What is a Variable?

- A user defined name to represent a piece of memory for storing evaluated value(s). A variable consists of 5 items:
  - **Name**: meaningful human readable name
  - **Data Type**: How to interpret variable for data representation
  - **Size**: How much storage memory is needed to store data value
  - **Value**: Actual value associated with variable stored in memory
  - **Storage location**: Usually hidden from user by the interpreter or compiler

For Our Purposes: Columns

Many variables. A columns of variables

Variable

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Size</th>
<th>Memory Location (hidden from user)</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Radius</td>
<td>float32</td>
<td>4 bytes</td>
<td>0x1800F040</td>
<td>3.23</td>
</tr>
<tr>
<td>currKey</td>
<td>char</td>
<td>1 byte</td>
<td>0x1800F049</td>
<td>'k'</td>
</tr>
<tr>
<td>firstName</td>
<td>string</td>
<td>6 bytes</td>
<td>0x1800B0E0</td>
<td>&quot;morgan&quot;</td>
</tr>
<tr>
<td>width</td>
<td>int32</td>
<td>4 bytes</td>
<td>0x1800CCE8</td>
<td>800</td>
</tr>
<tr>
<td>type</td>
<td>int8</td>
<td>1 byte</td>
<td>0x1800CCE7</td>
<td>27</td>
</tr>
</tbody>
</table>

- var label;
- value label (interpretation)
- SAS: proc contents

Naming Rules

Use Valid Names

- Length: reasonably short (8) but descriptive
- Syntax: similar to userid
  - Starts with a single letter followed by any number of letters, digits, or underscores.
  - Digits [0-9], Letters [a-zA-Z], Underscore `_`
  - Capitalization
    - STATA: differentiate
    - SAS: does not differentiate
    - Best to not use (too confusing for people)
- No spaces allowed
  - `_` or camelCase

Naming Rules, cont

write program for people

- Avoid Keywords (if, else, while, for, ...)
  - Result: Error / confusing
- Use Meaningful names
  - currStudent better than fido, purpleSloth, or currItem
- Write readable names
  - currStudent better than (cS, crSt, or crrStdnt)
- Convention
  - b_: binary (bincome, b_income, bIncome)
  - n_: number (nincome, n_income)
  - c_: string / character (cincome, c_income)
  - g_: groups (gincome)

What is a Data Type?

- How to interpret a storage location to retrieve the correct value.
- Integer, Floating point, Logical, Char, Strings are typical data types
- Other languages require you to explicitly specify the data type of variables
- SAS implicitly infers the data type from the first initialization(use) via the specified expression.
  - Number/Char
  - String static (be careful of values getting cutoff)

Data Types : 8 bits = 1 bytes

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Size (bits)</th>
<th>Min</th>
<th>Max</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>logical</td>
<td>1</td>
<td>0 (false)</td>
<td>1 (true)</td>
<td></td>
</tr>
<tr>
<td>int8</td>
<td>1</td>
<td>-128</td>
<td>+127</td>
<td>Numeric, integer, Exact</td>
</tr>
<tr>
<td>single</td>
<td>4</td>
<td>-3.4028e+038</td>
<td>+3.4028e+038</td>
<td>Numeric Real Approximate</td>
</tr>
<tr>
<td>double</td>
<td>8</td>
<td>-1.7977e+308</td>
<td>+1.7977e+308</td>
<td>Ditto</td>
</tr>
<tr>
<td>char</td>
<td>2</td>
<td>N/A</td>
<td>N/A</td>
<td>Encoded character</td>
</tr>
<tr>
<td>string</td>
<td>Varies len+1</td>
<td>N/A</td>
<td>N/A</td>
<td>String of encoded characters</td>
</tr>
</tbody>
</table>
### ASCII: character encoding

<table>
<thead>
<tr>
<th>ASCII Code</th>
<th>Character</th>
</tr>
</thead>
<tbody>
<tr>
<td>64</td>
<td>A</td>
</tr>
<tr>
<td>97</td>
<td>a</td>
</tr>
<tr>
<td>120</td>
<td>l</td>
</tr>
<tr>
<td>121</td>
<td>m</td>
</tr>
<tr>
<td>123</td>
<td>n</td>
</tr>
</tbody>
</table>

### Variable Types

<table>
<thead>
<tr>
<th>Type</th>
<th>Stored value</th>
<th>Interpreted value</th>
<th>Label</th>
<th>Interpreted Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Int</td>
<td>1000001 (65)</td>
<td>65</td>
<td>65 or older</td>
<td></td>
</tr>
<tr>
<td>Character</td>
<td>1000001 (65)</td>
<td>A</td>
<td>Asian</td>
<td></td>
</tr>
</tbody>
</table>

- 1 0 0 0 0 0 1 = 64 + 1 = 65
- 64 32 16 8 4 2 1

### Lab: Vars

- Number
  - Int (long), real (double, float), date time
- String/Character
  - Length matters
- Missing
  - .
  - ”
  - ‘’
  - SAS: .<0

### Lab 2 & Assignment 2: Objective

- To write conditional logic codes
- Subset columns (variables) from a table
- Subset rows (observations) from a table
- Recode, rename variables and calculate new variables
- Label variables and values

### Recommended Reading

- Carefully read each of the modules below. Each has very good explanations of exactly how to do certain things.
  - [http://www.ats.ucla.edu/stat/sas/modules/vars.htm](http://www.ats.ucla.edu/stat/sas/modules/vars.htm)
  - [http://www.ats.ucla.edu/stat/sas/modules/subset.htm](http://www.ats.ucla.edu/stat/sas/modules/subset.htm)
  - [http://www.ats.ucla.edu/stat/sas/modules/missing.htm](http://www.ats.ucla.edu/stat/sas/modules/missing.htm)
  - [http://www.ats.ucla.edu/stat/sas/modules/labels.htm](http://www.ats.ucla.edu/stat/sas/modules/labels.htm)
- Little SAS book
  - Sections in Chapter 3
Data Step

libname data "D:\HPM-Users\kum\phpm672lab2\data";
data outf;
set infn;
...code...
data mynsduh;
set data.nsdhu;
...code...

Subset columns (variables)

- SAS
  - Three places possible
  - Reading in, writing out, during datastep
  - keep, drop
    ```
    data mynsduh;
    set data.nsdhu (keep=var);
    ```
    ```
    data mynsduh;
    set data.nsdhu (drop=var);
    ```

Subset rows (observations)

- SAS
  - `where cond`;
  - `if cond`:

Calculate new variable (assignment)

- SAS (in data step)
  - `var1 = 1 ; * assignment;`
  - `num1=.; * numeric missing value;`
  - `str1="" ; * string missing value;`

Recode existing variables

- SAS (in data step)
  - No difference between existing/new
  - Use if/then/else to conditionally recode
    ```
    var1 = 3 ; * assignment new value;
    ```
    ```
    * One way:
    if race='Asian' then race='Other';
    else if race='Native' then race='Other';
    if race in (1, 2, ..., 'Asian') then a_race='Other';
    ```
    ```
    * Another way:
    if race in ('Asian', 'Native') then race='Other';
    ```
1/30/2018

Rename existing variable
- SAS (in data step)
  - Depending on where you do this, different behavior
  - `rename oldvar=newvar`

To write conditional logic codes
- SAS
  - `if cond then [do:] ...; [end:]`
  - `where cond ;`

Swap x1 & x2
- Write the code in SAS

Label variables
- SAS
  - `label var1 = "LABEL" ;`

Label values
- SAS: define format, then use in data step

```
proc format:
  value fname
  val1= "LAB1"
  val2= "LAB2" ;
* inside data step:
  format var1 fname.
```

Label Var vs Value

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Size</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>bcigever</td>
<td>int8</td>
<td>1 byte</td>
<td>1 or 0</td>
</tr>
</tbody>
</table>

- Labeling variable
  - Give a more human friendly name to the variable name.
  - Same as `bcigever` (the computer friendly name for the variable used in the programs)
  - Stored in the header information for the table
### Label Var vs Value

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Size</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>bcigever</td>
<td>int8</td>
<td>byte</td>
<td>1 or 0</td>
</tr>
</tbody>
</table>

- **labeling value**
  - Give a more human friendly name to the variable value.
  - Same as \(1(=\text{TRUE})\) or \(0(=\text{FALSE})\)
  - Internally, the computer stores 0 or 1
  - But, when printing the values for humans, the computer uses the format you created and designated to use for this variable.
  - Can be used on multiple variables
  - It can be permanent (if done in the data step) or temporary (if done in proc steps)
  - The format must be created BEFORE use
  - Stored in the header information for the table

### Type of variables (from analysis perspective)

- **Var Types**
  - Continuous (discrete is continuous in computers)
  - Categorical
  - Boolean
  - ID: no other information but to link tables together. i.e. random patient ID used in two tables.
  - Helps you starting thinking about what you can do with the information
  - Not all variables types exist in datasets.
  - Just state NA.

### Basic descriptive analysis

- **Numerical**
  - \(N, \text{ mean, max, min, std dev, unique values (mode)}\)
  - SAS: `proc means`
- **Categorical**
  - Frequencies, cross tabulation
  - SAS: `proc freq`:
    - `tables var1list/nocol norow nopercent;`
    - `tables var1*var2/nocol norow nopercent;`

### Reminder

- Make sure to understand lab 2
  - Try to finish by Tues. So you can spend next week doing assignment 2
  - BUT submit with assignment 2 the week after.
  - You MUST submit programs, logs, and output along with assignment 2
  - This is how you will LEARN
  - Most IMPORTANT part of class
- **Dataset(s) you want to use through out the class**
  - Flu dataset
  - Texas Inpatient Public Use Data File (PUDF)
    - [http://www.dshs.state.tx.us/thic/hospitals/inpatientpudf.shtml](http://www.dshs.state.tx.us/thic/hospitals/inpatientpudf.shtml)

### Assignment 1

- How was it?
  - Output extension wrong
    - `txt`
    - `lst`, but log file
    - No mean of all states
    - Output different