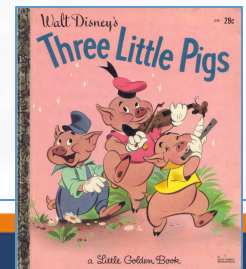


Maximizing Program Reviewability as Told Through the Story of *'The Three Little Pigs'*

Shelley Dunn
February 22, 2016

Objectives

- Provide rationale for program review
- Discuss how to maximize program reviewability
- Outline key components to creating easy to review (fairy tale-like) programs
- Provide examples to illustrate best practices and guidance on simplifying code, using macros, incorporating metadata, and assisting reviewability through documentation
- Keep the audience entertained, engaged, and open to learning (*bring on the 3 lil' pigs*)



Survey: Show of Hands/Input

- How many of you have programmed / written code?
- How many of you have reviewed programming code?
 - QC?
 - Code Review?
- How many of you have modified someone else's programming code? Any issues?
- Would you prefer to write your own code or review someone else's code? Why?



Rationale for Program Review

- Programming
 - Programming for **Clinical Trials** → End product in the form of Tables, Listings, Figures (TLFs) or Data Sets
 - Programming as part of **Software Development** → End product is the actual program
- Review
 - Quality control (QC) by an independent programmer is done to ensure **accuracy** of program(s) and output
 - Code review, as part of software development lifecycle (SDLC), is done to ensure program **accuracy and efficiency**



Program Accuracy!

- Inaccurate clinical programming → Refuse to file
- Inaccurate software development → Bugs

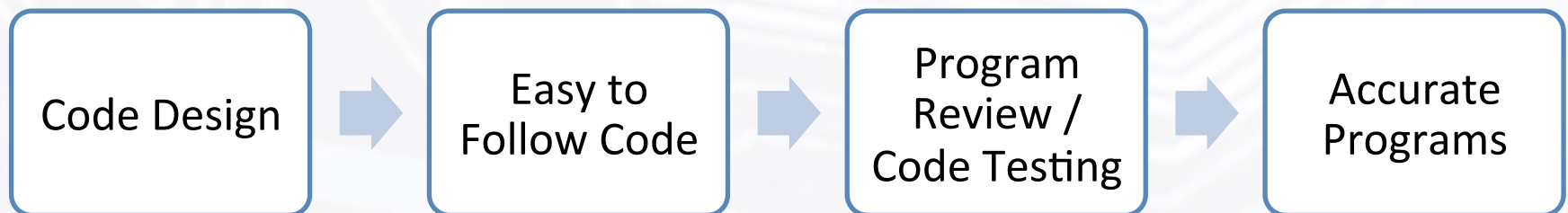
TRUE STORY!
[www.npr](http://www.npr.org)



- Cars have become computers on wheels. Crash the computer, and you could crash the car
- Two hackers decided they wanted to try doing that with a car that's considered pretty strong in terms of software, not just hardware. They chose the **Tesla Model S**. And — guess what — they broke in!

Main Ideas To Enhance Accuracy

- We want programs to be **accurate**!
- The best chance of creating **accurate** programs is **program review and/or code testing**
- **Program review and/or code testing** can be simplified or enhanced by producing **easy to follow code**
- **Easy to follow code** is more likely to occur when starting with a well thought out **code design**, e.g., write out input, set of steps, output, workflow, etc.



☐ Well thought out and well written code is easier to review and has a greater chance of accuracy 😊

Creating Easy to Follow Code

- In an ideal world, reviewing code would be as simple as reading a fairy tale (“Fairy Tale Programming”)



- Programming is an art form that can be relatively simplistic or fairly complex
- The goal is to **MAXIMIZE** program reviewability

Once Upon A Time...

Once upon a time there was a mother pig who had three little pigs, and as she had not enough to keep them, she sent them out to seek their fortune. The first little pig built a house with straw.

Along came a wolf, and knocked at the door, and said “Little pig, little pig, let me come in.



To which the pig answered, “No, no, not by the hair of my chiny chin chin.”

The wolf answered to that, “Then I’ll huff, and I’ll puff, and I’ll blow your house in.” So he huffed, and he puffed, ...

Create “Fairy Tale” Programming to Maximize Reviewability



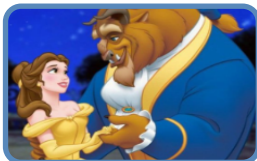
Make code easy to read so it reads like a fairy tale



Know expectations
Standard governance / rules



Make code flow in logical steps



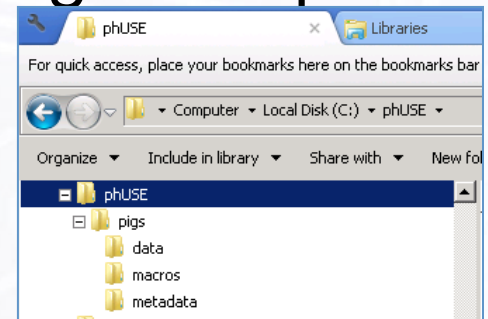
Uses documentation
Comments used throughout



Simplify code whenever possible

Know Expectations

- A place for everything and everything in it's place
 - Make sure you can find data
 - Make sure you can find macros
 - Make sure you can find metadata
- Standard programming requirements
 - Header
 - Spacing
 - Documentation and comments
 - Naming conventions (data sets) and placement of statements (e.g., length statements first in data step)



Pig Data

- Use the pig.sas7bdat data to create the story of the first little pig

Obs	mompig	lilpig	house	wolf
1	sow	first	straw	blew house in
2	sow	second	sticks	blew house in
3	sow	third	bricks	could not blow house in

Mock Listing

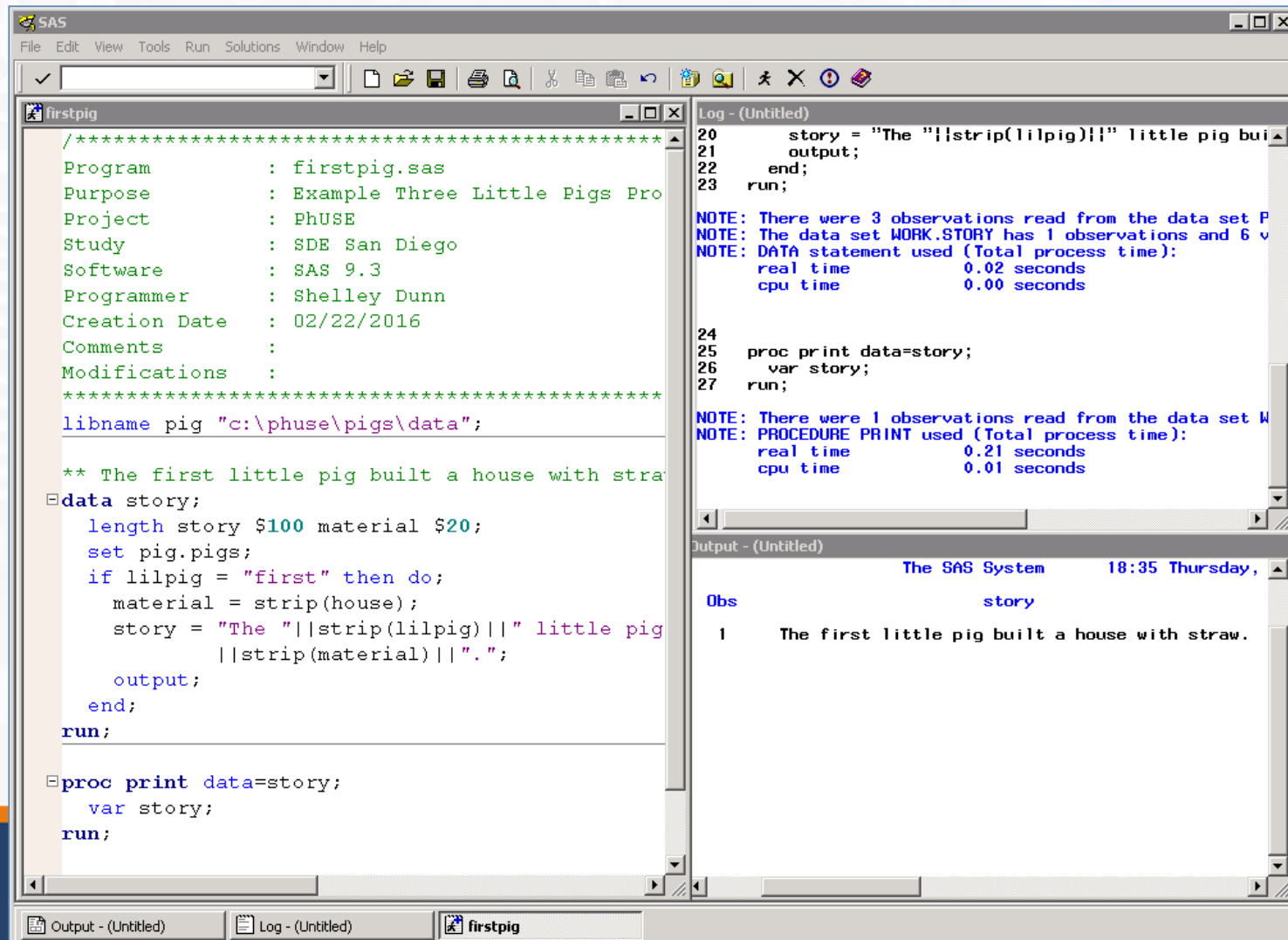
“The first little pig built a house with straw.”



SAS Code – First Lil' Pig

Header

Location of data
Add Comments
Naming Conventions



```

SAS
File Edit View Tools Run Solutions Window Help

firstpig
/*****
Program       : firstpig.sas
Purpose       : Example Three Little Pigs Pro
Project       : PhUSE
Study         : SDE San Diego
Software      : SAS 9.3
Programmer    : Shelley Dunn
Creation Date  : 02/22/2016
Comments      :
Modifications :
*****/
libname pig "c:\phuse\pigs\data";

** The first little pig built a house with straw

data story;
  length story $100 material $20;
  set pig.pigs;
  if lilpig = "first" then do;
    material = strip(house);
    story = "The " || strip(lilpig) || " little pig
           || strip(material) || ".";
    output;
  end;
run;

proc print data=story;
  var story;
run;

```

Log - (Untitled)

```

20      story = "The " || strip(lilpig) || " little pig bui
21      output;
22      end;
23      run;

NOTE: There were 3 observations read from the data set P
NOTE: The data set WORK.STORY has 1 observations and 6 v
NOTE: DATA statement used (Total process time):
      real time           0.02 seconds
      cpu time            0.00 seconds

24
25      proc print data=story;
26      var story;
27      run;

NOTE: There were 1 observations read from the data set W
NOTE: PROCEDURE PRINT used (Total process time):
      real time           0.21 seconds
      cpu time            0.01 seconds

```

Output - (Untitled)

The SAS System		18:35 Thursday,
Obs	story	
1	The first little pig built a house with straw.	

Output - (Untitled) Log - (Untitled) firstpig

Standard Programming Practices

Logical Steps



Examples*:

- Standard and completed header information
- Standard placement of certain statements (e.g., libname at the top of every program)
- Documentation used throughout the program to make it easy to follow
- Standard, easy to understand, naming conventions
- New working data set name for each new data set created
- All macros in a specified location (e.g., folder, top of program)
- All metadata in a specified location
- Standard spacing and indentation used



*Companies to define their own standard programming practices

Consider: Macros, External Files, and Documentation

- When do you create macro code?
 - `%pigs(first, straw); ** pig=first, house=straw **;`
 - `%pigs(second, sticks); ** pig=second, house=sticks**;`
 - `%pigs(third, bricks); ** pig=third, house=bricks **;`
- When to use macro variables? Document what they do!
 - `%let version=3.2; ** pull in SDTM Standard for v3.2 **;`
- Where are external files located? Document this!
 - `C:\phuse\pigs\metadata ** location of SDTM standard metadata **;`
- Documentation and comments are extremely helpful to a reviewer!
- **Make the reviewer's job EASY!!!**



Example Metadata File

- Simplify organization of metadata (not just code)
- Create a custom “Little Pig Events” (LP) SDTM domain to capture the events in The Three Little Pigs

DOMAIN	VARIABLE	ORIGIN	LABEL	CT	ORDER	LENGTH	SAS	DATA	ALGORITHM
LP	STUDYID	CRF Page 1	Study Identifier		1	12	C	text	Copy from DM CRF
LP	DOMAIN	Assigned	Domain Abbreviation		2	4	C	text	Set to LP
LP	USUBJID	Derived	Unique Subject Identifier		3	22	C	text	Copy from DM CRF
LP	SUBJID	CRF Page 1	Subject Identifier for the Study		4	6	C	text	Copy from DM CRF
LP	LPSEQ	Derived	Sequence Number		5	8	N	integer	Sort by key variables and number starting with 1
LP	LPTERM	CRF Pages 3, 6, 9	Reported Term for the Pig Event		6	200	C	text	From page 3 enter PIG variable. From page 6 enter HOUSE variable. From page 9 enter WOLF variable
LP	LPDECOD	CRF Page 3	Standardized Pig Term	PIGTERM	7	100	C	text	Map reported Pig Event to Pig Term based on Controlled Terminology
LP	LPCAT	Assigned	Category for Pig Event	PIGCAT	8	20	C	text	From page 3 enter "Pig Event" From page 6 enter "House Event" From page 9 enter "Wolf Event"
LP	VISITNUM	CRF Pages 3, 6, 9	Visit Number		9	8	N	integer	From Little Pig (LP) CRF
LP	VISIT	CRF Pages 3, 6, 9	Visit Name	VISIT	10	20	C	text	From Little Pig (LP) CRF
LP	EPOCH	Derived	Epoch	EPOCH	11	50	C	text	Derive based on SV
LP	LPSTDTC	CRF Page 3	Start Date/Time of Pig Event		12	20	C	datetime	From Little Pig (LP) CRF
LP	LPSTDY	Derived	Study Day of Start of Pig Event		13	8	N	integer	LP.LPSTDTC -DM.RFSTDTC + 1

Simplify Code if Possible

- Simply is a relative term
- First and foremost, the goal of a program must be met:
 - Accuracy
 - Reusability
 - Produces what it is meant to produce
- From there, how to simplify is a subjective task. Keep your reviewer, not your job security, in mind!

CODELIST	DESCRIPTION	VALUE	DECODE
PIGCAT	Category of The Three Little Pigs Event	Pig Event	
PIGCAT	Category of The Three Little Pigs Event	House Event	
PIGCAT	Category of The Three Little Pigs Event	Wolf Event	
PIGTERM	Standardized The Three Little Pigs Term	First Pig	
PIGTERM	Standardized The Three Little Pigs Term	Second Pig	
PIGTERM	Standardized The Three Little Pigs Term	Third Pig	
PIGTERM	Standardized The Three Little Pigs Term	Straw	
PIGTERM	Standardized The Three Little Pigs Term	Sticks	
PIGTERM	Standardized The Three Little Pigs Term	Bricks	
PIGTERM	Standardized The Three Little Pigs Term	blew house in	
PIGTERM	Standardized The Three Little Pigs Term	could not blow house in	

▶▶ Variable Metadata CT

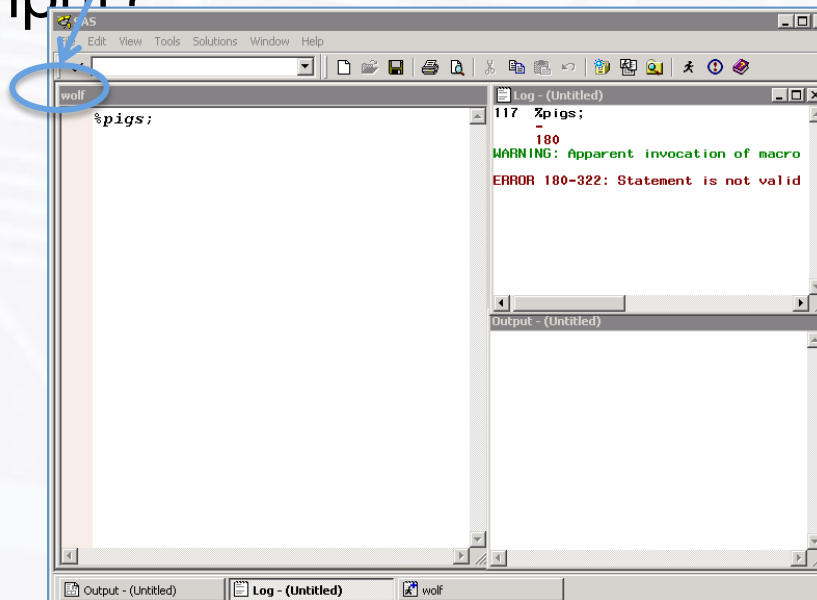
SAS Code Comments:

```
** Read in Pig CT, location c:\phuse\pigs  
\metadata\pigsCT.xlsx **;
```

```
** Create Records for Pigs **;  
** Create Records for Houses **;  
** Create Records for Wolf Outcome **;
```

Where's the Code?

- What if you had to review the following code?
- Why does wolf.sas, below, fail to maximize reviewability?
- Audience input?



wolf.sas (prior slide)

- Code doesn't run
- No documentation
 - No program header to explain the purpose of program
 - Location of code, files, etc. is unknown
- Not easy to read/review (doesn't read like a story)
 - Can't assess logical order
 - Unable to assess simplicity



Modifying Code



Challenging

- Standard practices not followed
- No documentation
- Overly complex code
- Code is difficult to follow, not logical
- Files not easily accessible



Ideal

- Standard practices followed
- Well documented
- Simplify code, when possible
- Code is easy to follow and logical
- Easily accessible files

Summary

- In an idea world, reviewing code would be as simple as reading a fairy tale
- Programs are reviewed by independent programmers for accuracy and efficiency
- “Easy to follow” code leads to better program reviews which leads to better chance of program accuracy
- To maximize program reviewability: know expectations, use logical steps, document code, and simplify code whenever feasible
- To help simplify code determine if and when to use macros and/or how best to access external files
- By maximizing program reviewability *all programmers and reviewers live happily ever after...*

THE END

