BC CANCER
Provincial Health Services Authority

Making Friends: Clean Code

A brief guide on loving yourself and others through better programming

Brendan Bakos

Outline

- Writing readable programs
- Improving comments
- Making friends

Writing Readable Programs

Key Concept

- Minimize the time it would take someone else to understand

Key Concept

- Minimize the time it would take someone else to understand
  - Write tidy, consistent code

Key Concept

- Minimize the time it would take someone else to understand
  - Write tidy, consistent code
  - Choose specific names that describe the value or purpose of the variable/dataset

Case Study

```
data dat;do i = 1 to 4;do j = 1 to 5;output;end;end;run;
```

	i	j
1	1	1
2	1	2
3	1	3
4	1	4
5	1	5
6	2	1
7	2	2
8	2	3
9	2	4
10	2	5
11	3	1
12	3	2
13	3	3
14	3	4
15	3	5
16	4	1
17	4	2
18	4	3
19	4	4
20	4	5

Case Study

```
data dat;do i = 1 to 4;do j = 1 to 5;output;end;end;run;
```

Case Study

```
data dat;do i = 1 to 4;do j = 1 to 5;output;end;end;run;
```

1. Make a statement

```
data dat;
  do i = 1 to 4;
  do j = 1 to 5;
  output;
  end;
  end;
  run;
```

Case Study

```
data dat;do i = 1 to 4;do j = 1 to 5;output;end;end;run;
```

1. Make a statement

```
data dat;
  do i = 1 to 4;
  do j = 1 to 5;
  output;
  end;
  end;
  run;
```

2. Tab

```
data dat;
      do i = 1 to 4;
      do j = 1 to 5;
      output;
      end;
      end;
  run;
```

Case Study

```
data dat;do i = 1 to 4;do j = 1 to 5;output;end;end;run;
```

1. Make a statement

```
data dat;
do i = 1 to 4;
do j = 1 to 5;
output;
end;
end;
run;
```

2. Tab

```
data dat;
      do i = 1 to 4;
      do j = 1 to 5;
      output;
      end;
      end;
run;
```

3. Tab tab tab

```
data dat;
     do i = 1 to 4;
          do j = 1 to 5;
               output;
          end;
     end;
run;
```

Case Study

```
data dat;do i = 1 to 4;do j = 1 to 5;output;end;end;run;
```

4. Descriptive variable names

```
data study_design;
    do trial = 1 to 4;
        do treatment = 1 to 5;
            output;
        end;
    end;
run;
```

Case Study

```
data dat;do i = 1 to 4;do j = 1 to 5;output;end;end;run;
```

4. Descriptive variable names

```
data study_design;
     do trial = 1 to 4;
          do treatment = 1 to 5;
               output;
          end;
     end;
 run;
```

BE CONSISTENT

General Guidelines

- Programs should look neat

General Guidelines

- Programs should look neat
  - Use white space

General Guidelines

- Programs should look neat
  - Use white space
  - Indents

General Guidelines

- Programs should look neat
  - Use white space
  - Indents
  - Match do/end

General Guidelines

- Programs should look neat
  - Use white space
  - Indents
  - Match do/end
  - One statement per line

General Guidelines

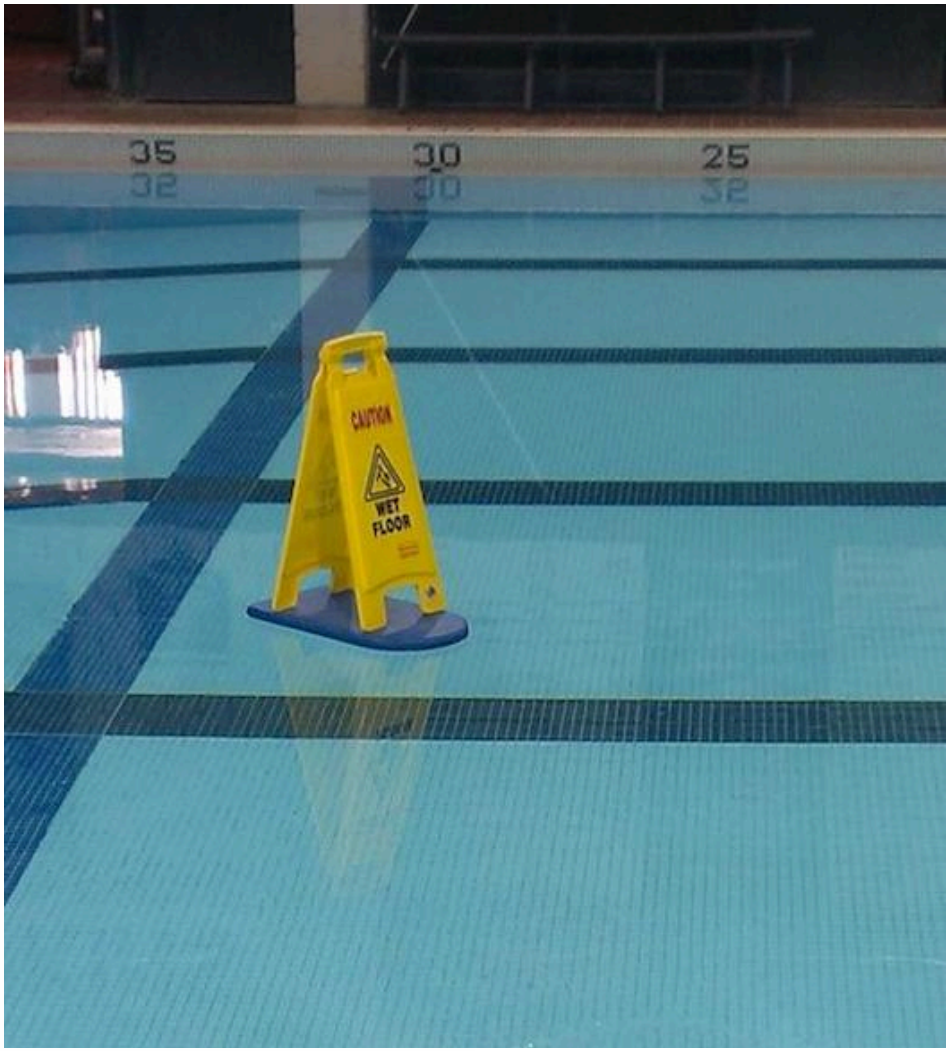
- Programs should look neat
  - Use white space
  - Indents
  - Match do/end
  - One statement per line
  - Consistency

/* Program Comments */

Key Concept

- Write comments that help the reader understand the code more easily
  - **Why**
  - What
  - How

Comments



Useless comments

```
data final_cohort;
      set raw_data;

      * remove participants under age 40;
      if age < 40 then delete;

      * categorize age;
      if 40 <= age <= 47 then age_cat=1;
          else if 47 < age  <= 55 then age_cat=2;
          else if  55 < age then age_cat=3;

      * remove average weekly calories from final dataset;
      drop avg_weekly_calories;
 run;
```

Comments that make friends

```
data final_cohort;
     set raw_data;

     /* participants under the age of 40 were ineligible to participate in the study */
     if age < 40 then delete;

     /* age category may look weird, but cluster anlaysis was used to determine groups */
     if 40 <= age <= 47 then age_cat=1;
         else if 47 < age  <= 55 then age_cat=2;
         else if  55 < age then age_cat=3;

     /* Dr. Doctor does not think average weekly caloric intake is necessary to adjust for
         as there is limited evidence it is associated with the outcome */
     drop avg_weekly_calories;
 run;
```

Headers

```
*************************************************************************************************;
**** Program Name: VanSUG_Presentation                                                        ****;
**** Objective...: Give insightful talk                                                       ****;
****                                                                                          ****;
**** Directory...: C:\headers                                                                 ****;
****                                                                                          ****;
**** Author......: BB                                                                        ****;
**** Date........: May 9, 2018                                                                ****;
**** SAS Version.: 9.4 (TS1M3)                                                                ****;
****                                                                                          ****;
**** Input Files.: None                                                                       ****;
**** Output Files: great_presentation.pptx                                                    ****;
****                                                                                          ****;
**** Notes.......:                                                                            ****;
****                                                                                          ****;
**** Date Modified  By      Reason                                                            ****;
**** -------------  ----    -------------------------                                         ****;
*************************************************************************************************;
```

Tying it all together

```
%macro pro_programmer_macro(var1, var2, var3, var4);
 %if %lowcase(&var4.)=incid %then %do;
 proc datasets lib=work nolist;
 delete parms_&var2._&var4;
 quit; run;
 %if &var3=1 %then %do;%do i=1 %to 2;
 data parms_0&i.;
 set modelfit1_0&i.;
 if incid_simple_grp="&var1.";
 model=&i.;
 run;
 proc append base=parms_&var2._&var4 data=parms_0&i. force;
 run;
 %end;%end;

      .
      .
      .
      .
 %end;
 %mend;
```

```
/*************************************************************************************************/
/* OBJECTIVE...: Get model fit parameters for all projection models for a given site             */
/* INSTRUCTIONS: This macro is used after running the incidence or mortality projections         */
/*               program (in C:\projections). The 'site' variable (aka incid_simple_grp)         */
/*               must have the correct capitalization. Specify incid or mort for incidmort       */
/*               depending on whether incidence or mortality projections were run.               */
/* NOTES.......: Refer to the documentation                                                      */
/*               (C:\Users\Enlightened_programmer\projections_documentation.docx) for the        */
/*               section of a given site and further information on the programs                 */
/*************************************************************************************************/
%macro get_model_parms_please(site, name, section, incidmort);

    %if %lowcase(&incidmort.)=incid %then %do;

        /* datasets are iteratively appended, so the base dataset needs to be cleared
            in case it already exists */
        proc datasets lib=work nolist;
            delete parms_&name._&incidmort;
        quit; run;

        %if &section=1 %then %do;
            %do i=1 %to 2;
                data parms_0&i.;
                    set modelfit1_0&i.;

                    if incid_simple_grp="&site.";

                    model=&i.;
                run;

                proc append base=parms_&name._&incidmort data=parms_0&i. force;
                run;
            %end;
        %end;


        .
        .
        .
        .
    %end;
%mend;
```

```
%macro pro_programmer_macro(var1, var2, var3, var4);
%if %lowcase(&var4.)=incid %then %do;
proc datasets lib=work nolist;
delete parms_&var2._&var4;
quit; run;
%if &var3=1 %then %do;%do i=1 %to 2;
data parms_0&i.;
set modelfit1_0&i.;
if incid_simple_grp="&var1.";
model=&i.;
run;
proc append base=parms_&var2._&var4 data=parms_0&i. force;
run;
%end;%end;

    .
    .
    .
    .
%end;
%mend;
```

```
/*****************************************************************************************/
/* OBJECTIVE...: Get model fit parameters for all projection models for a given site     */
/* INSTRUCTIONS: This macro is used after running the incidence or mortality projections */
/*               program (in C:\projections). The 'site' variable (aka incid_simple_grp) */
/*               must have the correct capitalization. Specify incid or mort for incidmort */
/*               depending on whether incidence or mortality projections were run.       */
/* NOTES.......: Refer to the documentation                                              */
/*               (C:\Users\Enlightened_programmer\projections_documentation.docx) for the */
/*               section of a given site and further information on the programs         */
/*****************************************************************************************/
%macro get_model_parms_please(site, name, section, incidmort);

    %if %lowcase(&incidmort.)=incid %then %do;

        /* datasets are iteratively appended, so the base dataset needs to be cleared
            in case it already exists */
        proc datasets lib=work nolist;
            delete parms_&name._&incidmort;
        quit; run;

        %if &section=1 %then %do;
            %do i=1 %to 2;
                data parms_0&i.;
                    set modelfit1_0&i.;

                    if incid_simple_grp="&site.";

                    model=&i.;
                run;

                proc append base=parms_&name._&incidmort data=parms_0&i. force;
                run;
            %end;
        %end;

    .
    .
    .
    .
    %end;
%mend;
```

Summary

- Make friends by being consistent, using white space, and using indents

- Make friends with insightful comments

/* Thank you */

Boswell D, Foucher T. The Art of Readable Code. "O'Reilly Media, Inc."; 2011.

Lafler KP. Best Practicing Programming Techniques for SAS Users. 2017.
https://support.sas.com/resources/papers/proceedings17/0175-2017.pdf

Winn TJ. Guidelines for Coding of SAS Programs. 2004.
http://www2.sas.com/proceedings/sugi29/258-29.pdf

Martin C, Martin L. Clean-up, Comments and Code – Making it Maintainable. 2004.
https://stats.idre.ucla.edu/wp-content/uploads/2016/02/ap2004.pdf