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。

ネットとかを検索しても，度数分布表の作り方は数多く紹介されています。そこでつかわれているのは table です。

```
table(x[5])
table(x$年齢)
```

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，b1 には欠損値がありますが，table(x$b1)で実行すると…

```
> table(x$b1)

1 2 3 4 5
2 6 4 4 3
>
```

このように出力されます。上の段がカテゴリ，下の段が度数を表しますので，１が２名，２が６名…「NA」は…？？わかりません…

欠損値も表に含めたいなら，以下のように exclude=NULL を入れる必要があります。

```
table(x$b1, exclude=NULL)
```

するとうまくいきます。

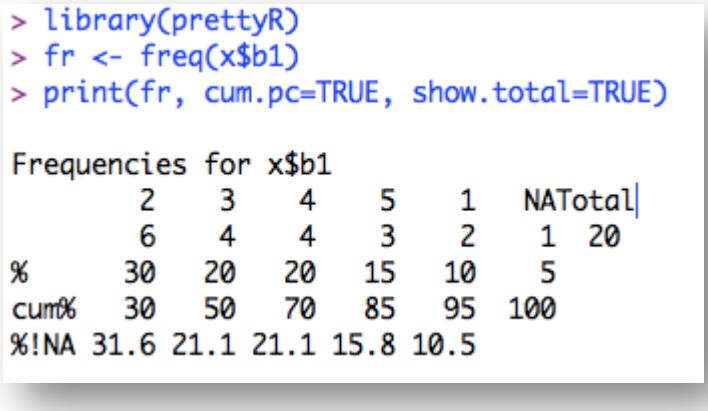
```
> table(x$b1, exclude=NULL)

   1    2    3    4    5 <NA>
   2    6    4    4    3    1
>
```

これで十分と言いたいところですが，欲張れば，%計算もやってほしい。ということで，これらの計算も一緒にやらせることはできないかと探したら，prettyR というパッケージにある，freq が使えるということでした。

そこで，まずは prettyR というパッケージを入手してください。

入手できたら，まずはそれを呼び出し，以下のように入力してください。

```
library(prettyR)
fr <- freq(x$b1)
print(fr, cum.pc=TRUE, show.total=TRUE)
```



これで b1 の，度数合計（Total），各カテゴリの度数と割合（NA を除いた%（%!NA）と，それを込みにした%（cum%），累積%（%）とともに作成してくれます。

ただし，このままだと，カテゴリを度数の多い順に勝手に並べ替えてしまいます（右図を見てください）。それを防ぐためには，2行目に decr.order=FALSE を入れ，fr <- freq(x$b1, decr.order=FALSE) としておく必要があります。

なおこの2行の命令を読み解いてみると，freq で計算した結果を一度 fr という新しい入れ物に入れ，さらに print で fr の出力調整をしているということになります。Rには，このように計算結果をいったん何かに入れ，結果表示コントロールを別に行って出力するという場合が少なくないようです。基本的に，これは結果を見やすくするためのようです。

ちなみに3行目の print の部分をつけないと，Total と cum%は出力されません。show.total=TRUE と cum.pc=TRUE が入っているところに注目すればすぐにピンとくるでしょう。

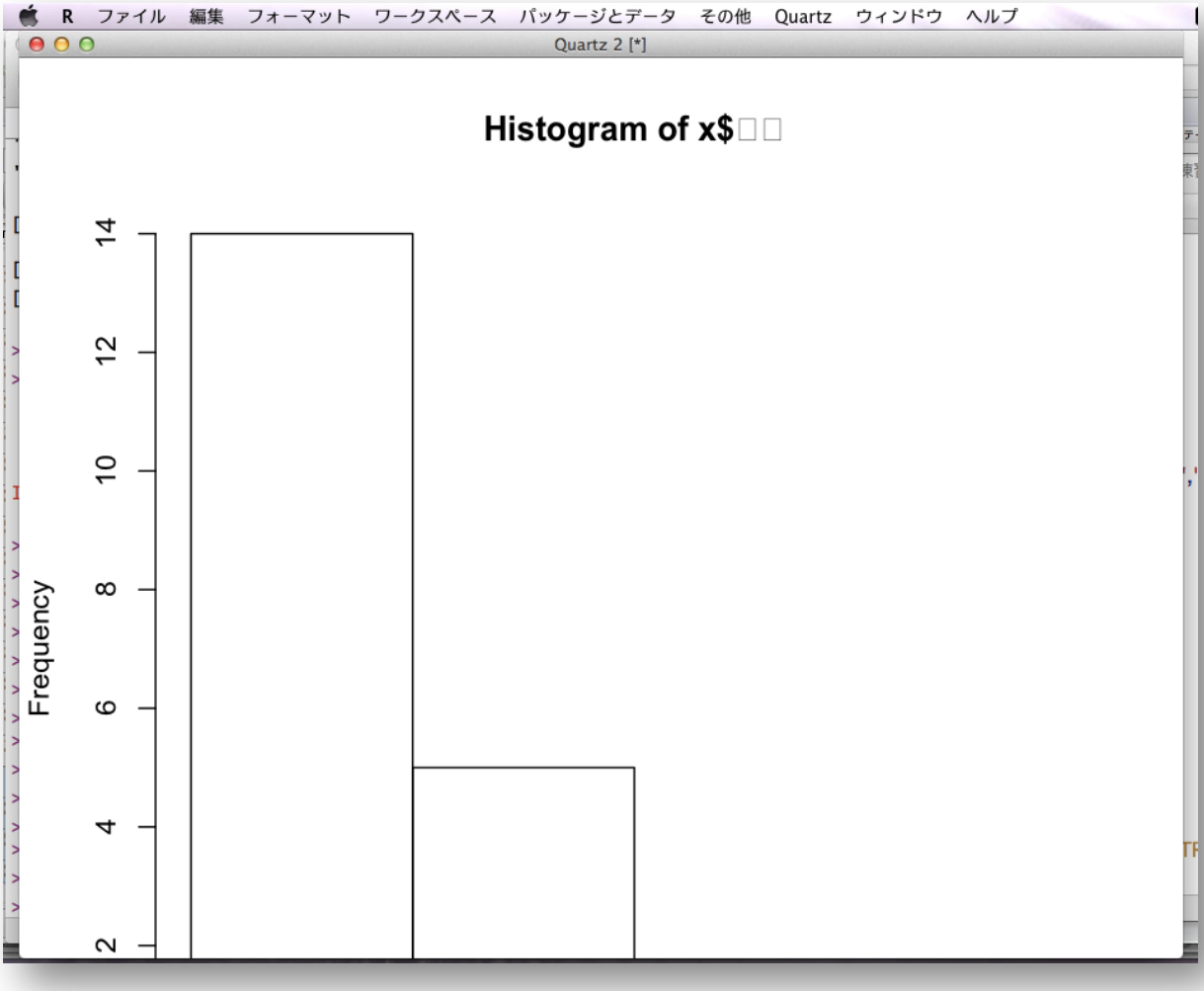
複数変数をまとめてやりたい時は，freq(x[2:5], decr.order=FALSE)といった指定でうまくいきます。また v3 <- c("性別","b2","b4","no")などと変数名をまとめておいて，freq(x[v3], decr.order=FALSE)で一気に計算することもできます。

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$年齢)
```

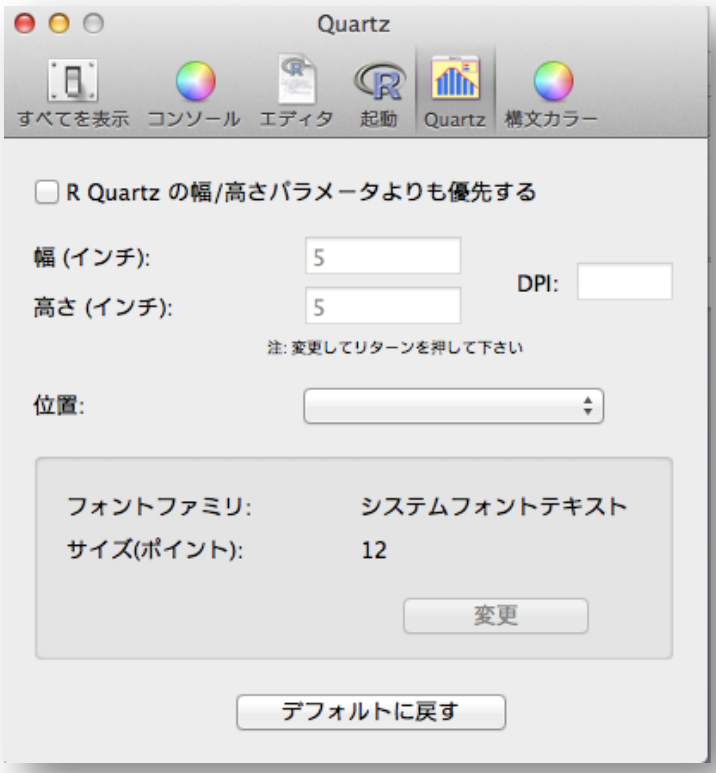
これだけを，実行してみてください。

これでどういう表示になるのか確認してください。たとえば私の Mac（MacBook Air の 11 インチ）では… 以下のように画面よりも大きな図（Quartz 2 という名前のウインド）になってしまいました…

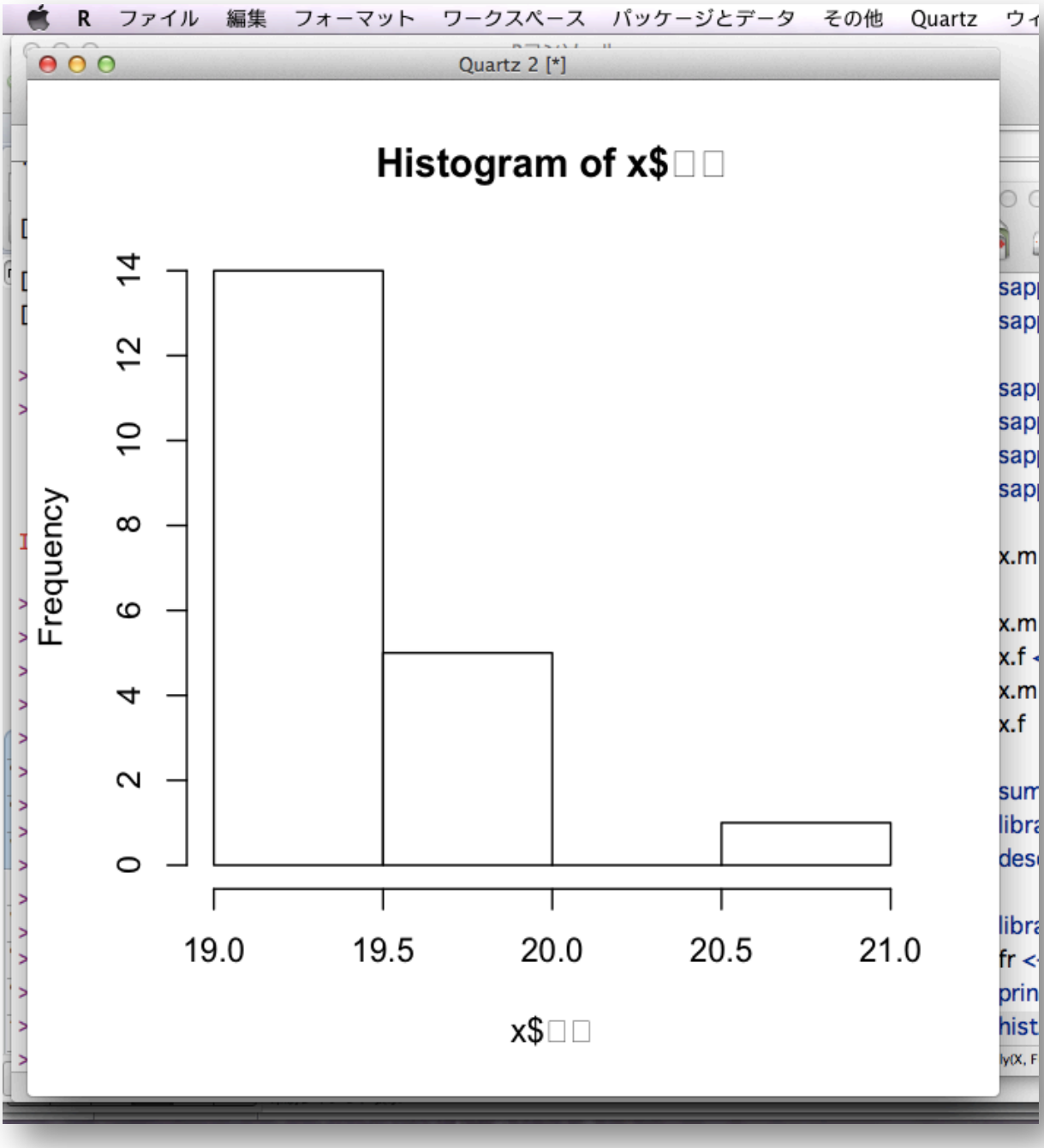


このように，R で作図をしようとすると，図用の新しいウインドが開きます。しかし，このウインドはスクロールしないので，まずは自分の利用環境に合ったウインドのサイズ指定をしておく方がよいでしょう。

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。ちなみに私の環境であれば，幅も高さも５インチで，だいたいきれい



に収まりました。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，サイズ的には画面におさまったものの，x$□□と，本来なら「年齢」と表示されるべきところが文字化けしてしまっています。デフォルトのフォント設定のままだと，アルファベットや数字は問題なく表示してくれますが，日本語はおかしくなるので，利用するフォントを設定しておきます。無難なところで，Osaka にするなら，`par(family="Osaka")` と入力します。そして，この命令を実行した後に，`hist(x$年齢)`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。

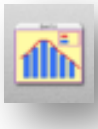
par という命令は，ヘルプで確認すればわかりますが，とても多くの指定ができます。フォントを指定する family=はそのひとつですが，たとえば，ps=でフォントのサイズを指定できます。デフォルトは12ですので，変えたい場合はps=20などとすれば，大きな文字で出力してくれます。またmfrow=c(2, 2)などとすれば，１枚(ページ)にいくつの図を入れるかを指定できます。c(2, 2)だと，２行２列，つまり４つの図を１枚に並べてくれます。なお，いまは図表のタイトルが「Histogram of x$年齢」になっていますが，hist(x$年齢 , main="○○")と入力すれば，自由に図表のタイトルをつけることができます。

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。

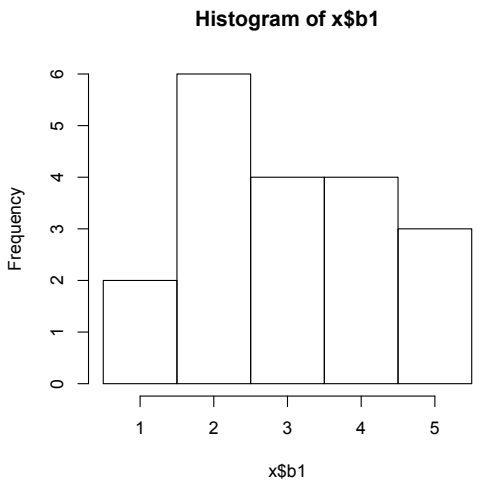
```
hist(x$b1)
hist(x$b2)
hist(x$b3)
```

「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。



さて，hist(x$年齢)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webなどで調べてみましょう。



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，私にとっては使いやすい（チェックしやすい）ものになります。

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)
 以下にエラー hist.default(x) :  'x' は数値でなければなりません
> hist(x[2:4])
 以下にエラー hist.default(x[2:4]) :  'x' は数値でなければなりません
> v <- c("no","性別","年齢","b1","b2","b3","b4","b5")
> hist(x[v])
 以下にエラー hist.default(x[v]) :  'x' は数値でなければなりません
```

結局，このようなものに行き着きました…

```
for (a in 4:8)  { hist(x[,a]) }
```

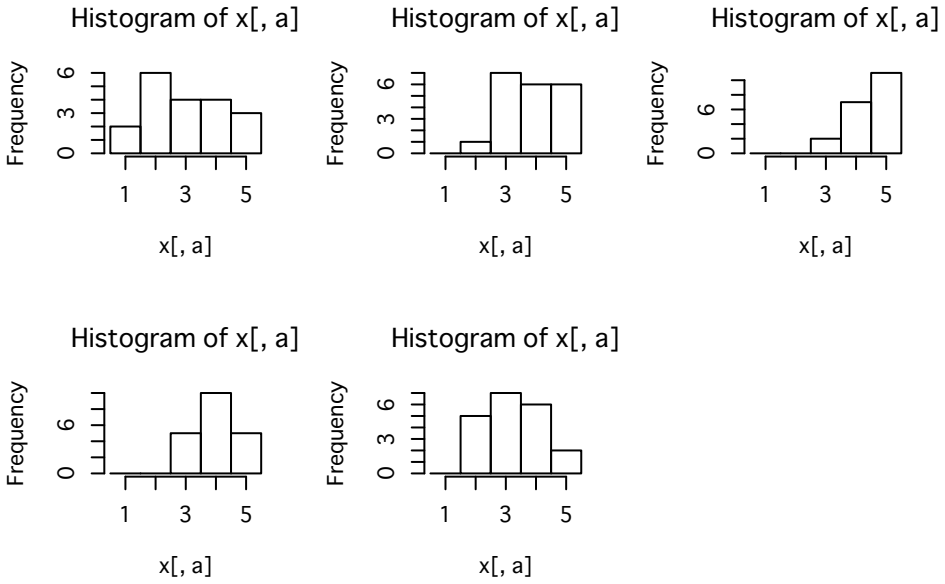
for (a in 4:8)は，aを4から8まで，1ずつ動かしなさいという命令です。そしてその命令を受けるのが{ }の中です。そこにhist(x[,a])を入れれば，aが先の命令を受けるので，hist(x[,4])，hist(x[,5])，… hist(x[,8])と順にやってくれるという仕組みです。

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(family="Osaka", mfrow=c(3,3))
for (a in 4:8)  { hist(x[,a], breaks=seq(0.5, 5.5, 1)) }
```

まあ，「これでもいいか…」くらいの出来です。問題はforを使うと次ページの結果のように変数名が表示されないこと。間違えないようにしないと…

これで9日目は終了です。明日は，相関係数を算出してみます。


Histogram of x[, a]
Frequency
0 3 6
1 3 5
x[, a]
Histogram of x[, a]
Frequency
0 3 6
1 3 5
x[, a]
Histogram of x[, a]
Frequency
0 6
1 3 5
x[, a]
Histogram of x[, a]
Frequency
0 6
1 3 5
x[, a]
Histogram of x[, a]
Frequency
0 3 6
1 3 5
x[, a]