９日目：項目のチェック（３）

今日は，度数分布表を作ったり，ヒストグラムを書いたりしてみましょう。なお，今回は項目のチェックが目的なので，あまり込みいった度数分布表などは作りませんので。あくまでも，ある項目で，１の回答が○名，２が△名…といったことを把握するレベルです。

度数分布表で最も簡便なのは table でしょう。

```
table(x[5])
table(x$age)
```

これだけの指示で簡単な度数分布表を作ってくれます。しかし，問題なのは欠損値をまったく無視してしまうところです。たとえば，b2 には欠損値がありますが，table(x$b2)で実行しても下図のように表示され，「NA」が含まれるのかどうかまったくわかりません。これは項目のチェックにはとてもリスキーです。そのため， exclude=NULL を必ず入れておいきます。

```
table(x$b2, exclude=NULL)
```

これであればうまくいきます。なお，table(x$age)の結果にカテゴリとしての１や２がなく，またそれらで０と表示されていないように，「NA」が含まれない場合は<NA>というカテゴリ自体が表示されません。

```
> table(x$b2)

3 4 5
7 6 6
>
> table(x$b2, exclude=NULL)

   3    4    5 <NA>
   7    6    6    1
```

欲張れば，prettyR などのパッケージを使って，%も合わせて算出するなどということもできます。個人的には，Ｒで計算をさせるのは，上の table 出力までで十分ではないかと感じています。項目分析の結果は，Ｒでの結果をざっとながめて終わりというものではなく，以後の分析過程でしばしば参照するようなものなので，実際の作業過程ではエクセルなどにコピペしてまとめておくことになります。そのため，どこまでＲの出力に求めるかは各自で決めてください。

では次に，ヒストグラムを作成してみます。繰り返しますが，ここでは項目チェックに使える簡単なものを目指します。

```
hist(x$age)
```

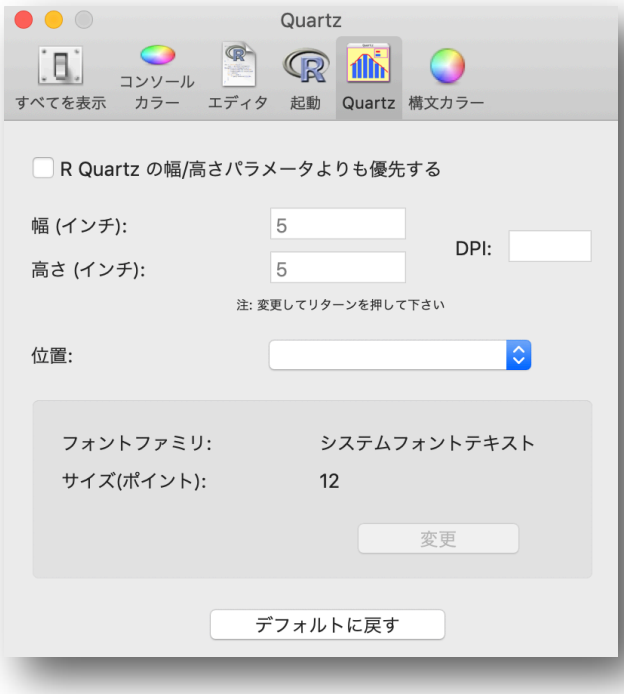
これだけを，実行してみてください。

これでどういう表示になるのか確認してください。おそらく，Quartz 2 という名前のウインドに出力されると思います。

このように，R で作図をしようとすると図用の新しいウインドが開きます。ただ，場合によっては，図が見切れているかもしれません。このウインドはスクロールしないのですが，他の Mac 用アプリケーションと同じで，画面の端や角にカソールをもっていき，ドラッグすればサイズを自由に変更できます。サイズを変更すると図も合わせて修正されますが，その際に見切れが修正される場合が多いようです。また，以下のようにして，自分の利用環境に合ったウインドのサイズ指定をしておくこともできます。

R のメニューバーから「環境設定」を選んでください。そしてアイコンの Quartz を選びます。すると右のような画面になります。

ここで，「R Quartz の幅/高さパラメータよりも優先する」にチェックを入れ，幅と高さを指定してやります。それぞれの環境で違うでしょうから，いくつか試してみてください。



さて，この資料では変数名をアルファベットでと指示していますが，日本語を使った場合には，Quartz 画面で「x$□□」などと文字化けします。デフォルトのフォント設定のままだと，アルファベットや数字は問題なく表示してくれますが，日本語はおかしくなるので，利用するフォントを設定する必要があります。無難なところで，Osaka にするなら，

par(family="Osaka")と入力します。そして，この命令を実行した後に，hist(x$日本語の変数名)を再度実行してみてください。これで日本語もちゃんと表示されると思います。

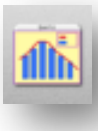
par という命令は，ヘルプで確認すればわかりますが，様々な指定ができます。フォントを指定する family=はそのひとつですが，たとえば，ps=でフォントのサイズを指定できます。デフォルトは 12 ですので，変えたい場合は ps=20 などとすれば，大きな文字で出力してくれます。また mfrow=c(2, 2)などとすれば，１枚(ページ)にいくつの図を入れるかを指定できます。c(2, 2)だと，２行２列，つまり４つの図を１枚に並べてくれます。これは結構活用できます。他にも様々な指定ができるのでヘルプなどを参考にしてください。

またこの「Quartz 2」というウインドですが，ひとつの手帳のように考えるとよいと思います。たとえば，３回 hist を実行すると，ウインドが３つ出てくるということはありません。１つのままです。以下の命令を実行してみてください。

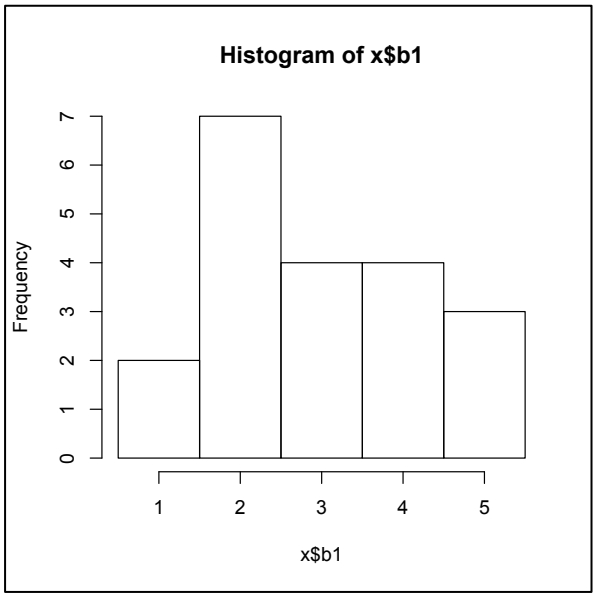
```
hist(x$b1)
hist(x$b2)
hist(x$b3)
```

「Quartz 2」のウインドには，最後の x$b3 の結果が表示されていると思います。x$b1 や x$b2 の結果はどこに行った？ということになるのですが，それらは見えない前のページに存在しています。ページをめくるには，メニューバーの「Quartz」から，「Back」や「Forward」を選ぶか，ショートカットで，「コマンド + ←/→」を使います。なお，ウインドを閉じてしまうとそれまでの結果も消えてしまいますのでご注意を。

また「Quartz」のウインドは複数開くことができます。R のアイコンの中から，右図のものをクリックすると，新しいウインド「Quartz 3」が開きます。複数開いている状態でさらに図を作成すると，その中のアクティブになっているウインド（ウインドの名前の最後に[*]のサインがついているもの）へ出力されます。もちろん，出力された図はコピペでワードやエクセルに貼り付けられます。



話がズレてしまいましたので，本題の hist にもどります。hist(x$age)の部分ですが，変数さえ指定すれば（x[,3]という「,」付きなら列番号指定も可），適当にヒストグラムを作成してくれます。これに関するオプションの設定もたくさんあるので，自分の望むものを出力できるように調べ，調整してみましょう。



今回のように，項目単位での回答の分布を見るなら，ちょっと掟破りっぽいものでもいいのかなと思ったりします。b1 から b5 は１から５までの５件法なので，それぞれのカテゴリに何ケースあるのかがはっきりとわかる方がよいでしょう。それなら，hist(x$b1, breaks=seq(0.5, 5.5, 1))などと指定するのはどうかと思います。breaks=seq の後は，グラフの左端，右端，幅の３つの数値を入れます。ヒストグラムとして適切かどうか，ということは置いておきますが，１から５の選択肢の真上に棒の部分がのるため，チェックしやすい出力といえるかもしれません。

さて，あとはいくつかの変数をまとめて，一気にヒストグラムを作らせたいのですが，これがなかなか難しいようです。

いくつか試しましたが，ことごとくエラー…

```
> hist(x)
 hist.default(x) でエラー:  'x' は数値でなければなりません
> hist(x[2:4])
 hist.default(x[2:4]) でエラー:  'x' は数値でなければなりません
> label <- c("no", "sex", "age", "b1", "b2", "b3", "b4", "b5")
> hist(x[label])
 hist.default(x[label]) でエラー:  'x' は数値でなければなりません
>
```

結局，ヘルプなどを参考にしてこのようなものに行き着きました…

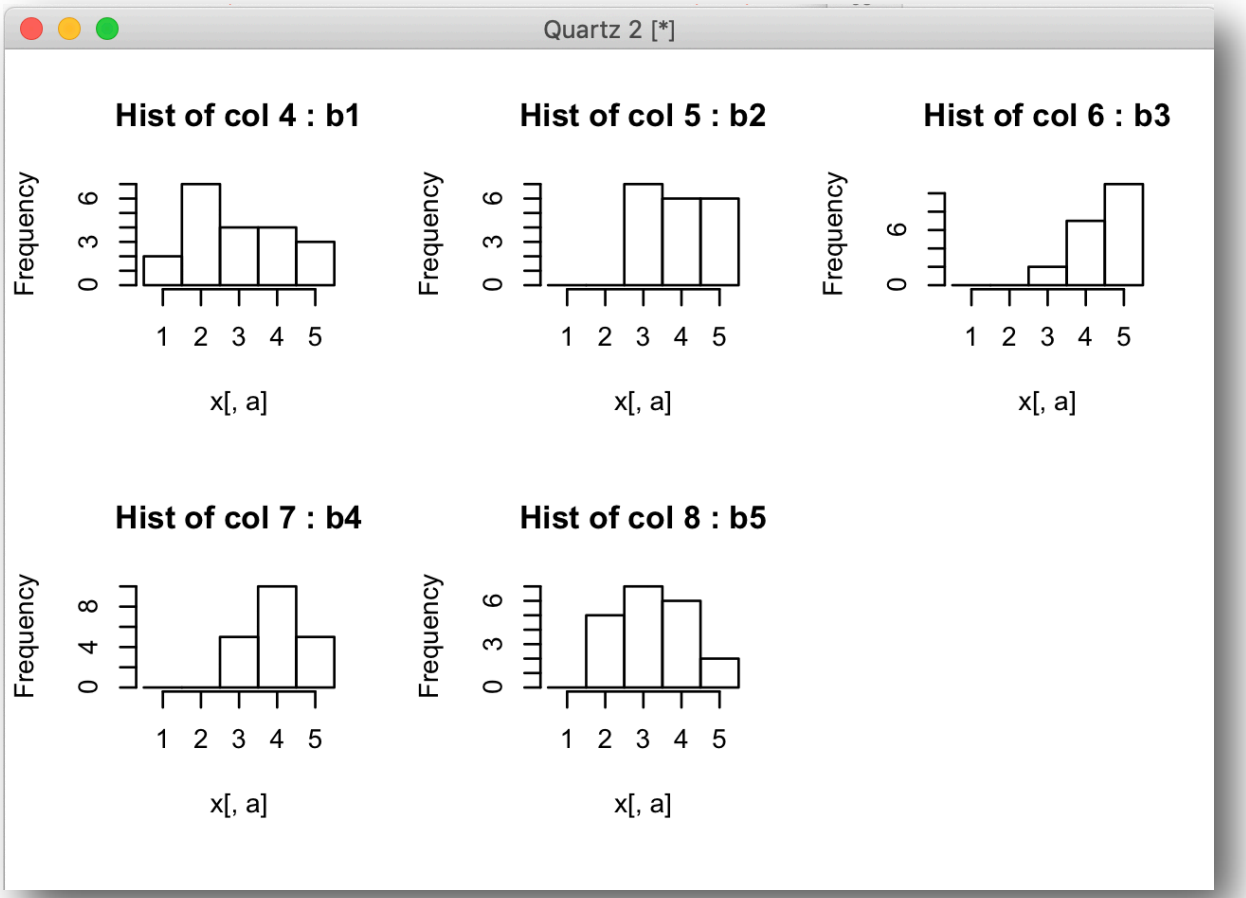
```
for (a in 4:8)
 {hist(x[,a], main=paste("Hist of col" ,a,":", names(x[a])))}
```

for (a in 4:8)は，a を４から８まで，１ずつ動かしなさいという命令です。そしてその命令を受けるのが{　}の中です。そこに hist(x[,a])を入れれば，a が先の命令を受けるので，hist(x[,4])，hist(x[,5])，… hist(x[,8])と順にやってくれるという仕組みです。main=は上部タイトルの表示をコントロールしますが，ここに列の番号と変数名を表示するようにしてみました。

これを使って，b1 から b5 までのヒストグラムを１枚（３行３列）に収まるように出力したのが以下の例です。

```
par(mfrow=c(3,3))
for (a in 4:8)  { hist(x[,a], breaks=seq(0.5, 5.5, 1),
 main=paste("Hist of col" ,a,":", names(x[a])))}
```

個人的には，なかなか満足ですが，x 軸の見出しは見なかったことに…



データチェックの最後に，欠損値の扱いについて少し触れておきます。

「NA」が含まれたデータは関数ごとに扱いが多少異なっているので注意が必要です。

どの変数に「NA」が含まれているのかを一気にチェックするには，たとえば is.na( )といった関数もあるのですが（どういう出力が得られるのかはやってみてください），以下のようにして，「NA」を含むケースだけを抽出して眺めるのもよいでしょう（最後の「,」を忘れないように）。

```
x[!complete.cases(x),]
```

「complete.cases(x)」は，「NA」を持たないかどうか（持たない場合が TRUE，持つ場

合が FALSE）を返しますが，見たいのは「NA」を持つ場合なので，否定を意味する「!」をつけて，反対の結果を得ようとしているわけです。その結果，以下のように「NA」を含むケースだけを得ることができます。

```
> x[!complete.cases(x),]
    no sex age b1 b2 b3 b4 b5
2 1002   2  20  2 NA  4  3  2
```

さて，明日からは相関などの算出に進みますが，この資料では欠損値があってもそれを含んだまま分析を進めます。これは練習の意味もあるのですが，実際は，大抵この時点（項目分析を行った後くらい）で，その後の欠損値の扱いの方針を決めます。

明日も触れますが，①欠損値を含むケースを最初からすべて取り除いてから分析（すべての分析で n が同じ）という方針，②分析ごとに欠損値を含むケースを取り除く（分析ごとに n が変わる）という方針のふたつが代表的でしょう。その他に，③欠損部分に何らかの値を代入するという方法もあります。

いずれもメリット，デメリットがあります。特に③については多重代入法など，よい方法も提案されていて，R でも対応できます。勉強してみてください。

なお，①の方針で欠損値を含むケースをすべて取り除いたデータセットを作成するには，先の complete.cases( )と， 7 日目に使った subset を合わせて，以下のようにすればできます。

```
x.com <- subset(x, complete.cases(x))
```

これで 9 日目は終了です。