

SAS: READING IN FILES 2

- Review of DATA step and some more
- Review of PROC step and some more
- Start univariate statistics

Reference: slides by Professor Papachristou (private communication).

REVIEW OF DATA STEP

```
data grades;  
  infile datalines dsd;  
  input  
    subject  
    gender $  
    exam1  
    exam2  
  ;  
datalines;  
10, M, 80, 84  
 7, M, 85, 89  
 4, F, 90, 86  
20, M, 82, 85  
14, F, 88, .  
.
```

Data manipulation occurs after input...

Total rows: 5 Total columns: 4				
	subject	gender	exam1	exam2
1	10	M	80	84
2	7	M	85	89
3	4	F	90	86
4	20	M	82	85
5	14	F	88	.

Keep/drop columns ($\approx$ select)

```
data grades;
  infile datalines dsd;
  input subject gender $ exam1 exam2;
  drop subject; /* keep/drop columns */
datalines;
10, M, 80, 84
7, M, 85, 89
4, F, 90, 86
20, M, 82, 85
14, F, 88, .
;
run;
```

Total rows: 5 Total columns: 3		Rows 1-5	
	gender	exam1	exam2
1	M	80	84
2	M	85	89
3	F	90	86
4	M	82	85
5	F	88	.

Write columns ($\approx$ mutate)

```
data grades;  
  infile datalines dsd;  
  input subject gender $ exam1 exam2;  
  examavg = (exam1+exam2)/2;  
datalines;  
10, M, 80, 84  
7, M, 85, 89  
4, F, 90, 86  
20, M, 82, 85  
14, F, 88, .  
;  
run;
```

Total rows: 5 Total columns: 5

	subject	gender	exam1	exam2	examavg
1	10	M	80	84	82
2	7	M	85	89	87
3	4	F	90	86	88
4	20	M	82	85	83.5
5	14	F	88	.	.

Write columns ($\approx$ mutate)

```
data grades;
  infile datalines dsd;
  input subject gender $ exam1 exam2;
  examavg = mean(exam1, exam2);
datalines;
10, M, 80, 84
7, M, 85, 89
4, F, 90, 86
20, M, 82, 85
14, F, 88, .
;
run;
```

Total rows: 5 Total columns: 5

	subject	gender	exam1	exam2	examavg
1	10	M	80	84	82
2	7	M	85	89	87
3	4	F	90	86	88
4	20	M	82	85	83.5
5	14	F	88	.	88

Facts when creating new variables:

1. New variables are initialized to missing
2. Missing values are $<$ any value.
3. Reference missing numeric values by `.`
4. Reference missing character values by `' '` (with or without space).

Aggregation functions like `mean()`, `sum()`, `n()` ignore missing values by default.

More math operations and functions:

- Exponentiation x^y is written `x**y`
- `abs()`, `log()`, `log10()`, `sqrt()`,
- `round( , n)` rounds to nearest multiple of `n` (includes decimals)

Write columns with if ... else

```
data grades;
  infile datalines dsd;
  input subject gender $ exam1 exam2;
  if exam2 >= 90 then grade = 'A';
  else if exam2 >= 80 then grade = 'B';
  else if exam2 >= 70 then grade = 'C';
datalines;
10, M, 80, 84
7, M, 85, 89
4, F, 90, 66
20, M, 82, 75
14, F, 88, .
;
run;
```

	subject	gender	exam1	exam2	grade
1	10	M	80	84	B
2	7	M	85	89	B
3	4	F	90	66	
4	20	M	82	75	C
5	14	F	88	.	

```

data grades;
  infile datalines dsd;
  input subject gender $ exam1 exam2;
  if exam2 >= 90 then grade = 'A';
  else if exam2 >= 80 then grade = 'B';
  else if exam2 >= 70 then grade = 'C';
  else if exam2 < 60 then grade = 'F';
datalines;
10, M, 80, 84
7, M, 85, 89
4, F, 90, 66
20, M, 82, 75
14, F, 88, .
;
run;

```

NA is less than 60 in SAS, so row 5 gets an F grade.

	subject	gender	exam1	exam2	grade
	10	M	80	84	B
	7	M	85	89	B
	4	F	90	66	
	20	M	82	75	C
	14	F	88	.	C

Write columns with if ... else

```
data grades;
  infile datalines dsd;
  input subject gender $ exam1 exam2;
  if exam2 >= 90 then grade = 'A';
  else if missing(exam2) then grade = 'Missing';
  else if exam2 >= 80 then grade = 'B';
  else if exam2 >= 70 then grade = 'C';
  else if exam2 < 60 then grade = 'F';
datalines;
10, M, 80, 84
7, M, 85, 89
4, F, 90, 66
20, M, 82, 75
14, F, 88, .
.
```

Total rows: 5 Total columns: 5						
	subject	gender	exam1	exam2	grade	
1	10	M	80	84	B	
2	7	M	85	89	B	
3	4	F	90	66		
4	20	M	82	75	C	
5	14	F	88	.	M	

(Fix truncated characters by writing in more spaces...)

```
data grades;
  infile datalines dsd;
  input subject gender $ exam1 exam2;
  if exam2 >= 90 then grade = 'A';
  else if missing(exam2) then grade = 'Missing';
  else if exam2 >= 80 then grade = 'B';
  else if exam2 >= 70 then grade = 'C';
  else if exam2 < 60 then grade = 'F';
datalines;
10, M, 80, 84
7, M, 85, 89
4, F, 90, 66
20, M, 82, 75
14, F, 88, .
.
```

(Or declare number of characters needed by the grade column)

```
data grades;
  infile datalines dsd;
  input subject gender $ exam1 exam2;
  length grade $7;
  if exam2 >= 90 then grade = 'A';
  else if missing(exam2) then grade = 'Missing';
  else if exam2 >= 80 then grade = ' B';
  else if exam2 >= 70 then grade = 'C';
  else if exam2 < 60 then grade = 'F';
datalines;
10, M, 80, 84
7, M, 85, 89
4, F, 90, 66
20, M, 82, 75
14, F, 88,
```

Total rows: 5 Total columns: 5

⏪ ⏩ Rows 1-5 ⏪ ⏩

	subject	gender	exam1	exam2	grade
1	10	M	80	84	B
2	7	M	85	89	B
3	4	F	90	66	
4	20	M	82	75	C
5	14	F	88	.	Missing

Filter rows with WHERE

```
data grades;
  infile datalines dsd;
  input subject gender $ exam1 exam2;
datalines;
10, M, 80, 84
7, M, 85, 89
4, F, 90, 66
20, M, 82, 75
14, F, 88, .
;
run;
data goodgrades;
  set grades;
  where exam1 > 80 and exam2 > 80;
run;
```

Table:

WORK.GOODGRADES

View:

Column names

Filter: (none)

Columns

☒

Select all

☒

123 subject

Total rows: 1

Total columns: 4

	subject	gender	exam1	exam2
1	7	M	85	89

Filter rows with WHERE

```
data grades;
  infile datalines dsd;
  input subject gender $ exam1 exam2;
datalines;
10, M, 80, 84
7, M, 85, 89
4, F, 90, 66
20, M, 82, 75
14, F, 88, .
;
run;
data goodgrades;
  set grades;
  where exam1 > 80 and exam2 > 80;
run;
```

- set helps read in existing SAS dataset for further DATA step manipulation.
- where step could not run before data was read in, so do above or replace with if statement.

REVIEW OF PROC STEP

Common PROCs

- PRINT —
- SORT —
- MEANS — Continuous variables
- FREQ — categorical variables: freq. tables, χ^2 -test
- UNIVARIATE — Continuous variables
- GLM — Linear model (later)

Syntax for procedures:

```
PROC PROCNAME DATA=datasetname <options> ;  
    substatements/<options> ;  
RUN;
```

Common substatements in PROC steps:

- LABEL relabels column names
- FORMAT changes table entry format
- VAR selects variables to analyze
- WHERE filters input dataset
- BY process data in separate groups, (fails on unsorted groups so use PROC SORT first)
- CLASS process data in separate groups (works on fewer PROC than BY)

Read in and store SAS dataset wbc (white blood cell):

```
filename myurl url "https://manctsui.github.io/datasets/WBC.xlsx";
PROC IMPORT
    DATAFILE=myurl
        OUT = wbc
        DBMS = xlsx
    REPLACE ;
    GETNAMES = yes;
RUN;
```

Total rows: 12 Total columns: 12

⏪ ⏩ Rows 1-12 ⏪ ⏩

	Name	State	Bday ▲	Hght	Wght	Sex	SexC	Educ	WBC1	WBC2	WBC3
1	Al	NJ	08/18/57	61	190	0	M	7	12.5	10.3	14.5
2	Jill	NJ	10/07/60	61	135	1	F	2	10.0	13.2	13.8
3	Anna	PA	05/26/62	59	119	1	F	6	13.9	12.5	.
4	Ty	DE	07/06/72	67	142	0	M	.	14.2	12.5	11.4
5	Mary	PA	09/03/74	70	140	1	F	8	10.5	14.5	12.7
6	Ruth	DE	10/07/75	60	110	1	F	3	12.0	13.8	13.2
7	Ada	NJ	11/28/79	63	120	1	F	3	14.6	10.7	14.7
8	Joe	PA	11/10/93	72	180	0	M	1	11.4	13.1	11.8
9	Tim	PA	08/19/99	68	225	0	M	2	11.1	10.4	14.1
10	Kate	DE	06/20/00	65	147	1	F	6	12.7	12.3	13.3
11	Jim	NJ	07/22/01	79	184	0	M	6	14.8	.	14.1
12	Moe	NJ	08/12/02	70	300	0	M	2	14.7	10.1	11.8

Here left-aligned numbers mean the numeric data are read as character strings, so I fixed the xlsx file's "." to empty cell and read it in again.

Obs	Name	State	Bday	Hght	Wght	Sex	SexC	Educ	WBC1	WBC2	WBC3	WBC4
1	Ruth	DE	10/07/75	60	110	1	F	3	12.0	13.8	13.2	11.2
2	Jim	NJ	07/22/01	79	184	0	M	6	14.8	.	14.1	10.9
3	Mary	PA	09/03/74	70	140	1	F	8	10.5	14.5	12.7	14.6
4	Ty	DE	07/06/72	67	142	0	M	.	14.2	12.5	11.4	12.5
5	Moe	NJ	08/12/02	70	300	0	M	2	14.7	10.1	11.8	14.7
6	Ada	NJ	11/28/79	63	120	1	F	3	14.6	10.7	14.7	13.4
7	Al	NJ	08/18/57	61	190	0	M	7	12.5	10.3	14.5	14.9
8	Jill	NJ	10/07/60	61	135	1	F	2	10.0	13.2	13.8	11.2
9	Kate	DE	06/20/00	65	147	1	F	6	12.7	12.3	13.3	15.0
10	Tim	PA	08/19/99	68	225	0	M	2	11.1	10.4	14.1	13.5
11	Joe	PA	11/10/93	72	180	0	M	1	11.4	13.1	11.8	10.3
12	Anna	PA	05/26/62	59	119	1	F	6	13.9	12.5	.	12.7

```
proc print data = wbc(obs=5);  
run;
```

Parenthesis lets us do some DATA step options in PROC step.

Obs	Name	State	Bday	Hght	Wght	Sex	SexC	Educ	WBC1	WBC2	WBC3	WBC4
1	Ruth	DE	10/07/75	60	110	1	F	3	12.0	13.8	13.2	11.2
2	Jim	NJ	07/22/01	79	184	0	M	6	14.8	.	14.1	10.9
3	Mary	PA	09/03/74	70	140	1	F	8	10.5	14.5	12.7	14.6
4	Ty	DE	07/06/72	67	142	0	M	.	14.2	12.5	11.4	12.5
5	Moe	NJ	08/12/02	70	300	0	M	2	14.7	10.1	11.8	14.7

```
proc print data = wbc(obs=5);  
    where Wght > 150;  
run;
```

where filters rows by given condition

Obs	Name	State	Bday	Hght	Wght	Sex	SexC	Educ	WBC1	WBC2	WBC3	WBC4
2	Jim	NJ	07/22/01	79	184	0	M	6	14.8	.	14.1	10.9
5	Moe	NJ	08/12/02	70	300	0	M	2	14.7	10.1	11.8	14.7
7	Al	NJ	08/18/57	61	190	0	M	7	12.5	10.3	14.5	14.9
10	Tim	PA	08/19/99	68	225	0	M	2	11.1	10.4	14.1	13.5
11	Joe	PA	11/10/93	72	180	0	M	1	11.4	13.1	11.8	10.3

```
proc print data = wbc(obs=5);  
  where Wght > 150;  
  var Name State Wght;  
run;
```

where filters rows; var selects columns.

Obs	Name	State	Wght
2	Jim	NJ	184
5	Moe	NJ	300
7	Al	NJ	190
10	Tim	PA	225
11	Joe	PA	180

```
proc print data = wbc(obs=5);  
  where Wght > 150;  
  var Name State Wght;  
  by State;  
run;
```

where filters rows; var selects columns; by groupings.

State=NJ			
Obs	Name	State	Wght
2	Jim	NJ	184
5	Moe	NJ	300
7	Al	NJ	190

State=PA			
Obs	Name	State	Wght
10	Tim	PA	225
11	Joe	PA	180

```
proc print label data = wbc(obs=5);
  title 'WBC dataset';
  label Wght = "Weight"
        Bday = "Birthday";
run;
```

label changes printed variable names

Obs	Name	State	Birthday	Hght	Weight	Sex	SexC	Educ	WBC1	WBC2	WBC3	WBC4
1	Ruth	DE	10/07/75	60	110	1	F	3	12.0	13.8	13.2	11.2
2	Jim	NJ	07/22/01	79	184	0	M	6	14.8	.	14.1	10.9
3	Mary	PA	09/03/74	70	140	1	F	8	10.5	14.5	12.7	14.6
4	Ty	DE	07/08/72	67	142	0	M	.	14.2	12.5	11.4	12.5
5	Moe	NJ	08/12/02	70	300	0	M	2	14.7	10.1	11.8	14.7

PROC FORMAT creates custom formats

```
proc format;  
    /* Start categorical variables with $. */  
    value $stateF 'PA' = 'Pennsylvania'  
                  'DE' = 'Delaware'  
                  'NJ' = 'New Jersey';  
    /* Good practice to end format names in F. */  
    value genderF 0 = 'Male' 1 = 'Female';  
run;  
  
PROC print wbc;  
    /* Format, when used, end with dot. */  
    format state $stateF.;  
run;
```

	Name	State	Bday	Hght	Wght	Sex	SexC	Educ	WBC1	WBC2	WBC3	WBC4
1	Ruth	Delaware	10/07/75	60	110	1	F	3	12.0	13.8	13.2	11.2
2	Jim	New Jersey	07/22/01	79	184	0	M	6	14.8	.	14.1	10.9
3	Mary	Pennsylvania	09/03/74	70	140	1	F	8	10.5	14.5	12.7	14.6
4	Ty	Delaware	07/06/72	67	142	0	M	.	14.2	12.5	11.4	12.5
5	Moe	New Jersey	08/12/02	70	300	0	M	2	14.7	10.1	11.8	14.7
6	Ada	New Jersey	11/28/79	63	120	1	F	3	14.6	10.7	14.7	13.4
7	Al	New Jersey	08/18/57	61	190	0	M	7	12.5	10.3	14.5	14.9
8	Jill	New Jersey	10/07/60	61	135	1	F	2	10.0	13.2	13.8	11.2
9	Kate	Delaware	06/20/00	65	147	1	F	6	12.7	12.3	13.3	15.0
10	Tim	Pennsylvania	08/19/99	68	225	0	M	2	11.1	10.4	14.1	13.5
11	Joe	Pennsylvania	11/10/93	72	180	0	M	1	11.4	13.1	11.8	10.3
12	Anna	Pennsylvania	05/26/62	59	119	1	F	6	13.9	12.5	.	12.7

PROC FORMAT creates custom formats

```
proc format;
  value $stateF 'PA' = 'Pennsylvania'
               'DE' = 'Delaware'
               'NJ' = 'New Jersey';
  value genderF 0 = 'Male' 1 = 'Female';
run;

/* `set` reads in wbc but outputs
   `DATA wbc` writes out to wbc.
   So the format is now applied to the file wbc */
DATA wbc;
  set wbc
  format state $stateF.;
run;
```

PROC MEAN WITH OPTIONS

```
proc means data=wbc;  
run;
```

The MEANS Procedure						
Variable	Label	N	Mean	Std Dev	Minimum	Maximum
Bday	Bday	12	7967.75	6244.69	-866.000000	15564.00
Hght	Hght	12	66.250000	5.925676	59.000000	79.000000
Wght	Wght	12	166.000000	54.412565	110.000000	300.000000
Sex	Sex	12	0.500000	0.522233	0	1.000000
Educ	Educ	11	4.181818	2.442055	1.000000	8.000000
WBC1	WBC1	12	12.700000	1.722049	10.000000	14.800000
WBC2	WBC2	11	12.127272	1.526493	10.100000	14.500000
WBC3	WBC3	11	13.218181	1.154831	11.400000	14.700000
WBC4	WBC4	12	12.908333	1.702649	10.300000	15.000000

```
proc means data=wbc n mean std maxdec=2;  
    var wght sex wbc1 wbc2  
run;
```

The MEANS Procedure				
Variable	Label	N	Mean	Std Dev
Wght	Wght	12	166.00	54.41
Sex	Sex	12	0.50	0.52
WBC1	WBC1	12	12.70	1.72
WBC2	WBC2	11	12.13	1.53

```
proc means data=wbc n mean std maxdec=2;
  var wght wbc1 wbc2;
  class sexC;
run;
```

The MEANS Procedure						
SexC	N Obs	Variable	Label	N	Mean	Std Dev
F	6	Wght	Wght	6	128.50	14.29
		WBC1	WBC1	6	12.28	1.82
		WBC2	WBC2	6	12.83	1.33
M	6	Wght	Wght	6	203.50	54.17
		WBC1	WBC1	6	13.12	1.67
		WBC2	WBC2	5	11.28	1.41

PROC FREQ WITH OPTIONS

PROC FREQ creates tables. Here by freqs: sex, state

```
PROC FREQ DATA=wbcc;  
    TABLES sex state;  
RUN;
```

The FREQ Procedure				
Sex				
Sex	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	6	50.00	6	50.00
1	6	50.00	12	100.00
State				
State	Frequency	Percent	Cumulative Frequency	Cumulative Percent
DE	3	25.00	3	25.00
NJ	5	41.67	8	66.67
PA	4	33.33	12	100.00

PROC FREQ for frequency tables

```
PROC FREQ DATA=wbc;  
  TABLES clinic / nocum; /* Omit cumulative freq.*/  
  TITLE 'Frequency Distribution of Clinical Center and Gender'  
RUN;
```

Clinical Center		
clinic	Frequency	Percent
Birmingham	18	18
Chicago	29	29
Minneapolis	36	36
Pittsburgh	17	17

PROC FREQ: 2-way frequency table

```
PROC FREQ DATA=wb;
TABLES sex*state; /* Use asterisk. */
TITLE 'Cross Tabulation of Sex and State';
RUN;
```

Cross Tabulation of Sex and State				
The FREQ Procedure				
Frequency Percent Row Pct Col Pct	Table of Sex by State			
	State(State)			
Sex(Sex)	DE	NJ	PA	Total
0	1	3	2	6
	8.33	25.00	16.67	50.00
	16.67	50.00	33.33	
	33.33	60.00	50.00	
1	2	2	2	6
	16.67	16.67	16.67	50.00
	33.33	33.33	33.33	
	66.67	40.00	50.00	
Total	3	5	4	12
	25.00	41.67	33.33	100.00

PROC FREQ: 2-way frequency table in long format

```
PROC FREQ DATA=wb;
TABLES sex*state / LIST;
TITLE 'Cross Tabulation of Sex and State';
RUN;
```

Cross Tabulation of Sex and State					
The FREQ Procedure					
Sex	State	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	DE	1	8.33	1	8.33
0	NJ	3	25.00	4	33.33
0	PA	2	16.67	6	50.00
1	DE	2	16.67	8	66.67
1	NJ	2	16.67	10	83.33
1	PA	2	16.67	12	100.00

PROC FREQ: 2-way frequency table with no row and no col frequencies

```
PROC FREQ DATA=wbcc;  
TABLES sex*state / NOROW NOCOL;  
TITLE 'Cross Tabulation of Sex and State';  
RUN;
```

Cross Tabulation of Sex and State					
The FREQ Procedure					
Frequency Percent	Table of Sex by State				
	Sex(Sex)	State(State)			
		DE	NJ	PA	Total
	0	1 8.33	3 25.00	2 16.67	6 50.00
	1	2 16.67	2 16.67	2 16.67	6 50.00
Total	3 25.00	5 41.67	4 33.33	12 100.00	

PROC FREQ: 2-way frequency table, with chi-squared test performed.

```
PROC FREQ DATA=wbcc;  
TABLES sex*state / CHISQ;  
TITLE 'Cross Tabulation of Sex and State';  
RUN;
```

Cross Tabulation of Sex and State

The FREQ Procedure

Frequency Percent Row Pct Col Pct	Table of Sex by State				
	Sex(Sex)	State(State)			
		DE	NJ	PA	Total
0	1	3	2	6	
	8.33	25.00	16.67	50.00	
	16.67	50.00	33.33		
	33.33	60.00	50.00		
1	2	2	2	6	
	16.67	16.67	16.67	50.00	
	33.33	33.33	33.33		
	66.67	40.00	50.00		
Total	3	5	4	12	
	25.00	41.67	33.33	100.00	

Statistics for Table of Sex by State

Statistic	DF	Value	Prob
Chi-Square	2	0.5333	0.7659
Likelihood Ratio Chi-Square	2	0.5412	0.7629
Mantel-Haenszel Chi-Square	1	0.1325	0.7158
Phi Coefficient		0.2108	
Contingency Coefficient		0.2063	
Cramer's V		0.2108	
WARNING: 100% of the cells have expected counts less than 5. Chi-Square may not be a valid test.			

Sample Size = 12

PROC FREQ: OTHER USEFUL OPTIONS

- CHISQ performs χ^2 -test for 2-way tables.
- MISSING includes missing data as separate category
- LIST makes condensed table

Optional options:

- NOCOL omits column %
- NOROW omits row %
- NOCUM omits cumulative frequencies and %
- NOFREQ omits frequencies
- NOPERCENT omits %