エビデンスベーストヘルスケア特講 I：量的データ解析の基本操作

中澤　港

2013 年 5 月 22 日

今回は，EZR を使って，量的データを解析する際に必要となる基本操作を演習する。簡単のため，演習を通して同じデータを利用する。使うデータフレームは，MASS ライブラリに組み込まれている leuk である。このデータフレームには，白血球数を示す変数 wbc，白血病の白血球の形態的な特徴の有無を示す変数 ag，生存時間（週数）を示す変数 time が含まれている。生存時間は急性骨髄性白血病で死亡した 33 人の患者について与えられている。白血球数は診断の時点でカウントされた値である。患者は診断時点でのアウエル小体 (Auer rods) 及び／または骨髄における白血球の顆粒化が存在すれば，ag の値が present，どちらも無ければ absent とされた（要因型の変数）。

データソースは，Feigl P, Zelen M (1965) Estimation of exponential survival probabilities with concomitant information. *Biometrics*, 21: 826-838. である。

基本操作として演習する主な内容は，分布の確認（図示，記述統計量の計算，正規性の検定），生存時間の中央値の計算，2 群間の生存時間の比較である。

1　図示

データの大局的性質を把握するには，図示をするのが便利である。人間の視覚的認識能力は，パターン認識に関してはコンピュータより遥かに優れていると言われているから，それを生かさない手はない。また，入力ミスをチェックする上でも有効である。

変数が表す尺度の種類によってさまざまな図示の方法があるが，量的データでは，ヒストグラム，ボックスプロット（箱ひげ図），ストリップチャート，散布図といったものが代表的である。

R-3.0.1 などで CRAN から EZR を導入した場合は，まず R を起動し，`library(Rcmdr)`** をしてから，「ツール」の「Rcmdr プラグインのロード」を選び，プラグインとして RcmdrPlugin.EZR を選んで OK ボタンをクリックし，Rcmdr を再起動すると EZR が利用可能になる。EZR のインストールパッケージでインストールした場合は，起動すると最初から EZR が使える状態になる。**
次いで，`MASS`** パッケージの** `leuk` **データフレームを読み込む。「ファイル」の「パッケージに含まれるデータを読み込む」を選び，表示されるウィンドウの左の枠で MASS をダブルクリックし，次に右の枠で leuk をダブルクリックし，[OK] ボタンをクリックする。**

このデータフレームを使って，いくつかのグラフを描いてみよう。

ヒストグラム　変数値を適当に区切って度数分布を求め，分布の様子を見る。R では `hist()` 関数を用いる。デフォルトでは「適当な」区切り方として “Sturges” というアルゴリズムが使われるが，明示的に区切りを与えることもできる。また，デフォルトでは区間が「〜を超えて〜以下」であり，日本で普通に用いられる「〜以上〜未満」ではないことにも注意されたい。「〜以上〜未満」にしたいときは，`right=FALSE` というオプションを付ければ良い。R コンソールで年齢 (`wbc`) のヒストグラムを描かせるには，`hist(leuk$wbc)` だが，「0 以上 20000 未満」から 20000 ごとに区切って「80000 以上 100000 未満」までのヒストグラムを描くように指定するには，`hist(leuk$wbc, breaks=0:5*20000, right=FALSE)` とする。

EZR では「グラフ」の「ヒストグラム」を選ぶ。leuk データでは，変数として `wbc` **を選べば，白血球数のヒストグラムが描ける。**

正規確率プロット　連続変数が正規分布しているかどうかを見るものである（正規分布に当てはまっていれば点が直線上に並ぶ）。R では `qqnorm()` 関数を用いる。例えば，leuk データフレームの白血球数 (`wbc`) について正規確率プロットを描くには，`qqnorm(leuk$wbc)` とする。

Rcmdr では「グラフ」の「QQ プロット」を選ぶ。leuk データで変数として wbc を選ぶと，まったく正規分布でないので直線状でないことがわかる。EZR では「オリジナルメニュー」の「グラフ」の「QQ プロット」を選べばいいのだが，CRAN から RcmdrPlugin.EZR をインストールした場合，これをロードするとオリジナルメニューの構造が崩れてしまうので選べない。現在のところ，Rcmdr で実行するか，コマンドを打つしかない（「グラフ」の「整列チャート」で近いものが描けるが横軸が異なる）。

ドットチャート　すべてのデータをドットとして，縦軸を数値軸としてプロットする。同じ値は横に並べる。カテゴリ変数によりグループ別にプロットする場合もある。R コンソールでは，stripchart() 関数の method="stack"オプションと vert=TRUE オプションにより実現できる。この例の場合は, stripchart(wbc~ag，data=leuk，vert=TRUE，method="stack") とする。

EZR では，「グラフ」から「ドットチャート」を選び，変数を指定するだけで描画できる。y 軸を対数軸にするオプションも指定できる。

幹葉表示 (stem and leaf plot)　大体の概数（整数区切りとか 5 の倍数とか 10 の倍数にすることが多い）を縦に並べて幹とし，それぞれの概数に相当する値の細かい部分を葉として横に並べて作成する図。R では stem() 関数を用いる。この場合なら stem(leuk$wbc) とする。

EZR では「グラフ」の「幹葉表示」を選び，変数として wbc を選ぶ。デフォルトでは外れ値を除外するところにチェックが付いているので，除外したくない場合はチェックを外してから OK ボタンをクリックする。

箱ヒゲ図 (box and whisker plot)　縦軸に変数値をとって，第 1 四分位を下に，第 3 四分位を上にした箱を書き，中央値の位置にも線を引いて，さらに第 1 四分位と第 3 四分位の差（四分位範囲）を 1.5 倍した線分をヒゲとして第 1 四分位の下と第 3 四分位の上に伸ばし，ヒゲの先より外れた値を外れ値として○をプロットした図を描くのが基本形である。上下のひげは，90 パーセンタイルと 10 パーセンタイルにしたり，95 パーセンタイルと 5 パーセンタイルにしたり，最大値と最小値にしたりというバリエーションがある（但し，原則として順序統計量を使う）。カテゴリによって層別した箱ヒゲ図を横に並べて「層別箱ひげ図」を描くと，大体の分布の様子と外れ値の様子が同時に比較できるので便利である。R では boxplot() 関数を用いる。leuk データで ag 別に白血球数 (wbc) の箱ヒゲ図を描くには，boxplot(wbc ~ ag，data=leuk) とする。

EZR では「グラフ」の「箱ひげ図」を選び，変数として wbc を選び（層別にする場合は右側の「群別する変数」で ag を選んで），上下のひげの位置として欲しいものをクリックしてから OK ボタンををクリックすればいい。EZR では「y 軸を対数軸に」の左側にチェックを入れることで，縦軸を対数軸にすることも可能である。見た目が違うだけだが，右裾が長い分布の場合は便利な機能である。似た用途のグラフとして，層別の平均とエラーバーを表示して折れ線で結ぶことも「グラフ」の「棒グラフ（平均値）」や「折れ線グラフ（平均値）」でできる。エラーバーとしては標準誤差，標準偏差，信頼区間から選択できる。

レーダーチャート　複数の連続変数を中心点から放射状に数直線としてとり，データ点をつないで表される図である。それら複数の変数によって特徴付けられる性質のバランスをみるのに役立つ。１つのケースについて１つのレーダーチャートができるので，他のケースと比較するには，並べて描画するか，重ね描きする。R では，plotrix ライブラリか fmsb ライブラリをインストールする必要がある。どちらも CRAN のミラーサイトからダウンロードしてインストールできる（install.packages("plotrix") と install.packages("fmsb")）。その上で, 例えば後者の場合なら, library(fmsb) としてから example(radarchart) とすれば使い方がわかる。**Rcmdr や EZR では描けない。**

散布図 (scatter plot)　２つの連続変数の関係を２次元の平面上の点として示した図である。R では plot() 関数を用いる。異なる群ごとに別々のプロットをしたい場合は plot() の pch オプションで塗り分けたり，points() 関数を使って重ね打ちしたりできる。点ごとに異なる情報を示したい場合は symbols() 関数を用いることができるし，複数の連続変数間の関係を調べるために，重ね描きしたい場合は matplot() 関数と matpoints() 関数を，別々のグラフとして並べて同時に示したい場合は pairs() 関数を用いることができる。データ点に文字列を付記したい場合は text() 関数が使えるし，マウスで選んだデータ点にだけ文字列を付記したい場合は identify() 関数が使える。leuk データで，R コンソールを使って，横軸に白血球数 (wbc)，縦軸に生存時間 (time) をとってプロットしたいときは，plot(time ~ wbc，data=leuk) と打てば良い。

EZR では「グラフ」の「散布図」で描ける。いろいろなオプションがある。

生存曲線 (survival curve) 生存時間解析では，観察期間内にすべてのイベントが発生するとは限らないので，観察打ち切りを正しく扱って生存曲線を描画する必要がある。`leuk` データには打ち切りが無い（すべての人について死亡イベントが発生している）ので，観察打ち切りかどうかを示す変数（打ち切りが 0，イベント発生が 1）を作らねばならない。この場合，コンソールでは `leuk$flag <- rep(1,33)` と打っておき，`library(survival)` と打って，`plot(survfit(Surv(time, flag)~1, data=leuk))` とすれば 95% 信頼区間付きで，生存曲線が描ける。

> **EZR では，「アクティブデータセット」の「変数の操作」の「計算式を入力して新たな変数を作成」を選び，変数名として `Flag` などとし，式としては 1 と入力して OK ボタンをクリックする。「グラフ」から「他の因子で調整した生存曲線の表示」を選び，「観察期間の変数」として `time`，「イベント (1)，打ち切り (0) の変数」として `Flag` を選び，「調整に用いる変数」も `Flag` にして OK ボタンをクリックする（ここでは Flag の値はすべて 1 なので，調整は行われない）。ここで，例えば `wbc` を指定すると，Cox 回帰で白血球数の影響を調整した生存曲線が描かれる。**

2 記述統計と分布の正規性の検定，外れ値の検定

以前説明したように，平均値，中央値，標準偏差，四分位範囲などの記述統計量を計算するには，EZR では，「統計解析」の「連続変数の解析」の「連続変数の要約」を選ぶだけでいい。

分布の正規性の検定は，EZR では，「統計解析」の「連続変数の解析」の「正規性の検定」を選び，変数として `wbc` を選んで OK ボタンをクリックするだけでいい。自動的にコルモゴロフ＝スミルノフ検定とシャピロ＝ウィルク検定の結果が表示される。この例では結果が異なるが，これらの検定は分布の正規性へのアプローチが異なるので，結果が一致しないこともある。多くの検定手法がデータの分布の正規性を仮定しているが，この検定で正規性が棄却されたからといって機械的に変数変換やノンパラメトリック検定でなくてはいけないとは限らない。独立 2 標本の平均値の差がないという仮説を検定するための t 検定は，かなり頑健な手法なので，正規分布に従っているといえなくても，そのまま実行してもいい場合も多い。

外れ値の検定は，EZR では，「統計解析」の「連続変数の解析」の「外れ値の検定」を選んで，変数として `wbc` を指定して OK ボタンをクリックするだけでいい。外れ値を NA で置き換えた新しい変数を作成することも右のオプション指定から容易にできる。しかし，以前にも説明したが，この結果だけで機械的に外れ値を除外することはお薦めできない。また，この例では “No outliers were identified.” と表示されるので，統計学的に外れ値があるとはいえない。

3 生存時間解析

生存時間解析は Rcmdr 本体には入っていない。しかし，プラグインが 2 種類発表されている。1 つは John Fox 教授自身が開発した RcmdrPlugin.survival であり，もう 1 つは Dr. Daniel C. Leucuta というルーマニアの研究者が開発した RcmdrPlugin.SurvivalT である[*1]。いずれも，survival ライブラリの機能の基本的なものに対して，グラフィカルなユーザーインターフェースを提供する。

EZR では，「統計解析」の「生存期間の解析」からログランク検定による 2 群の生存時間の比較や，コックス回帰を選ぶことができる。今回はカプラン＝マイヤ法による生存時間の中央値の推定と，ログランク検定のみ説明する。

なお，EZR でサポートされていないような，より高度な生存時間解析をするには，`survival` パッケージなどを使ってコマンドを打つことになるが，通常の解析ではそこまでする必要が生じることは稀であろう。

3.1 生存時間解析とは

実験においては，化学物質などへの 1 回の曝露の影響を時間を追ってみていくことが良く行われる。時間ごとに何らかの量の変化を追うほかに，エンドポイントを死亡とした場合，死ぬまでの時間を分析することで毒性の強さを評価することができる。このような期間データを扱う方法としては，一般に生存時間解析 (Survival Analysis または Event History Analysis) と呼ばれるものがある。なかでもよく知られているものが Kaplan-Meier の積・極限推定量である（現在では一般に，カプラン＝マイヤ推定量と呼ばれている）。カプラン＝マイヤ推定量は，イベントが起こった各時点での，イベントが起こる可能性がある人口（リスク集合）あたりのイベント発生数を 1 から引いたものを掛け合わせて得られる，ノンパラメトリックな最尤推定量である。また，複数の期間データ列があったときに，それらの差を検定したい場合は，ログランク検定や一般化ウィルコクソン検定が使われる。

それらのノンパラメトリックな方法とは別に，イベントが起こるまでの時間が何らかのパラメトリックな分布に当てはまるかどうかを調べる方法もある。当てはめる分布としては指数分布やワイブル分布がある。中間的なものとして，イベントが起こるまで

[*1] このプラグインについての説明が，Leucuta DC, Achimas-Cadariu A (2008) Statistical graphical user interface plug-in for survival analysis in R statistical and graphics language and environment. *Applied Medical Informatics*, 23(3-4): 57-62. という論文として発表されている。

の期間に対して，何らかの別の要因群が与える効果を調べたいときに，それらが基準となる個体のハザードに対して $\exp(\sum \beta z_i)$ という比例定数の形で掛かるとする比例ハザード性だけを仮定し，分布の形は仮定しないコックス回帰（比例ハザードモデルとも呼ばれる）は，セミパラメトリックな方法といえる。別の要因群の効果は，パラメトリックなモデルに対数線形モデルの独立変数項として入れてしまう加速モデルによって調べることもできる。

R では生存時間解析をするための関数は survival パッケージで提供されており，R コンソールで library(survival) または require(survival) とタイプして survival パッケージをメモリにロードした後では，Surv() で生存時間クラスをもつオブジェクトの生成，survfit() でカプラン＝マイヤ法，survdiff() でログランク検定，coxph() でコックス回帰，survreg() で加速モデルの当てはめを実行できる。なお，生存時間解析について，より詳しく知りたい方は，大橋，浜田 (1995) などを参照されたい。

3.2 カプラン＝マイヤ法

まず，カプラン＝マイヤ推定量についての一般論を示す。イベントが起こる可能性がある状態になってから，イベントが起こった時点を $t_1, t_2, ...$ とし，t_1 時点でのイベント発生数を d_1，t_2 時点でのイベント発生数を d_2，以下同様であるとする。また，時点 $t_1, t_2, ...$ の直前でのリスク集合の大きさを $n_1, n_2, ...$ で示す。リスク集合の大きさとは，その直前でまだイベントが起きていない個体数である。観察途中で死亡や転居などによって打ち切りが生じるために，リスク集合の大きさはイベント発生によってだけではなく，打ち切りによっても減少する。従って n_i は，時点 t_i より前にイベント発生または打ち切りを起こした個体数を n_1 から除いた残りの数となる。なお，イベント発生と打ち切りが同時点で起きている場合は，打ち切りをイベント発生直後に起きたと見なして処理するのが慣例である。このとき，カプラン＝マイヤ推定量 $\hat{S}(t)$ は，

$$\hat{S}(t) = (1 - d_1/n_1)(1 - d_2/n_2)... = \prod_{i<t}(1 - d_i/n_i)$$

として得られる。その標準誤差はグリーンウッドの公式により，

$$var(\hat{S}) = \hat{S}^2 \times \sum_{i<t} \frac{d_i}{n_i(n_i - d_i)}$$

で得られる。通常，階段状の生存曲線のプロットも同時に行う。

R コンソールでは，library(survival) としてパッケージを呼び出し，dat <- Surv(生存時間，打ち切りフラグ) 関数で生存時間データを作り（打ち切りフラグは 1 でイベント発生，0 が打ち切り。ただし区間打ち切りの場合は 2 とか 3 も使う），res <- survfit(dat~1) でカプラン＝マイヤ法によるメディアン生存時間が得られ（群分け変数 C によって群ごとにカプラン＝マイヤ推定をしたい場合は，res <- survfit(dat~C) とすればよい），plot(res) とすれば階段状の生存曲線が描かれる。イベント発生時点ごとの値を見るには，summary(res) とすればよい[*2]。

既に生存曲線の描画の際に，plot() 関数に与える中身として survfit() 関数を用いた。survift() 関数の結果は，plot() 関数に与えれば生存曲線を描いてくれるし，print() 関数に与えれば（関数をそのままコンソールに打てば，デフォルトで print() が呼び出されるのが普通なので，print() で明示的に括らなくてもよい），生存時間の中央値と 95% 信頼区間の計算結果が表示される。

EZR では，「統計解析」の「生存期間の解析」の「生存時間の記述と群間の比較（Logrank 検定）」を選び，観察期間の変数として time を選び，イベント (1)，打ち切り (0) の変数として Flag を選んで，OK ボタンをクリックする。
生存曲線のグラフも描画されるが，出力ウィンドウに，すべてのイベント発生時点における生残率とその標準誤差，95% 信頼区間が表示された後（コンソールでは survfit() の結果を summary() に与えると表示されるもの），最後に生存時間の中央値と 95% 信頼区間がシンプルに表示される。

```
> summary.km(survfit=km)
  records median.median 95% CI
1      33            22   5-43
```

[*2] 参考までに書いておくと，生データがイベント発生の日付を示している場合，間隔を計算するには difftime() 関数や ISOdate() 関数を使う。例えば 1964 年 8 月 21 日生まれの人の今日の年齢は integer(difftime(ISOdate(2007,6,13),ISOdate(1964,8,21)))/365.24 とすれば得られるし，さらに 12 を掛ければ月単位になる。

練習問題

survival ライブラリに含まれているデータ aml は，急性骨髄性白血病 (acute myelogenous leukemia) 患者が化学療法によって寛解した後，ランダムに 2 群に分けられ，1 群は維持化学療法を受け（維持群），もう 1 群は維持化学療法を受けずに（非維持群），経過観察を続けて，維持化学療法が再発までの時間を延ばすかどうかを調べたデータである[a]。以下の３つの変数が含まれている。

time　生存時間あるいは観察打ち切りまでの時間（週）

status　打ち切り情報（0 が観察打ち切り，1 がイベント発生）

x　維持化学療法が行われたかどうか（Maintained が維持群，Nonmaintained が非維持群）

薬物維持化学療法の維持群と非維持群で別々に，再発までの時間の中央値をカプラン＝マイヤ推定し，生存曲線をプロットせよ。

[a] 出典: Miller RG: Survival Analysis. John Wiley and Sons, 1981. 元々は，Embury SH, Elias L, Heller PH, Hood CE, Greenberg PL, Schrier SL: Remission maintenance therapy in acute myelogenous leukaemia. *Western Journal of Medicine*, 126, 267-272, 1977. のデータ。研究デザインは Gehan と似ている。

R コンソールでは以下を入力する。

```
library(survival)
print(res <- survfit(Surv(time,status)~x, data=aml))
plot(res,xlab="(Weeks)",lty=1:2,main="Periods until remission of acute myelogenous leukaemia")
legend("right",lty=1:2,legend=levels(aml$x))
```

2 行目でカプラン＝マイヤ法での計算がなされ，下枠内が表示される。

```
               records n.max n.start events median 0.95LCL 0.95UCL
x=Maintained          11    11      11      7     31      18      NA
x=Nonmaintained       12    12      12     11     23       8      NA
```

この表は，維持群が 11 人，非維持群が 12 人，そのうち再発が観察された人がそれぞれ 7 人と 11 人いて，維持群の再発までの時間の中央値が 31 週，非維持群の再発までの時間の中央値が 23 週で，95% 信頼区間の下限はそれぞれ 18 週と 8 週，上限はどちらも無限大であると読む。

3 行目で 2 群別々の再発していない人の割合の変化が生存曲線として描かれ，4 行目で凡例が右端に描かれる。

EZR では，まず survival パッケージ内の aml をアクティブにする。「ファイル」の「パッケージに含まれるデータを読み込む」で，パッケージとして survival をダブルクリックし，次いでデータフレームとして leukemia をダブルクリックしてから OK ボタンをクリックする。

カプラン＝マイヤ推定は，「統計解析」の「生存期間の解析」の「生存時間の記述と群間の比較（Logrank 検定）」を選び，観察期間の変数として time を選び，イベント (1)，打ち切り (0) の変数として status を選び，群分け変数として x を選んで OK ボタンをクリックするだけで完了する。以下のように，次に説明するログランク検定の検定結果まで表示される。なお，中央値の 95% 信頼区間がコンソールで survfit() 関数を使って求めた値と若干異なっているが，これは標準誤差の計算式の違いによる。

```
n median medianCI p.value
x=Maintained    11     31    13-NA  0.0653
x=Nonmaintained 12     23     5-33
```

3.3　ログランク検定

2 群の生存曲線に差が無いという帰無仮説を検定するための代表的な方法が，ログランク検定である。考え方を簡単な例で説明する。

8 匹のラットを 4 匹ずつ 2 群に分け，第 1 群には毒物 A を投与し，第 2 群には毒物 B を投与して，生存期間を追跡したときに，第 1 群のラットが 4,6,8,9 日目に死亡し，第 2 群のラットが 5,7,12,14 日目に死亡したとする。この場合，観察期間内にすべてのラットが死亡し，正確な生存時間がわかっているので，観察打ち切りがないデータとなっていて計算しやすい。

ログランク検定の思想は，大雑把に言えば，死亡イベントが起こったすべての時点で，群と生存／死亡個体数の 2 × 2 クロス集計表を作り，それをコクラン＝マンテル＝ヘンツェル流のやり方で併合するということである。上記の例では，死亡イベントが起こった時点 1〜8 において各群の期待死亡数を計算し，各群の実際の死亡数との差をとって，それに時点の重みを掛けたものを，各時点における各群のスコアとして，群ごとのスコアの合計を求める。２群しかないので，各時点において群１と群２のスコアの

絶対値は同じで符号が反対になる。２群の生存時間に差がないという帰無仮説を検定するためには，群１のスコアの２乗を分散で割った値をカイ二乗統計量とし，帰無仮説の下でこれが自由度１のカイ二乗分布に従うことを使って検定する。なお，重みについては，ログランク検定ではすべて 1 である。一般化ウィルコクソン検定では，重みを，2 群を合わせたリスク集合の大きさとする（そうした場合，もし打ち切りがなければ，検定結果は，ウィルコクソンの順位和検定の結果と一致する）。つまり，**ログランク検定でも一般化ウィルコクソン検定でも，実は期間の情報はまったく使われず，死亡順位の情報だけが使われている。**

記号で書けば次の通りである。第 i 時点の第 j 群の期待死亡数 e_{ij} は，時点 i における死亡数の合計を d_i，時点 i における j 群のリスク集合の大きさを n_{ij}，時点 i における全体のリスク集合の大きさを n_i とすると，

$$e_{ij} = d_i \cdot n_{ij}/n_i$$

と表される。上の例では，$e_{11} = 1 \cdot n_{11}/n_1 = 4/8 = 0.5$ となる。時点 i における第 j 群の死亡数を d_{ij}，時点の重みを w_i と表せば，時点 i における群 j のスコア u_{ij} は，

$$u_{ij} = w_i(d_{ij} - e_{ij})$$

となり，ログランク検定の場合（以下，重みは省略してログランク検定の場合のみ示す）の群 1 の合計スコアは

$$u_1 = \sum_i d_{i1} - e_{i1}$$

となる。上の例では，

$$u_1 = (1 - 4/8) + (0 - 3/7) + (1 - 3/6) + (0 - 2/5) + (1 - 2/4) + (1 - 1/3) + (0 - 0/2) + (0 - 0/1)$$

である。これを計算すると約 1.338 となる。分散は，分散共分散行列の対角成分を考えればいいので，

$$V = V_{jj} = \sum_i \frac{(n_i - n_{ij})n_{ij}d_i(n_i - d_i)}{{n_i}^2(n_i - 1)}$$

となる。この例の数値を当てはめると，

$$V = \frac{(8-4)\times 4}{8^2} + \frac{(7-3)\times 3}{7^2} + \frac{(6-3)\times 3}{6^2} + \frac{(5-2)\times 2}{5^2} + \frac{(4-2)\times 2}{4^2} + \frac{(3-1)\times 1}{3^2}$$

となり，計算すると，約 1.568 となる。したがって，$\chi^2 = 1.338^2/1.568 = 1.14$ となり，この値は自由度 1 のカイ二乗分布の 95% 点である 3.84 よりずっと小さいので，有意水準 5% で帰無仮説は棄却されない。つまりこれだけのデータでは，差があるとはいえないことになる（もちろん，サンプルサイズを大きくすれば違う結果になる可能性もある）。

R では，`Surv(time,event)` と `group`（注：ここで time は生存時間，event は 1 がイベント観察，0 が観察打ち切りを示すフラグ，group がグループを示す）を，`survdiff()` 関数に与えることによってログランク検定が実行できる。打ち切りレコードがない場合は，event は省略できる。なお，生存時間解析の関数はすべて `survival` パッケージに入っているので，まず `library(survival)` とすることは必須である。

この例では，R コンソールでは以下を入力する。

```
library(survival)
time <- c(4,6,8,9,5,7,12,14)
event <- c(1,1,1,1,1,1,1,1)
group <- c(1,1,1,1,2,2,2,2)
dat <- Surv(time,event)
survfit(dat~group)
survdiff(dat~group)
```

得られる結果の一番下を見ると，$\chi^2 = 1.2$，自由度 1，$p = 0.268$ となっているので，有意水準 5% で，2 群には差がないことがわかる。

カプラン＝マイヤ法の練習問題で使った aml データで，維持群と非維持群の生存時間に差が無いという帰無仮説をログランク検定するには，R コンソールでは以下のように打つ（EZR での実行方法は既に示した通り）。

```
library(survival)
survdiff(Surv(time,status)~x, data=aml)
```

表示される結果は EZR での計算結果と同じなので省略する。p 値が 0.0653 なので，有意水準 5% で統計学的に有意な差があるとはいえない。**`MASS` パッケージの `leuk` データで，`ag` の有無によって生存時間に差があるかどうかも試されたい。**