### The Art of R Programming

A Tour of Statistical Software Design

by Norman Matloff

errata updated to print 14

<table>
<thead>
<tr>
<th>Page</th>
<th>Error</th>
<th>Correction</th>
<th>Print corrected</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>If not, see Appendix A. for installation instructions.</td>
<td>If not, see Appendix A for installation instructions.</td>
<td>Print 3</td>
</tr>
<tr>
<td>51</td>
<td><code>vud &lt;- diff(d)</code></td>
<td><code>vud &lt;- diff(v)</code></td>
<td>Print 3</td>
</tr>
<tr>
<td>53</td>
<td><code>for (gen in c(&quot;M&quot;,&quot;F&quot;)) grps[[gen]] &lt;- which(aba==gen)</code></td>
<td><code>for (gen in c(&quot;M&quot;,&quot;F&quot;)) grps[[gen]] &lt;- which(aba[,1]==gen)</code></td>
<td>Print 2</td>
</tr>
<tr>
<td>65, 66</td>
<td><code>newimg@grey &lt;- (1-q) * img@grey + q * randomnoise</code></td>
<td><code>newimg@grey[rows,cols] &lt;- (1-q) * img@grey[rows,cols] + q * randomnoise</code></td>
<td>Print 4</td>
</tr>
<tr>
<td>67</td>
<td><code>x &lt;- c(5,12,13)</code></td>
<td><code>x[2 %% 2 == 1,]</code></td>
<td>Print 3</td>
</tr>
<tr>
<td></td>
<td><code>x[2 %% 2 == 1,]</code></td>
<td><code>[1,] 1 2</code></td>
<td>Print 3</td>
</tr>
<tr>
<td></td>
<td><code>[,1] [,2]</code></td>
<td><code>[2,] 3 4</code></td>
<td>Print 3</td>
</tr>
<tr>
<td>77</td>
<td>Recall that due to the symmetry of the matrix, we skip the early part of each row, as is seen in the expression <code>((i+1):(1x-1))</code> in line 18. But that means that the call to <code>which.min()</code> in that line will return the minimum's index relative to the range <code>((i+1):(1x-1))</code>.</td>
<td>Recall that due to the symmetry of the matrix, we skip the early part of each row, as is seen in the expression <code>((i+1):(1x-1))</code> in line 18. But that means that the call to <code>which.min()</code> in that line will return the minimum's index relative to the range <code>((i+1):(1x-1))</code>.</td>
<td>Print 3</td>
</tr>
<tr>
<td>93</td>
<td><code>nwords &lt;- length(ssnyt)</code></td>
<td><code>frequ9 &lt;- sapply(ssnyt[round(0.9*nwords):nwords],length)</code></td>
<td>Print 2</td>
</tr>
<tr>
<td></td>
<td><code>barplot(frequ9)</code></td>
<td><code>barplot(frequ9)</code></td>
<td></td>
</tr>
<tr>
<td>Page</td>
<td>Error</td>
<td>Correction</td>
<td>Print corrected</td>
</tr>
<tr>
<td>------</td>
<td>-------</td>
<td>------------</td>
<td>----------------</td>
</tr>
<tr>
<td>116</td>
<td>shang3</td>
<td>shang4</td>
<td>Print 2</td>
</tr>
<tr>
<td>128</td>
<td>fl.1 a bc 5 2 1 12 1 1 13 1 0</td>
<td>fl.1 a bc 5 2 0 12 1 1 13 2 1</td>
<td>Print 2</td>
</tr>
<tr>
<td>130</td>
<td>&gt; ctt/5</td>
<td>&gt; ctab/5</td>
<td>Print 2</td>
</tr>
<tr>
<td>131</td>
<td>&gt; apply(ctt,1,sum)</td>
<td>&gt; apply(cttab,1,sum)</td>
<td>Print 2</td>
</tr>
<tr>
<td>133</td>
<td>f(argslist[[1]],argslist[[2]],...)</td>
<td>f(argslist[[1]],argslist[[2]],...)</td>
<td>Print 2</td>
</tr>
<tr>
<td>137</td>
<td>This says that z[1], 0.88114802, fell into bin 9, which was (0.8,0.9]; z[2], 0.28532689, fell into bin 3; and so on.</td>
<td>This says that z[1], 0.88114802, fell into bin 9, which was (0.8,0.9]; z[2], 0.28532689, fell into bin 3; and so on.</td>
<td>Print 2</td>
</tr>
<tr>
<td>148</td>
<td>Good software design, however, should be mean that you can glance through a function's code . . .</td>
<td>Good software design, however, should mean that you can glance through a function's code . . .</td>
<td>Print 3</td>
</tr>
<tr>
<td>151</td>
<td>&gt; f(3,2) [1] 1 &gt; g &lt;- function(h,a,b) h(a,b) &gt; g(f1,3,2) [1] 5 &gt; g(f2,3,2) [1] 1</td>
<td>&gt; f(3,2) [1] 1 &gt; g &lt;- function(x) x^2 &gt; body(g) &lt;- quote(2*x+3) &gt; g function (x) 2 * x + 3 &gt; g(0) [1] 19</td>
<td>Print 5</td>
</tr>
<tr>
<td>Page</td>
<td>Error</td>
<td>Correction</td>
<td>Print corrected</td>
</tr>
<tr>
<td>------</td>
<td>-------</td>
<td>------------</td>
<td>----------------</td>
</tr>
</tbody>
</table>
| 151  | ```
> g <- function(h,a,b) h(a,b)
> body(g) <- quote(2*x + 3)
> g
function (x)
2 * x + 3
> g(3)
[1] 9
``` | ```
> g <- function(h,a,b) h(a,b)
> body(g) <- quote(2*x + 3)
> g
function (x)
2 * x + 3
> x <- 3
> g(3)
[1] 9
``` | Print 2 |
| 155  | ```
> f(2)
[1] 88
``` | ```
> f(2)
[1] 112
``` | Print 2 |
| 160  | ```
> oddsevens
function(v){
  odds <- which(v %% 2 == 1)
  evens <- which(v %% 2 == 0)
  list(o=odds,e=evens)
}
``` | ```
> oddsevens
function(v){
  odds <- which(v %% 2 == 1)
  evens <- which(v %% 2 == 0)
  list(o=odds,e=evens)
}
``` | Print 2 |
| 163  | ```
makecorpdfs(c("MICROSOFT CORPORATION","ms","INTEL CORPORATION","intel","SUN MICROSYSTEMS, INC.","sun","GOOGLE INC.","google")
``` | ```
makecorpdfs(c("MICROSOFT CORPORATION","ms","INTEL CORPORATION","intel","SUN MICROSYSTEMS, INC.","sun","GOOGLE INC.","google"))
``` | Print 2 |
<p>| 164  | . . . when we discuss appropriate use global variables in the next section. | . . . when we discuss appropriate use of global variables in the next section. | Print 3 |
| 176  | 3. Within f(), piece together the results of (b) to solve the original problem. | 3. Within f(), piece together the results of (2) to solve the original problem. | Print 3 |
| 178  | . . . while the right subtree stores the elements that are larger than the value in this mode. | . . . while the right subtree stores the elements that are larger than the value in this node. | Print 3 |</p>
<table>
<thead>
<tr>
<th>Page</th>
<th>Error</th>
<th>Correction</th>
<th>Print</th>
</tr>
</thead>
</table>
| 185  | 26 \end{Code}  
27  
28 Let's test it.  
29  
30 \begin{Code}  
31 > b <- newbookvec(c(3,4,5,5,12,13))  
32 > b  
33 $vec  
34 [1] 3 4 5 5 12 13  
35  
36 $wrts  
37 [1] 0 0 0 0 0 0  
38  
39 attr(,"class")  
40 [1] "bookvec"  
41 > b[2]  
42 [1] 4  
43 > b[2] <- 88 # try writing  
44 > b[2] # worked?  
45 [1] 88  
46 > b$wrts # write count incremented?  
47 [1] 0 1 0 0 0 0 | Let's test it.  
> b <- newbookvec(c(3,4,5,5,12,13))  
> b  
$vec  
[1] 3 4 5 5 12 13  
$wrts  
[1] 0 0 0 0 0 0  
attr(,"class")  
[1] "bookvec"  
> b[2]  
[1] 4  
> b[2] <- 88 # try writing  
> b[2] # worked?  
[1] 88  
> b$wrts # write count incremented?  
[1] 0 1 0 0 0 0 | Print 3 |
| 191  | The expression notp[-i] computes the product of all the elements of notp, . . . | The expression prod(notp[-i]) computes the product of all the elements of notp, . . . | Print 3 |
| 194  | For instance, to find our more about the chi-square function for quantiles, . . . | For instance, to find out more about the chi-square function for quantiles, . . . | Print 3 |
| 197  | > a <- matrix(c(1,1,-1,1),nrow=2,ncol=2)  
> b <- c(2,4)  
> solve(a,b)  
[1] 3 1  
> solve(a)  
[,1] [,2]  
[1,] 0.5 0.5  
[2,] -0.5 0.5 | > a <- matrix(c(1,1,-1,1),nrow=2)  
> b <- c(2,4)  
> solve(a,b)  
[1] 1 3  
> solve(a)  
[,1] [,2]  
[1,] 0.5 -0.5  
[2,] 0.5 0.5 | Print 3 |
<p>| 206  | Recalling that R lists are often used to store several related variables in one basket, we se up a list condat. | Recalling that R lists are often used to store several related variables in one basket, we set up a list condat. | Print 3 |
| 228  | &gt; save(hz,&quot;hzfile&quot;) | &gt; save(hz, file=&quot;hzfile&quot;) | Print 3 |</p>
<table>
<thead>
<tr>
<th>Page</th>
<th>Error</th>
<th>Correction</th>
<th>Print corrected</th>
</tr>
</thead>
<tbody>
<tr>
<td>264</td>
<td>On a Mac, call <code>macintosh()</code>.</td>
<td>On a Mac, call <code>quartz()</code></td>
<td>Print 3</td>
</tr>
<tr>
<td>276</td>
<td><code>g &lt;- function(t) { return (t^2+1)^0.5 } # define g()</code></td>
<td><code>g &lt;- function(t) { return ((t^2+1)^0.5) } # define g()</code></td>
<td>Print 3</td>
</tr>
<tr>
<td>295</td>
<td>returns the minimum value of <code>d[i,j]</code>, <code>i != j</code>, and the row/col attaining that minimum, for square symmetric matrix <code>d</code>; no special policy on ties; motivated by distance matrices</td>
<td># returns the minimum value of <code>d[i,j]</code>, <code>i != j</code>, and the row/col attaining that minimum, for square symmetric matrix <code>d</code>; no special policy on ties; motivated by distance matrices</td>
<td>Print 3</td>
</tr>
<tr>
<td>345</td>
<td>As of this writing, GPU has not yet become common among R users.</td>
<td>As of this writing, GPU <strong>programming</strong> has not yet become common among R users.</td>
<td>Print 3</td>
</tr>
</tbody>
</table>