Baayen, R. H. (2008). *Analyzing Linguistic Data: A practical introduction to statistics using R.* Cambridge: Cambridge University Press.

金田　拓

2 Graphical data exploration

✓ 2.2 Visualizing single random variables　　pp. 24~27　　(in pp. 21~32)

Windows, Mac OS X を使用している場合、「コピー」や、「別名で保存」によって保存することもできるが、作画関数（作画デバイス）を使って、jpeg や pdf ファイルに結果を出力することもできる。ファイルはディレクトリで指定した階層に保存される。

作画関数を使用した場合、コマンドの結果は R の画面ではなく、ファイルに書き込まれることとなる。この効果は dev.off( )関数でファイルが閉じられるまで続く。

例１： 400×420 ピクセルの jpeg ファイル、barplot.jpeg を、無ければ作成して開くコマンド。

```
> jpeg(“barplot.jpeg”, width = 400, height = 420)
> truehist(ratings$Frequency, xlab = “log word frequency”)
> dev.off( )
```

2 行目のコマンドは、jpeg( )関数が使われているので、出力結果は R の画面には出力されず、1 行目で開いた barplot.jpeg に書き込まれることとなる。

3 行目は、現行のデバイスをオフにする機能。閉じないと延々結果が書き込まれ続けることとなる。必要な処理が終わったら、すぐに閉じることを推奨。

例２：postscript( )関数で、Adobe PostScript ファイル、barplot.ps を、無ければ作成して開くコマンド

```
> postscript(“barplot.ps”, horizontal = FALSE, height = 6, width = 6, family = “Helvetica”, paper =
+ “special”, onefile = FALSE)
> truehist(ratings[1]$Frequency, xlab = “log word frequency”)
> dev.off( )
```

Adobe Postscript に出力される作画デバイス描画が縦か横かは horizontal の option で決定される。TRUE なら横、FALSE なら縦。何も入れないと横になる。

height, width はそれぞれ plot の高さ、幅を指定する。単位はインチ（約 2.5cm）

family でフォント、paper で保存する postscript の形式を指定する。onefile=FALSE は結果を単一のページにまとめるか、別にするかを指定する。

[1] テキスト p. 25 では **items** となっているが、items を入力すると、Frequency という列は存在しないためエラーとなる。**ratings の誤植と推測される**。

✓ ヒストグラムには、左端の端点の位置によって、形が変わってしまうという欠点がある。

例：1.5 という項目があったとして、目盛り 1.0 のヒストグラムにこの値を入れるとする。左の端点が 0.0 の場合、これは左から 2 列目の 1.0~2.0 の値に含まれるが、左の端点が 0.7 だった場合、この値は左から 1 列目の 0.7~1.7 に含まれ、2 列目 1.7~2.7 には含まれない。

➢ hist( )関数の初期設定では区切り幅は適当に選択される。そのため、より適切なヒストグラムを描くためには操作が必要となる。これまでのパートで用いた truehist( )関数は、Scott(1992)[2]が提唱した方法を用いて、恣意的な作図にならないよう調整が施されている。

➢ 歪みを防ぎ、なるべく自然な状態の変数のヒストグラムを作る方法として、density( )関数により密度推定を行い、密度曲線(density curve)を描くことができる（反応時間は連続変数なので、目盛りを細かくすれば密度推定の分布に近づいていくと予想される）。

1. p. 26 の Figure 2.2 のように描画するためには、軸の幅を適切に指定する必要がある。まず、plot = FALSE の option を使って「描画はせずに」ヒストグラムを変数 h に格納する。

```
> h = hist(lexdec$RT, freq = FALSE, plot = FALSE)
```

2. これで必要な高さと幅を、それぞれ、h$density と h$breaks として取り出すことが可能となった。次に、密度曲線を変数 d に格納する

```
> d = density(lexdec$RT)
```

3. x 軸、y 軸の幅を指定する必要があるので、range( )関数を用いて最少・最大値を代入する

```
> xlimit = range(h$breaks, d$x)
> ylimit = range(0, h$density, d$y)
```

4. これで hist( )関数を使って描画する準備が整ったので、以下の式で出力が可能。

```
> hist(lexdec$RT, freq = FALSE, xlim = xlimit, ylim = ylimit, main = "", xlab = "log RT", ylab = "",
+ col = "lightgrey", border = "darkgrey", breaks = seq(5.8, 7.6, by = 0.1))
```

breaks は truehist( )関数だと自動でやってくれる機能だが、hist( )関数で行うこの場合、自分でデータの幅、つまり列をいくつ刻みにするかを指定しなくてはならない。この場合、5.8 から 7.6 までを、0.1 刻みで描くという指定を行っている。

[2] Scott, D. W. (1992). *Multivariate Density Estimation: Theory, Practice, and Visualization*. New York: Wiley)

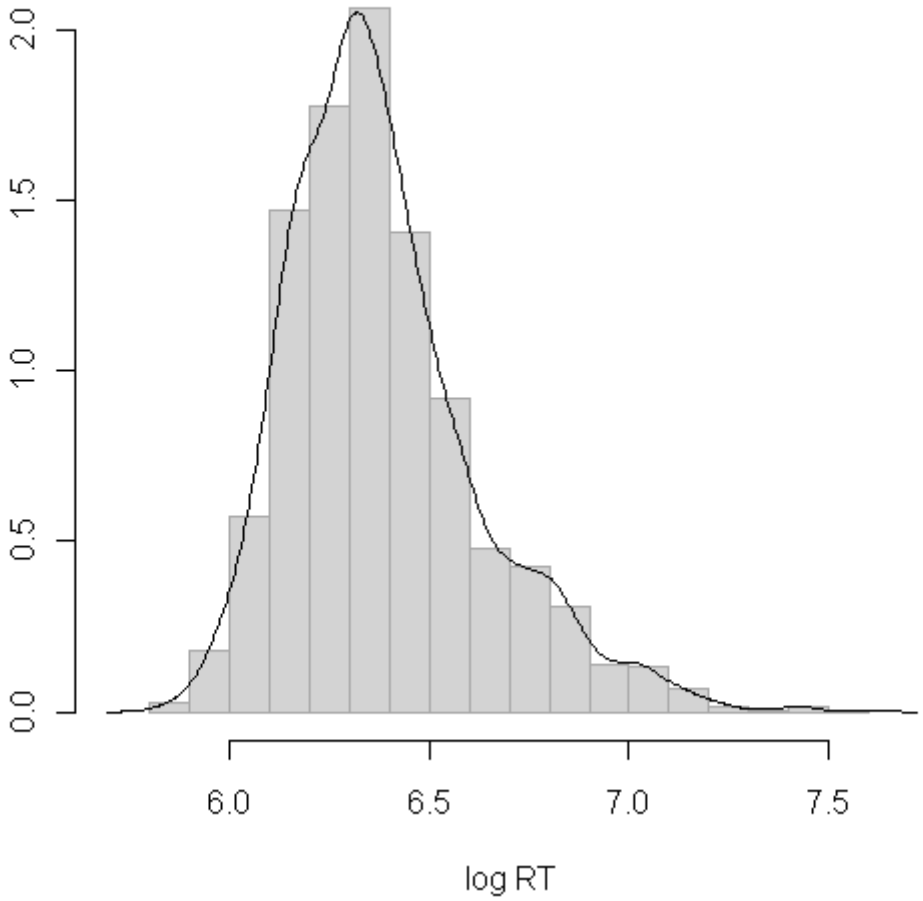
5.　x と y の座標ベクトルを lines に入力することで、密度推定の曲線を引くことができる

```
lines(d$x, d$y)
```

実は上↑の式は、density を格納した d に座標情報を伝える機能があるため、

```
> lines(d)
```

と入力するだけで同じ結果が得られる



他、plot ( )関数を用いることで、ヒストグラムや密度を簡単に描画することができる

```
> plot(h)
```

で変数 h、つまり先ほど格納したヒストグラム

```
> plot(d)
```

で変数 d、つまり先ほど格納した密度曲線を描くことができる。