Errata for Carpenter's Guide to Innovative SAS Techniques

Section 2.1.3, pp 42-43

In each of the four text code boxes the `sodium>'142'` becomes `sodium>14.2.` *e.g.*

```
data labs(keep=subject visit labdate)❼;
   set
advrpt.lab_chemistry(rename=(labdt=labdate))❻;
   if sodium>14.2;
   run;
```

Section 2.9.5, p. 96

Top of page; change length of sodium, potassium, and chloride from $12 to 8 (5th line of the text box).

```
data _null_;
   length subject $3 ❿
          visit    8
          labdt    8
          sodium potassium chloride 8;
   declare hash chem (dataset:'advrpt.lab_chemistry', ordered:'Y') ;
      chem.definekey ('subject','visit','labdt','sodium','potassium','chloride');
      chem.definedata('subject','visit','labdt','sodium','potassium','chloride');
      chem.definedone () ;
   call missing(subject,visit,labdt, sodium, potassium,chloride);
   chem.output(dataset:'nodups');
   run;
```

Section 3.6.5, p. 150

Middle to the bottom of the page.

The VNEXT routine can be especially helpful as it can return not only the variable's name, but its type (numeric/character) and length as well. In addition it can be used to step through, one-at-a-time, all the variables (including temporary variables) in a data set.

```
data attrib(keep=name p_type p_len);
   set advrpt.lab_chemistry;
   retain name '        ' p_type ' ' p_len .; ❶
   call vnext(name,p_type,p_len); ❷
   output attrib;
   stop;
   run;
```

Here VNEXT is executed once so it will gather information on the first variable on the PDV

❶ P_TYPE will be used to store the type (N or C) of the variable read by VNEXT. The variable P_LEN will hold the length of the variable.

❷ In this example we are executing VNEXT once so VNEXT will retrieve only the attributes of the first variable on the PDV, which in this case is SUBJECT.

❸ The attributes of the variable SUBJECT are stored.

```
3.6.5 Using Variable Information Functions
VNEXT - Attributes of the first Variable
Obs     name      p_type    p_len

 1     SUBJECT      C         3 ❸
```

Section 4.3, p. 192
The text in the code box is randomly bolded. There should be no bolding.

Section 4.4, p. 193
The symbols in the code box are corrupted.

|	❹
{	❷

```
title1 '4.4a Showing SORT Meta-data';
proc sort data=advrpt.lab_chemistry
          out=lab_chem
          noduplicates; ❹
   by subject visit labdt; ❷
   run;
proc contents data=lab_chem;
   run;
```

Section 5.1, p. 198
The 5.1 in the text box at the bottom of the page is bolded. Remove bolding.

```
    .... Code not shown ....
 proc format;
   value $genlnk ❶
      'M' = '#males'
      'F' = '#females';
   run;

ods pdf anchor='master';
ods proclabel='Overall';
proc tabulate data=tabdat.clinics;
   class sex ;
   classlev sex/ style={url=$genlnk. ❷
                        foreground=blue};
   var wt;
   table sex=' ',
         wt*(n median min max)
         / box='Gender';
   run;

ods pdf anchor='males'; ❸
ods proclabel='Males'; ❹
title2 link='#master' ❺ 'Return to Master';
title3 c=blue 'Males';
proc print data=tabdat.clinics;
   where sex='M'; ❻
   var lname fname ht wt;
   run;
              .... Code not shown ....
```

Section 11.4.4, p.351
The code box contains a couple of errors.

a) The word Master appears three times and should be lower case when it is associated with a link. It is OK to be capitalized in the title.

b) There is an extraneous character and a font change in TITLE2 at ❺.

c) The format labels for #Males and #Females (lines 4 & 5 in the text box) ❶ should be lowercase.