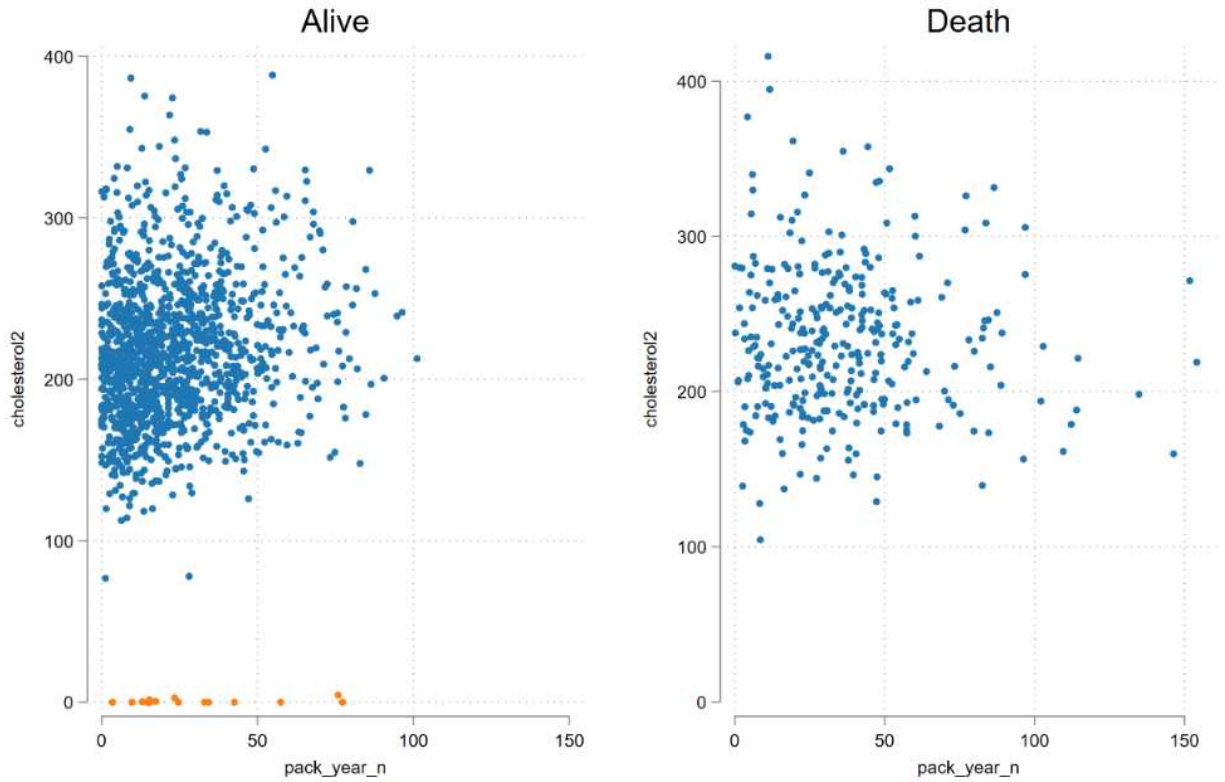
3.3.3 連続変数 × 連続変数

次は、2 つの連続変数間について、一方が欠損値の場合にもう一方の変数の分布がどのようになっているかを確認します。

part2.do

```
set seed 12345

gen cholesterol2 = cholesterol
recode cholesterol2 (.=0)
capture graph drop *

// 生存者におけるグラフ
twoway ///
       (scatter cholesterol2 pack_year_n ///
              if death ==0 & cholesterol!=., jitter(2 2)) ///
      (scatter cholesterol2 pack_year_n ///
              if death ==0 & cholesterol==., jitter(2 2) ///
      ylabel(0(100)400) xlabel(0(50)150) ///
      scheme(white_tableau) legend(off) title("Alive") name(foobar1, replace))

// 死亡者におけるグラフ
twoway ///
       (scatter cholesterol2 pack_year_n ///
              if death ==1 & cholesterol!=., jitter(2 2)) ///
      (scatter cholesterol2 pack_year_n ///
              if death ==1 & cholesterol==., jitter(2 2) ///
      ylabel(0(100)400) xlabel(0(50)150) ///
      scheme(white_tableau) legend(off) title("Death") name(foobar2, replace))

// 2 つのグラフを結合
graph combine foobar1 foobar2, ///
      cols(2) name(chol_smk, replace) scheme(white_tableau)
```

scatter コマンドで散布図のプロットを行なっています。欠損値の有無、死亡の有無で分けて描画していますので、都合 4 回 **scatter** を行い、それらを結合させています。**jitter** オプションで撒布をランダムに散らしていますので、**set seed 12345** によって乱数固定をかけなければ、毎回微妙に異なるグラフになります。



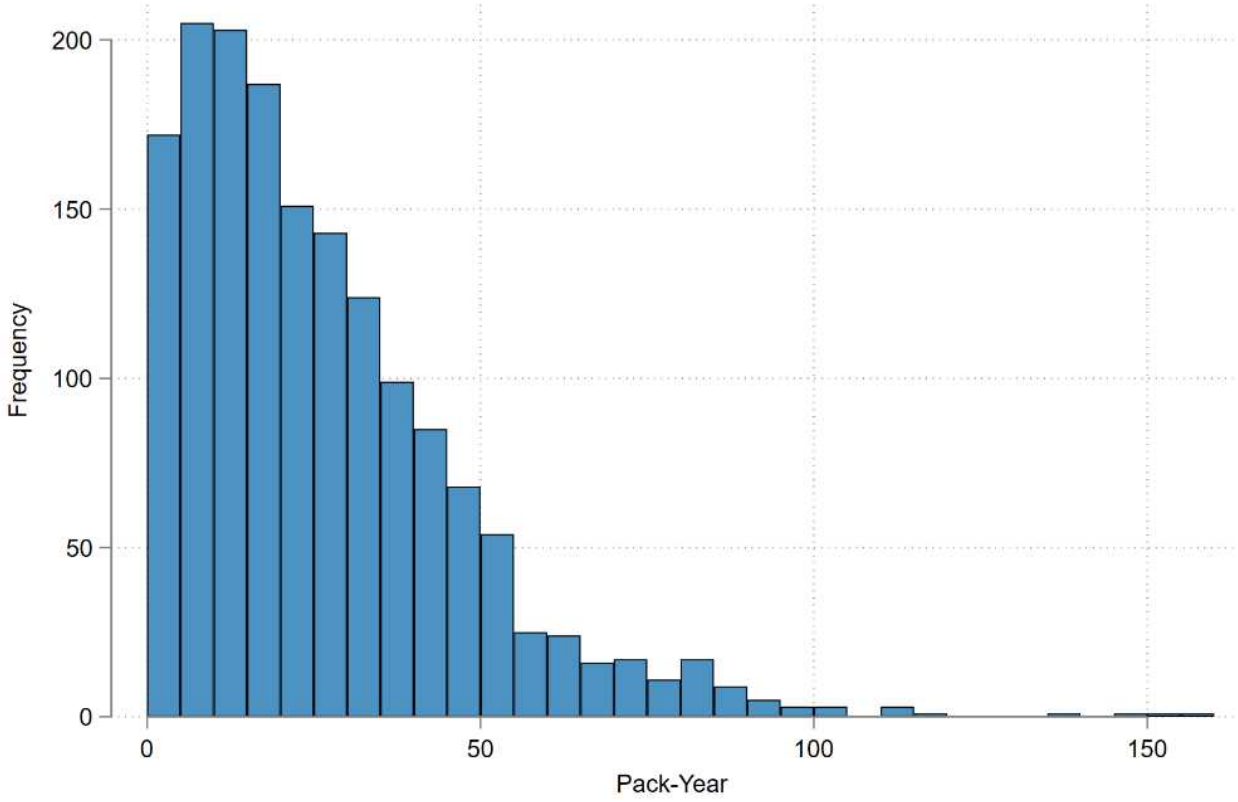
4 Continuous variable（連続変数）

4.1 ヒストグラム

Stata では **histogram** サブコマンドでヒストグラムを描画します。

part2.do

```
twoway histogram pack_year_n, ///
      start(0) width(5) freq /// BIN の開始、BIN 幅、縦軸の指定
      scheme(white_tableau) /// グラフテーマ
      xtitle("Pack-Year") /// 横軸タイトル
      name(hist, replace)
```

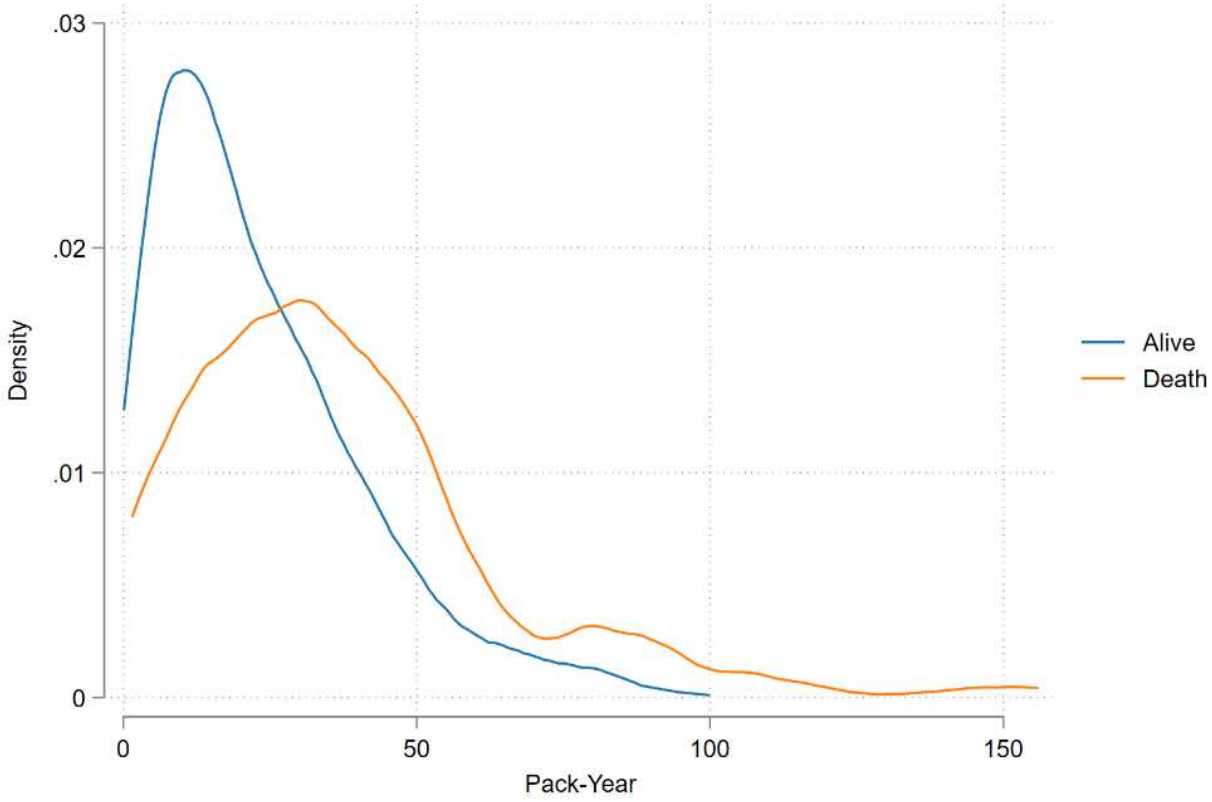


4.2 密度関数プロット

Stata では **kdensity** サブコマンドで密度関数プロットを描画します。**kdensity** サブコマンドを 2 回使用しています。一回目では **if death==0** を指定し、二回目で **if death==1** を指定し、生存者・死亡者のグラフを描いています。

part2.do

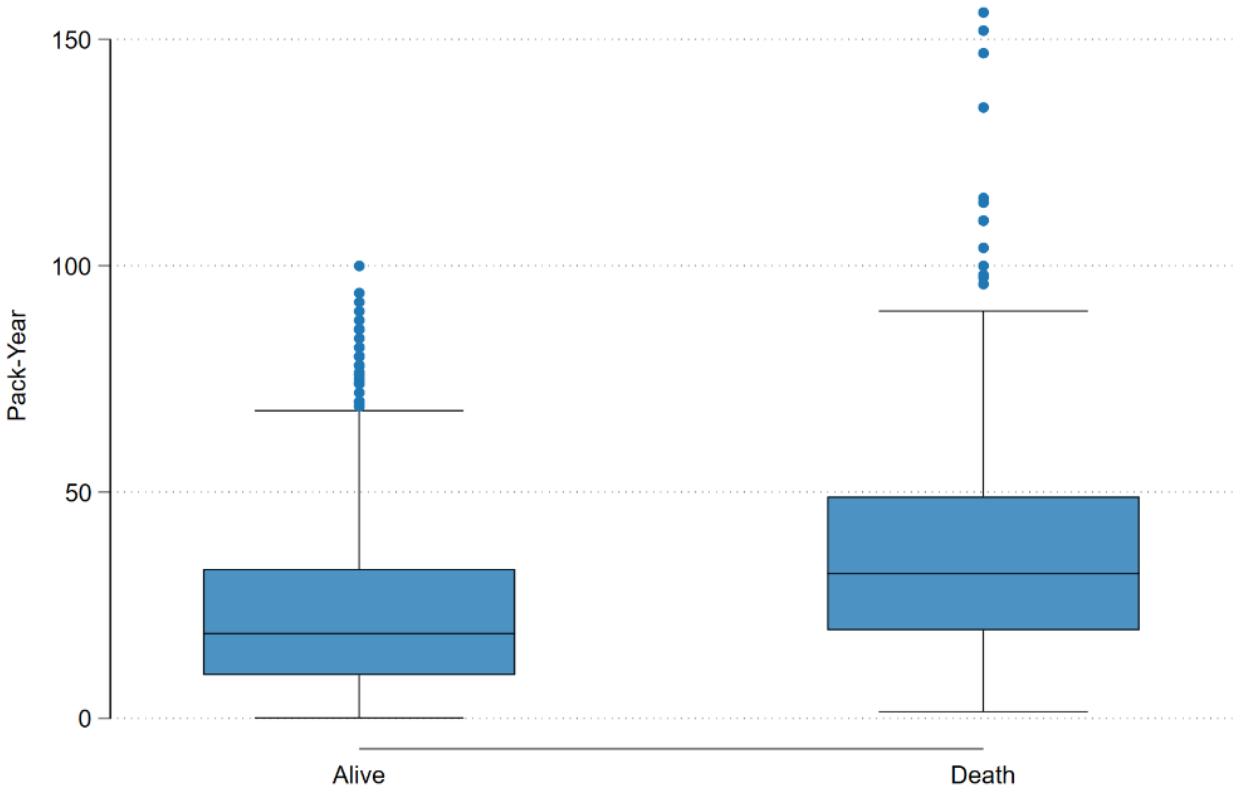
```
twoway ///
      (kdensity pack_year_n if death==0) /// 生存者のグラフ
      (kdensity pack_year_n if death==1, /// 死亡者のグラフ
      scheme(white_tableau) /// グラフテーマ
      legend(label(1 "Alive") label(2 "Death")) /// 凡例の指定
      xtitle("Pack-Year") /// X 軸タイトル
      ytitle("Density") /// Y 軸タイトル
      name(dens, replace))
```



4.3 箱ひげ図

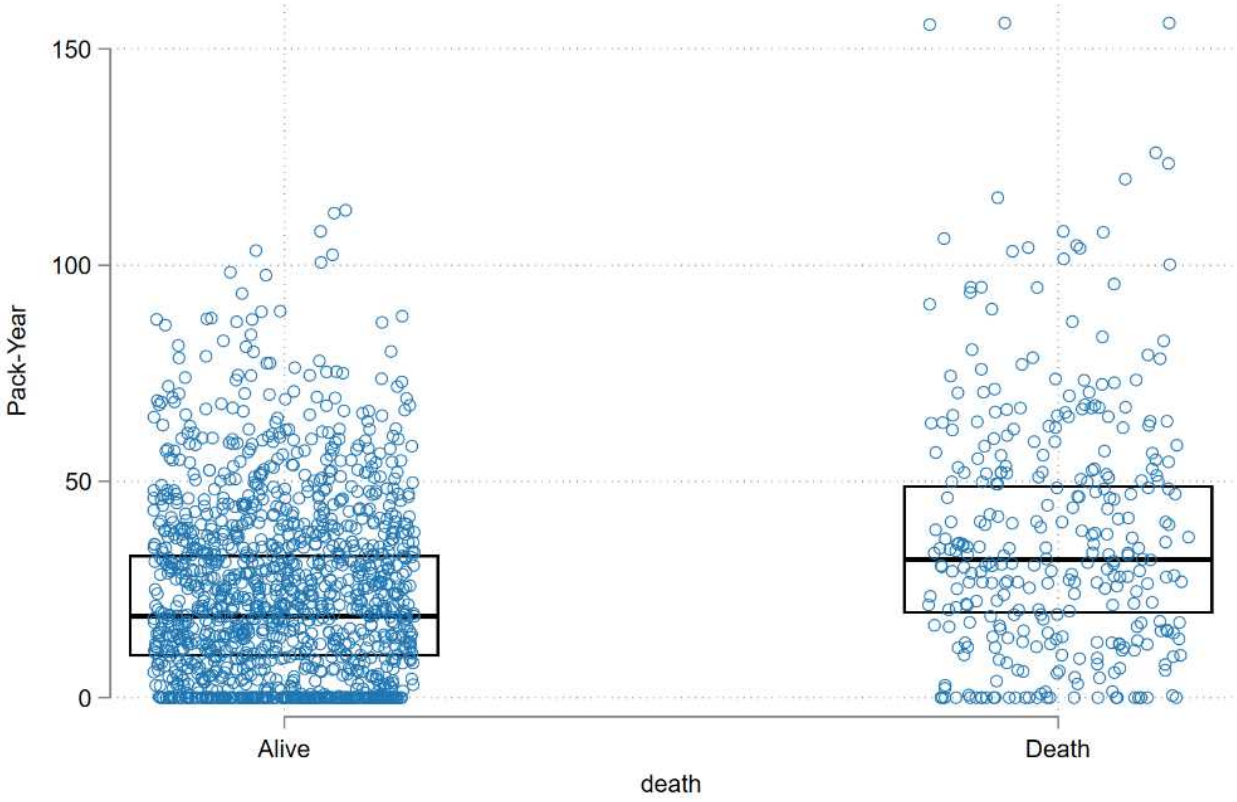
Stata での箱ひげ図は、ここまでで利用してきた **twoway** 系列のコマンドとは別に用意されています。**graph** コマンドのサブコマンド **box** を利用して描画します。

```
graph box pack_year_n , ///
      scheme(white_tableau) /// グラフテーマ
      ytitle("Pack-Year") /// Y 軸
      over(death) /// death によってグラフを分ける。なお by(death)も可
      name(box_over, replace)
```



箱ひげ図にドットプロットを載せるには、通常のコマンドでは難しく、外部コマンド **stripplot** を使用する必要があります。このコマンドを用いるとドットプロットに箱ひげ図を載せるイメージでグラフが作成できます。

```
stripplot pack_year_n, ///
      scheme(white_tableau) /// グラフテーマ
      vertical /// 縦向きのグラフ
      ytitle("Pack-Year") /// Y 軸
      over(death) /// death によってグラフを分ける
      box /// BOX プロットを追加
      jitter(30 10) /// ドットを散らす
      name(box_dot, replace)
```



4.4 ドットプロット

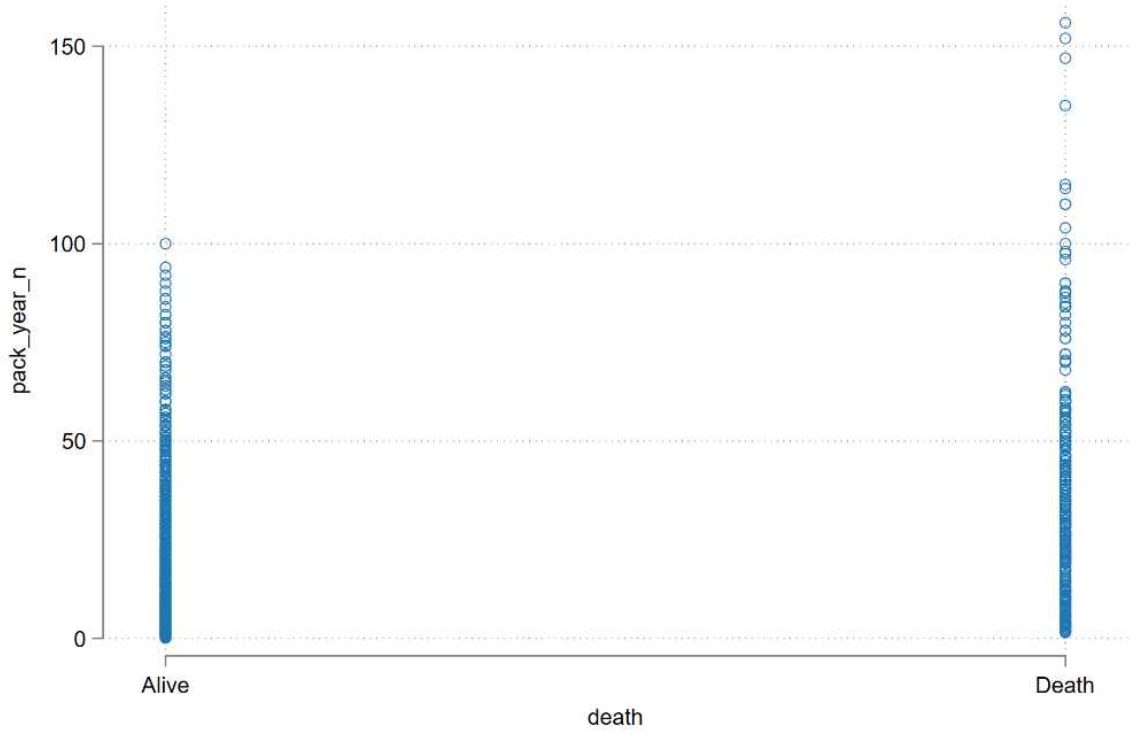
Stata においてドットプロットの描画には、いくかの方法があります。1 つめは上記で使用した **stripplot** コマンドを利用する方法です。その他に **dotplot** コマンドや **gr7** コマンドがあります。**dotplot** コマンドは Stata 社公式のコマンドですが、やや柔軟性に欠けています。**gr7** は公式の古いコマンドですので、利用している人があまりいませんし、やや時代がかったグラフが描画されます。

この中では **stripplot** コマンドが一番柔軟に描画可能です。例えば、縦向き（**vertical**）グラフと横向き（**horizontal**）グラフの両方が描画できるのは **stripplot** コマンドのみです。

ここでは、**stripplot** コマンドを用いた描画方法を紹介します。

4.4.1 何も施していないドットプロット

```
stripplot pack_year_n, sheme(white_tableau) over(death) vertical
```

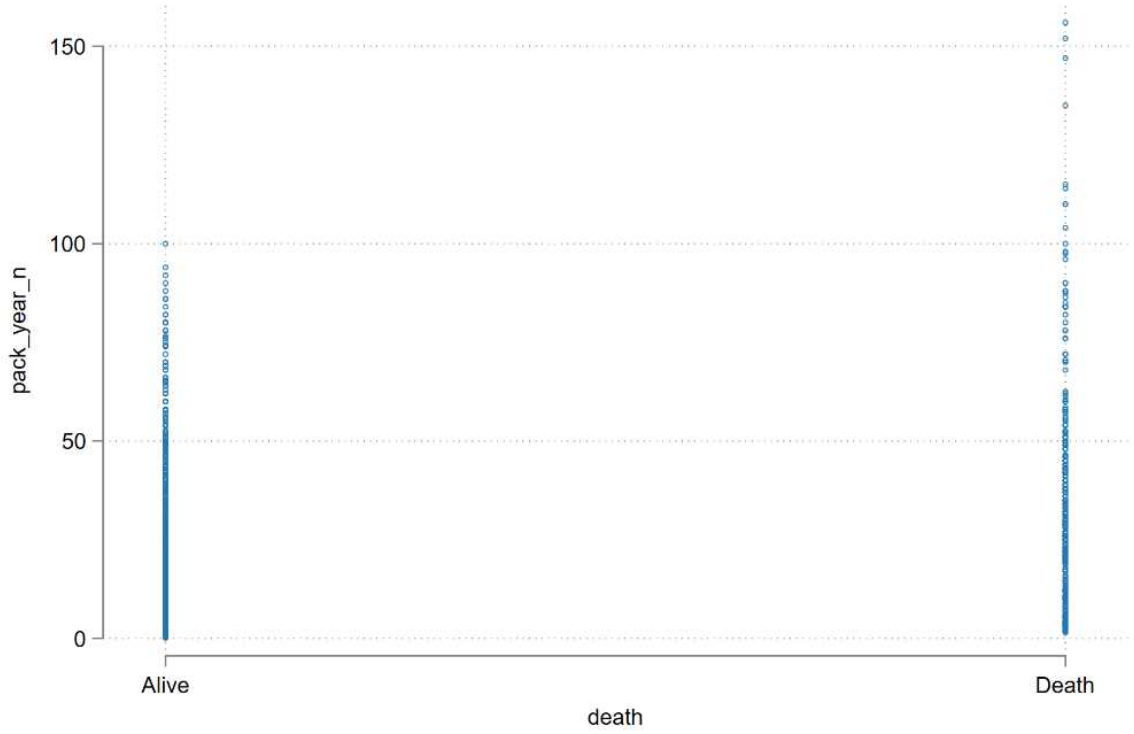


4.4.2 ポイントサイズを小さくしたドットプロット

ポイントサイズの調整は、**msize** オプションで実行します。

part2.do

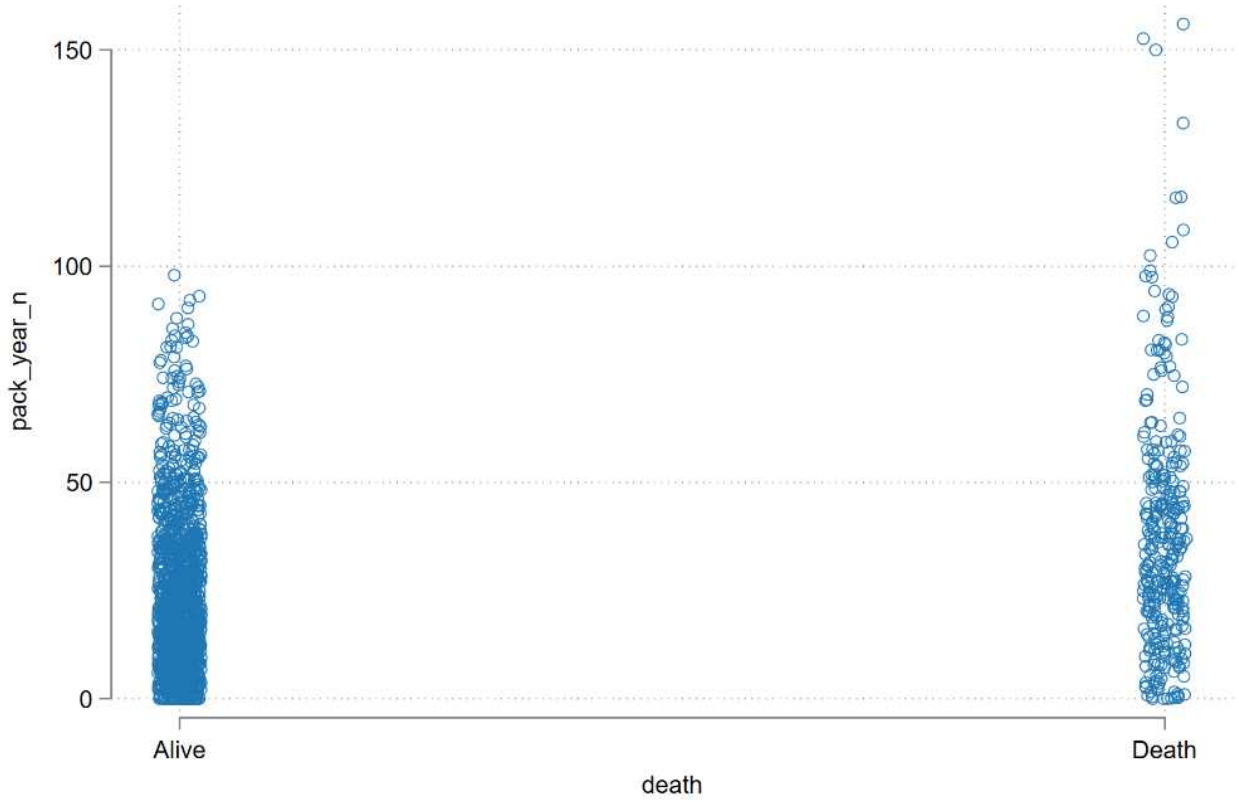
```
stripplot pack_year_n, sheme(white_tableau) over(death) vertical msize(0.5)
```



4.4.3 ポイントを少しずらしたジッタープロット

ジッタープロットを行なうためには、**jitter** オプションを用います。

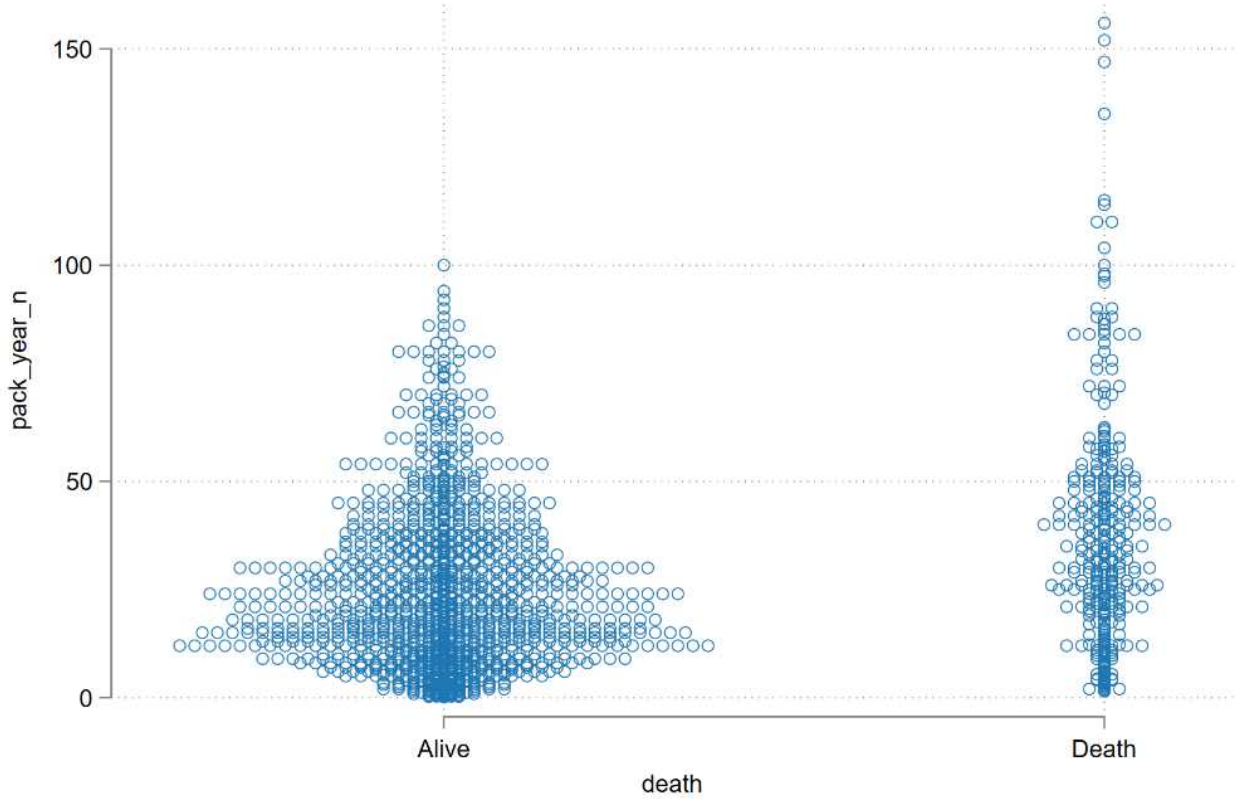
```
                                                                    part2.do
stripplot pack_year_n, ///
     sheme(white_tableau) over(death) vertical jitter(5 5) center
```



4.4.4 ビースウォーム（っぽい）プロット

R で実行されていたように、確率分布に基づいてずらした Beeswarm プロットはサポートされていません。が、オプションの組み合わせにより、それっぽいプロットを行なうことが出来ます。ここでは **stack** オプションを用いることにより、ビースウォームプロットに似せました。

```
                                                                    part2.do
stripplot pack_year_n, ///
     scheme(white_tableau) over(death) vertical center ///
     stack
```



4.5 レインクラウドプロット

Rain cloud プロットについて Stata で実行可能かどうか？について、2022 年 11 月 20 日に、Statalist で質問上がっていました。Nicolas J Cox 先生の回答では「It's a hybrid box plot, density trace and parallel coordinates (profile, slope) plot. You need **twoway** for that if it has not been written up as a single command before.」とのことであり、単一コマンドでの描画はサポートされていないようです。Lollipop plot と異なって、私も作成方法がすぐに思いつきませんでした。

なお、回答者の Nicolas J Cox 先生はここまでで使用した外部コマンド **missingplot** や **stripplot** の作成者です。

https://www.statalist.org/forums/forum/general-stata-discussion/general/1690233-raincloud-plot-in-stata

5 Nominal variable（名義変数）

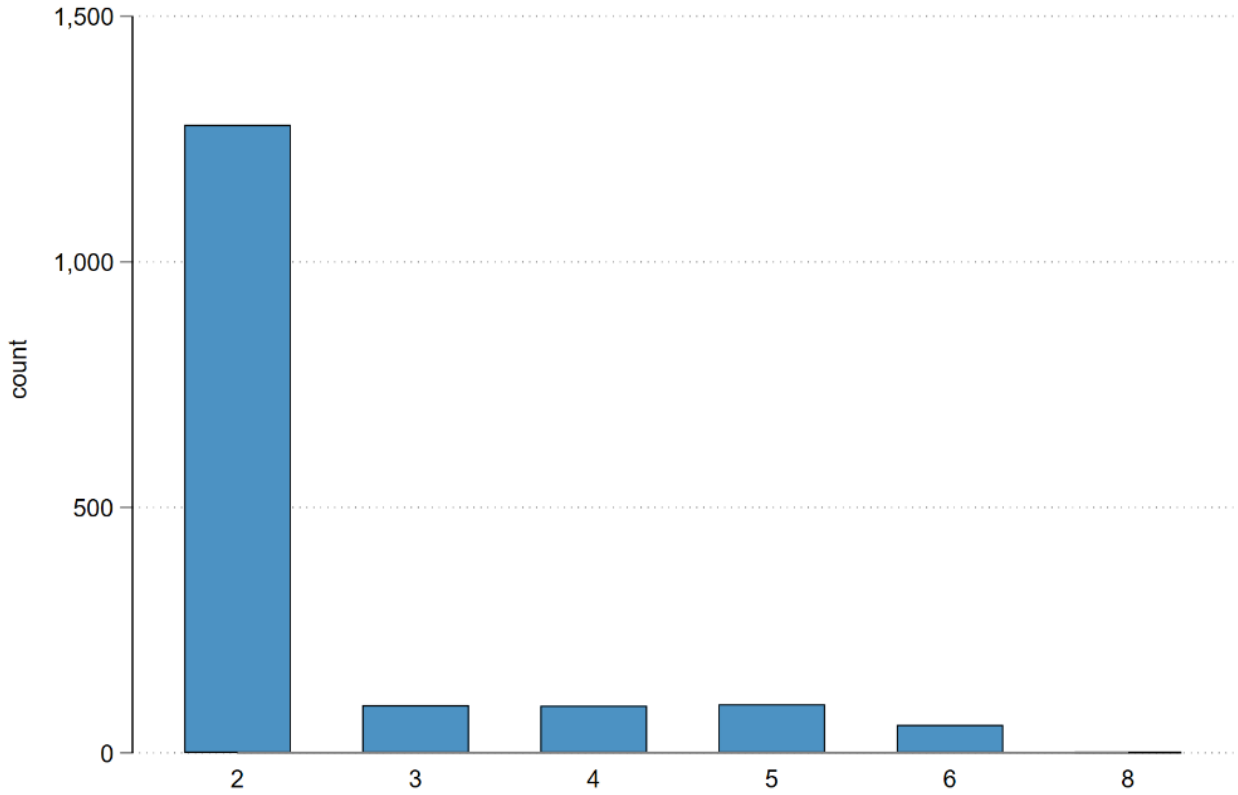
5.1 棒グラフ

5.1.1 ラベルなし

Stata では、**graph** コマンドのサブコマンド **bar** を用いて棒グラフを描画します。ここでは変数 **seqn** について数え上げ(**count**)をするように指定しています。

part2.do

```
graph bar (count) seqn, scheme(white_tableau) over(marital) ytitle("count")
```

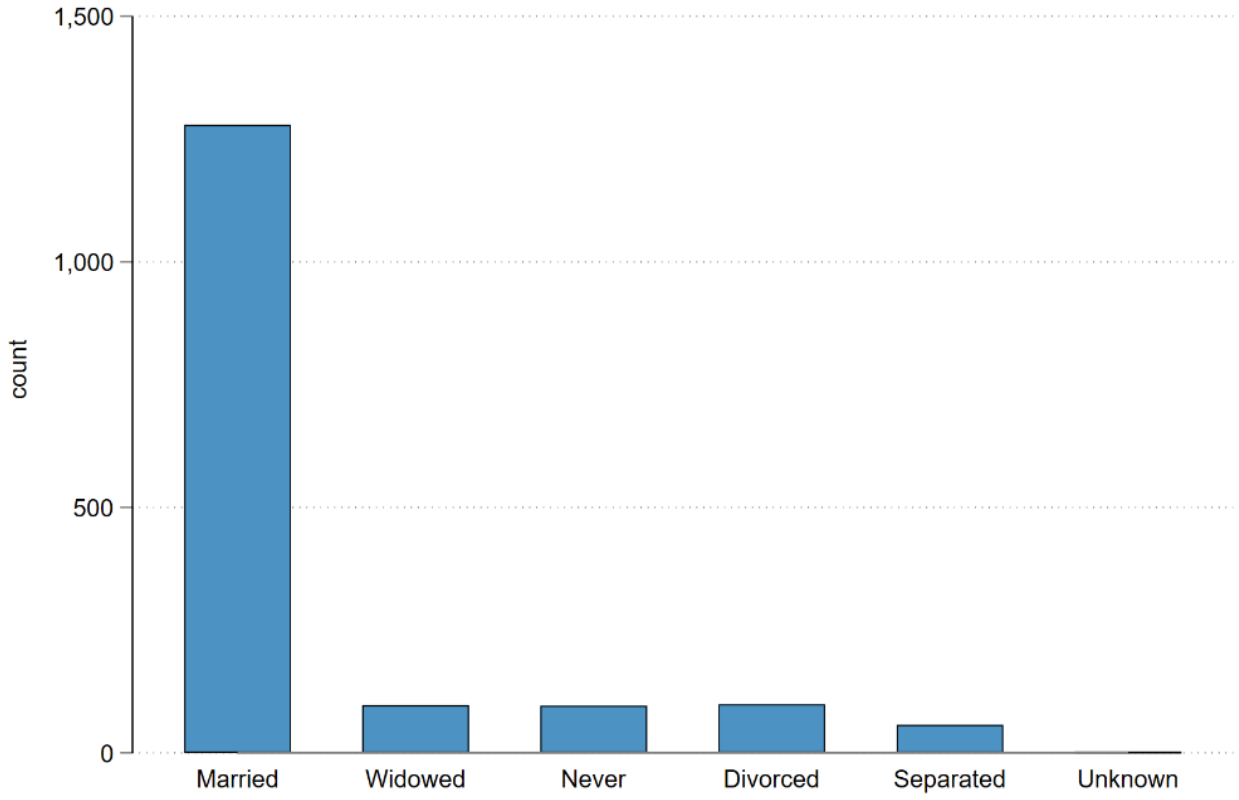


5.1.2 ラベルあり

ラベルありの棒グラフにするためには、変数に値ラベルをつける必要があります。

part2.do

```
label define marital /// 値ラベルの定義
      1 "Under17" 2 "Married" 3 "Widowed" ///
      4 "Never" 5 "Divorced" 6 "Separated" 8 "Unknown"
label values marital marital /// 値ラベルを変数に割り当て
graph bar (count) seqn, scheme(white_tableau) over(marital) ytitle("count")
```

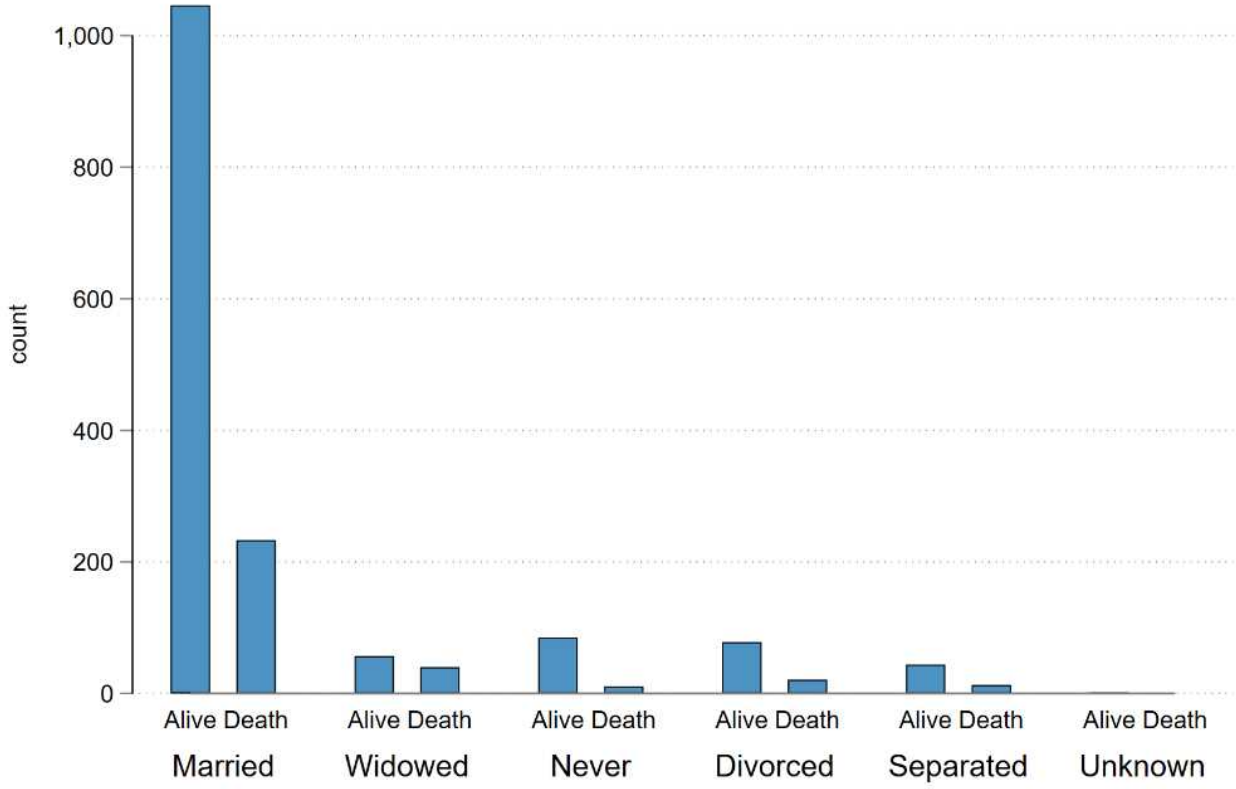


5.2 横並び棒グラフ

グラフに層別には **over** オプションを利用します。この **over** オプションは複数回利用することが可能です。このオプションの出現順をかえると、また異なるグラフになります。

part2.do

```
graph bar (count) seqn, ///
      scheme(white_tableau) ///
      over(death) over(marital) /// over オプションを重ねて利用
      ytitle("count")
```



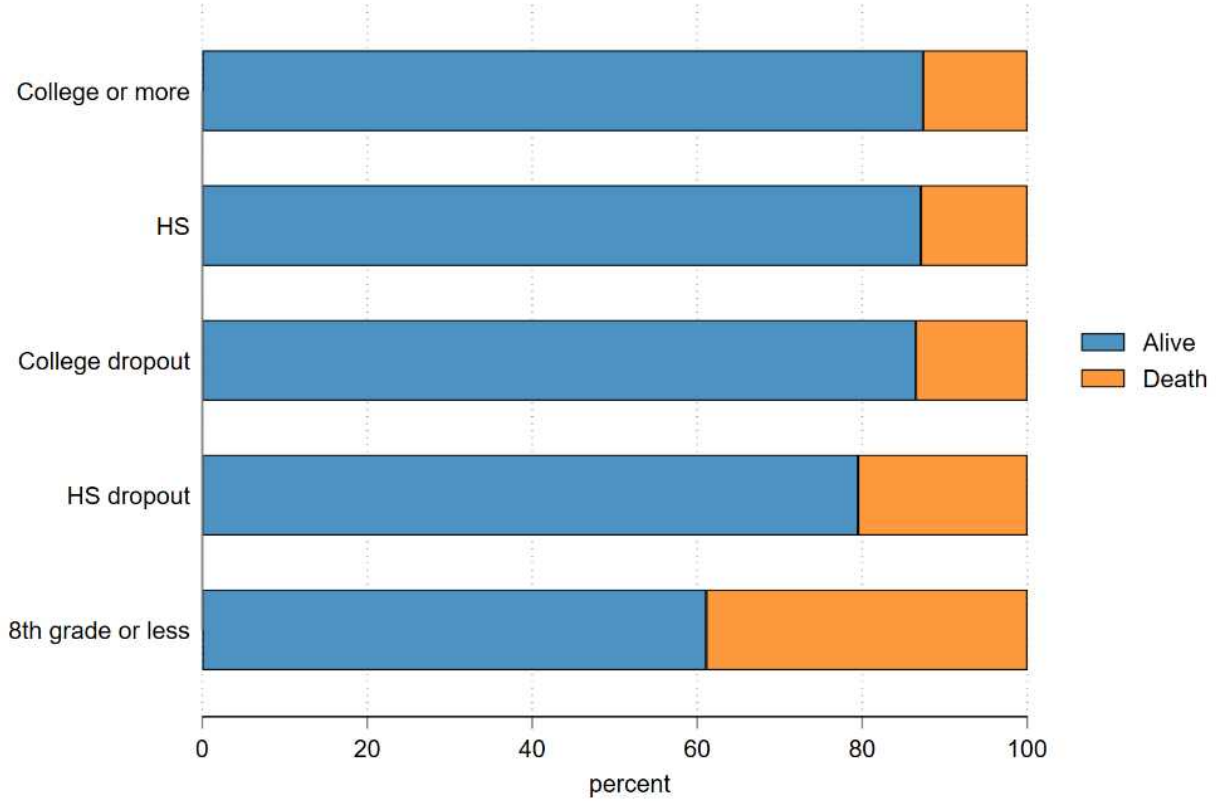
5.3 100%積み上げ棒グラフ

今回は、横向きの棒グラフですので、**graph** コマンドのサブコマンド **hbar** を用います。また、積み上げのためのオプションとして **stack** を用いています。積み上げる中身（死亡数と生存数）を予め算出した後にグラフ描画に入っていますので、やや手間がかかっています。

part2.do

```
label define education /// 値ラベルの定義
      5 "College or more" 4 "College dropout" 3 "HS" ///
      2 "HS dropout" 1 "8th grade or less"
label values education education // 値ラベルの割当て

bys education: egen frac_1 = sum(death) // 各教育における死亡数
bys education: gen frac_3 = _N // 各教育における人数
gen frac   = frac_1/frac_3 // 各教育における死亡割合
gen frac_2 = frac_3 - frac_1 // 各教育の生存数

graph hbar (mean) frac_2 frac_1, /// 生存数と死亡数で積み上げる
      scheme(white_tableau) ///
      over(education, sort(frac)) /// 各教育のバーをつくり、死亡割合でソートする
      stack /// 積み上げグラフにする
```

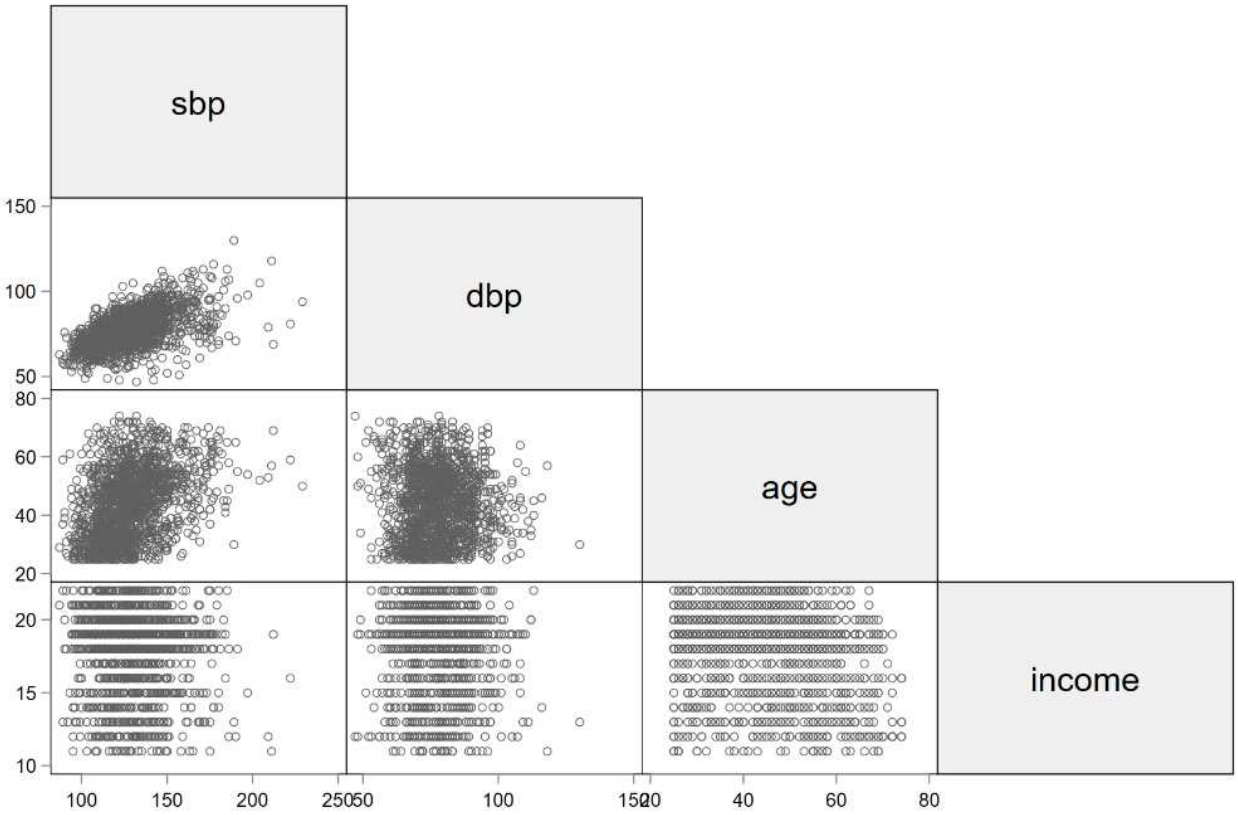
```
      percent /// 100%の積み上げにする。
      legend(label(1 "Alive") label(2 "Death"))
```



6 Multivariable（多変量）

Stata において、利用する変数同士の関係を網羅的に可視化するためのプロットとしては、**graph matrix** があります。ただし、これは散布図を描くのみですので、R の **ggpairs 関数**とは異なって変数型に合わせてグラフを調整するという機能はありません。下記では一部変数でのみを用いて作図しています。

```
graph matrix sbp dbp age income, ///
      scheme(white_tableau) ///
      half /// 散布図を片側だけにした
      msymbol(smcircle_hollow)
```


sbp
dbp
age
income
150
100
50
80
60
40
20
20
15
10
100
150
200
250
50
100
150
20
40
60
80