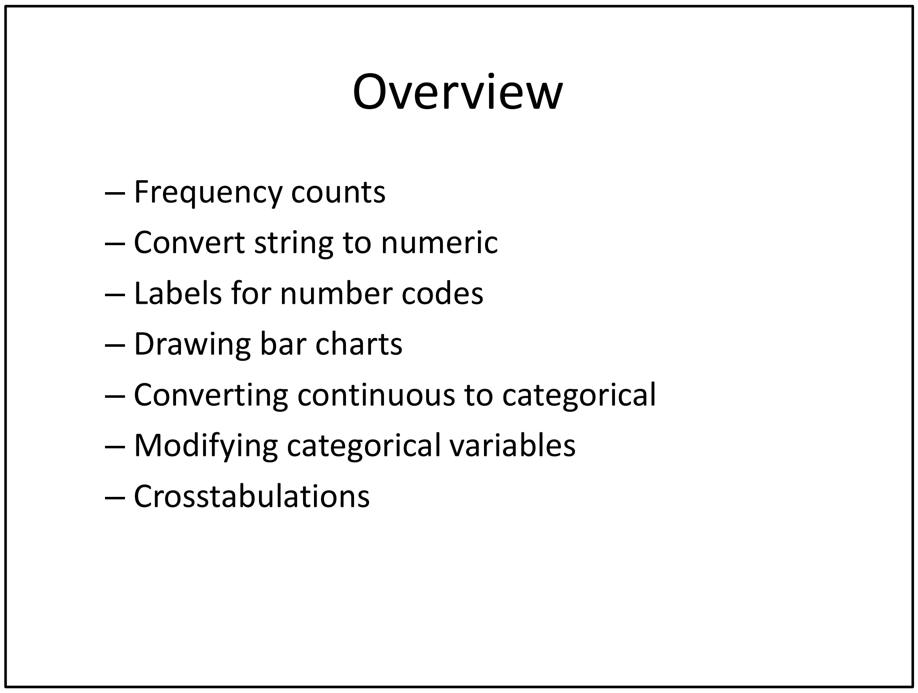
Introduction to SAS, Working with categorical variables

Steve Simon

Created: 2021-05-30



Overview

– Frequency counts
– Convert string to numeric
– Labels for number codes
– Drawing bar charts
– Converting continuous to categorical
– Modifying categorical variables
– Crosstabulations

author: Steve Simon date: created 2021-05-30 purpose: to produce slides for module04 videos license: public domain

Here is an overview of what I want to cover in this module.

We're going to look at a different data set, one with mostly categorical variables. I'll introduce proc format, which allows you to attach labels to categorical data, talk about recoding, and show some tables using proc freq. I'll also show you a simple bar chart.

[For my own use]General structure of this program.

Part00. Documentation header

Notes00. This is the standard documentation header;

Part01. Tell SAS where to find and store things;

This needed to have the output fit on PowerPoint;

Notes01. The filename statement tells you where

Part02. Reading using proc import;

Notes02. As a general rule, proc import works

Part03. Print the first ten lines;

Notes03. It’s always a good idea to peek at the

Output, page 1. If you look at the first few

Part04. Counts, proc freq;

Notes04. For any categorical variables, your

Output, page 2. There is a mix of three passenger

Output, page 3. The survived variable is a number

Part05. Convert string to numeric, data step;

Notes05. For the one continuous variable (age)

Output, page 4. The numeric variable, age_c, can

Part06. Using proc format to code categorical data;

Notes06. For variables like Survived which are

Output, page 5. Notice that the format statement

Part07. Bar charts, proc sgplot (1/3);

Notes07. I don’t normally like bar charts, but

Output, page 6. This is a basic bar chart.;

Part08. Bar charts, proc sgplot (2/3);

Notes08. Getting percentages is a bit tricky.

Output, page 7. You don’t need to print out this

Part09. Bar charts, proc sgplot (3/3);

Notes09. You use this newly created dataset

Output, page 8. Note that the yaxis max=100

Part10. Converting continuous to categorical (1/5);

Notes10. If you want to create categories from a

Part11. Converting continuous to categorical (2/5);

Notes11. Always cross check your results against the original variable.

Output, page 9. These results look good.

Part12. Converting continuous to categorical (3/5);

Notes12. You can control the order by using

Part13. Converting continuous to categorical (4/5);

Notes13. The format statement attaches a label

Part14. Converting continuous to categorical (5/5);

Notes14. Again, a quality check is important.

Output, page 12. The tables are ordered from

Output, page 13. to teenager to adult,

Output, page 14. then missing.;

Part15. Modifying a categorical variable;

Notes15. Here’s how you combine categories for

Output, page 15. Here is a quality check.

Part16. Crosstabulation (1/4);

Notes16. To examine relationships among

Output, page 16. Notice that SAS provides three

Part17. Crosstabulation (2/4);

Notes17. You get row percentages by excluding

Output, page 17. Notice that among the men, 83%

Part18. Crosstabulation (3/4);

Notes18. You get column percentages by excluding

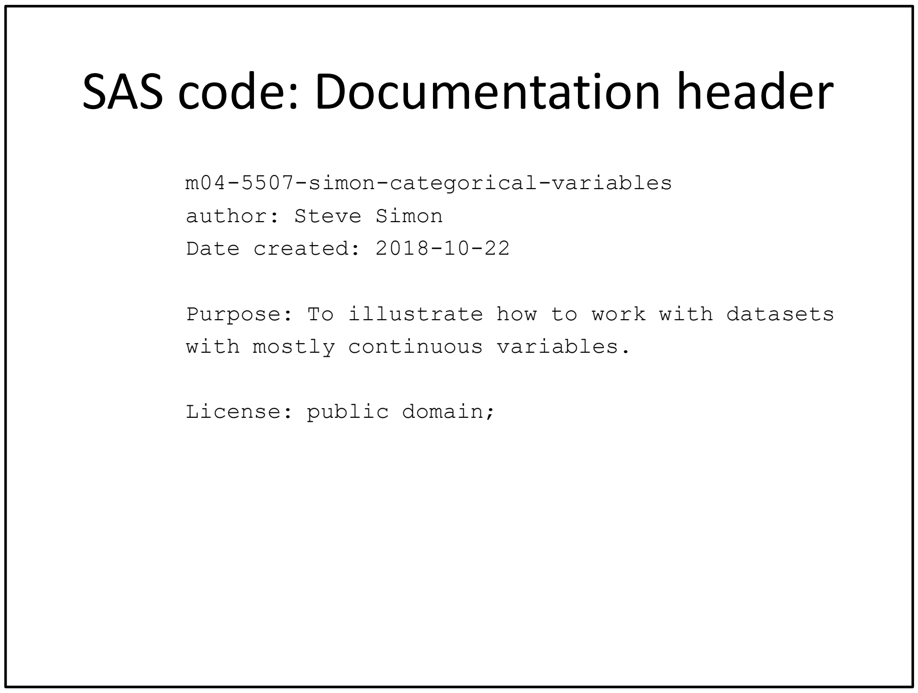
Output, page 18. Column percentages add up to 100%

Part19. Crosstabulation (4/4);

Notes19. You get cell percentages by excluding

Output, page 19. Cell percentages add up to 100%

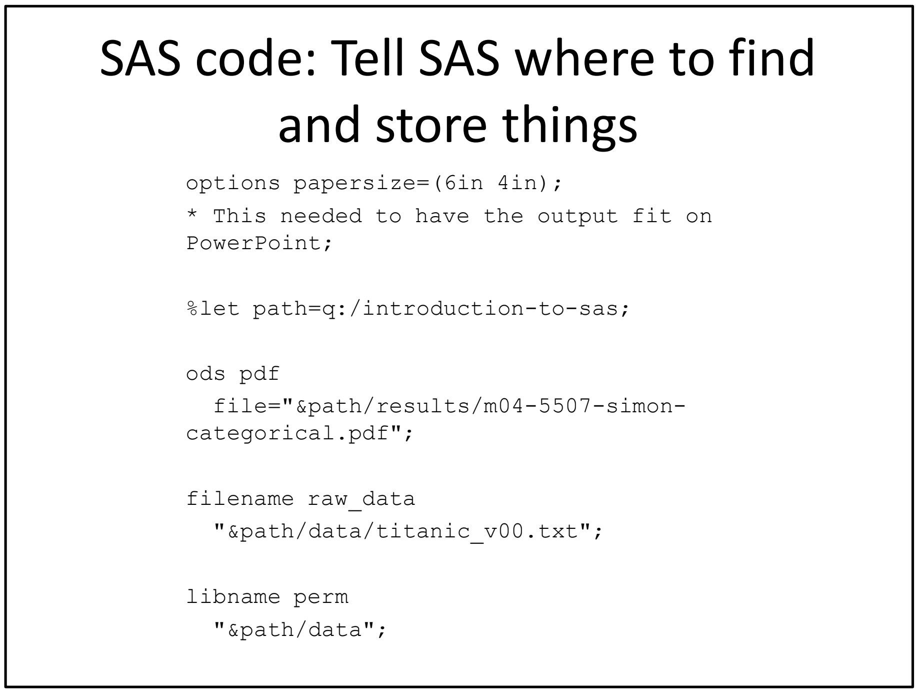
Part20. End of program;



SAS code: Documentation header

```
m04-5507-simon-categorical-variables
author: Steve Simon
Date created: 2018-10-22

Purpose: To illustrate how to work with datasets
with mostly continuous variables.

License: public domain;
```

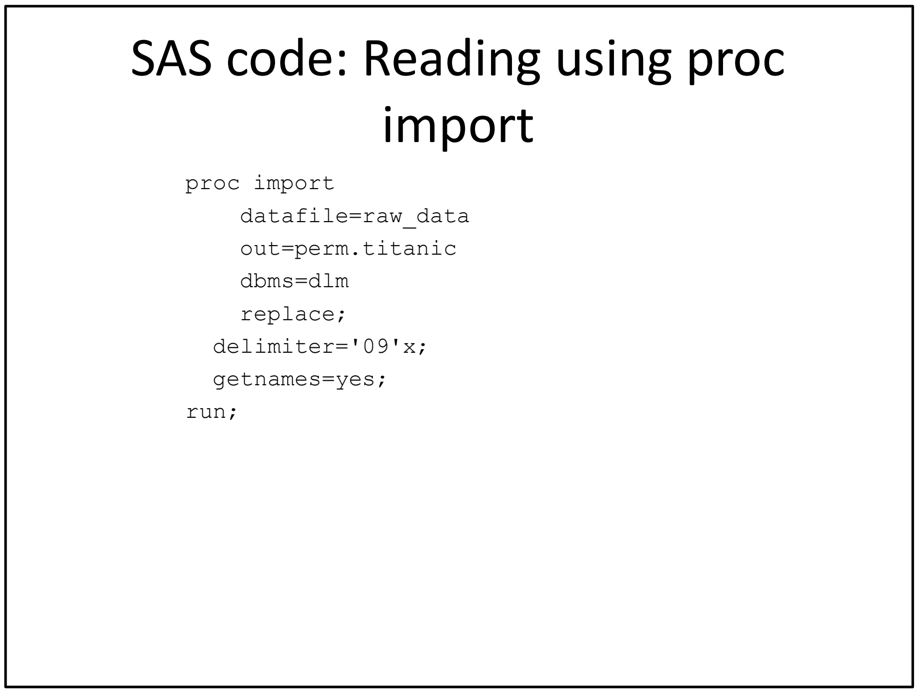
Notes00. This is the standard documentation header



SAS code: Tell SAS where to find and store things

```
options papersize=(6in 4in);
* This needed to have the output fit on
PowerPoint;

%let path=q:/introduction-to-sas;

ods pdf
  file="&path/results/m04-5507-simon-
categorical.pdf";

filename raw_data
  "&path/data/titanic_v00.txt";

libname perm
  "&path/data";
```

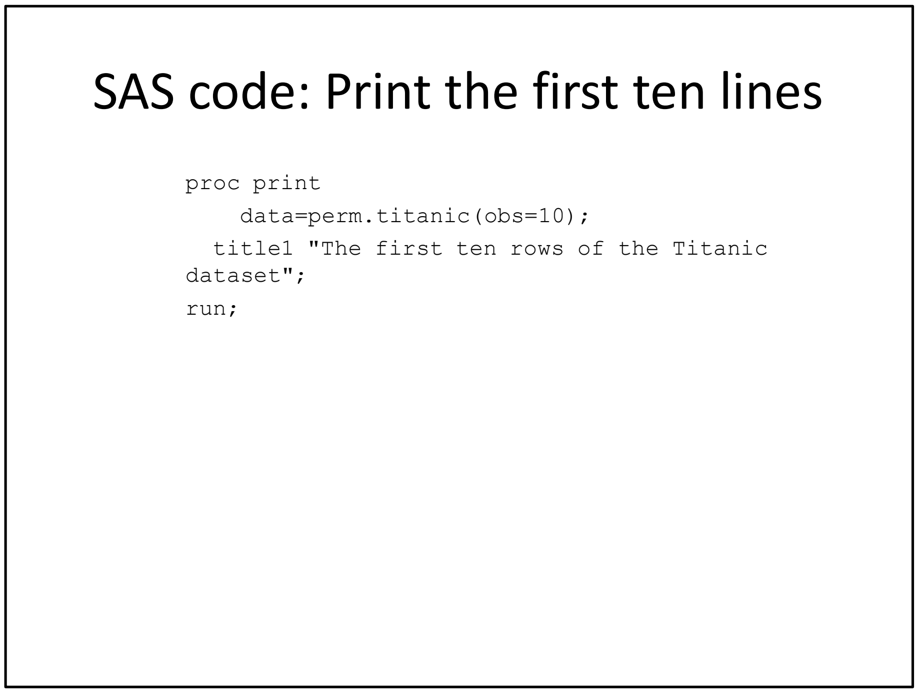
Notes01. The filename statement tells you where the raw data is stored. The libname statement tells you where SAS will store any permanent datsets. The ods statement tells you that SAS is going to store the results with a particular filename and in pdf format.



SAS code: Reading using proc import

```
proc import
    datafile=raw_data
    out=perm.titanic
    dbms=dlm
    replace;
  delimiter='09'x;
  getnames=yes;
run;
```

Notes02. As a general rule, proc import works best for simple delimited files where the first row contains the variable names. With complicated text files (such as files where the data for an individual extends across more than one line) or files without variable names in the first row are usually better handled by a data step.

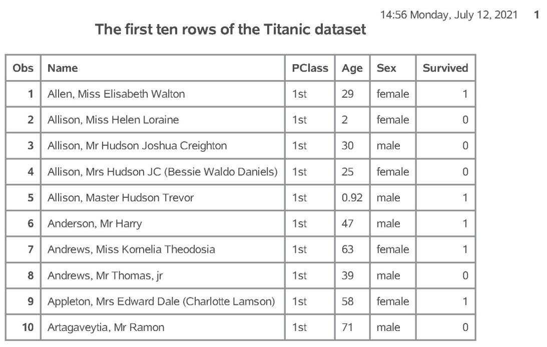


SAS code: Print the first ten lines

```
proc print
    data=perm.titanic(obs=10);
  title1 "The first ten rows of the Titanic
dataset";
run;
```

Notes03. It’s always a good idea to peek at the first few rows of data.

SAS output: Print the first ten lines



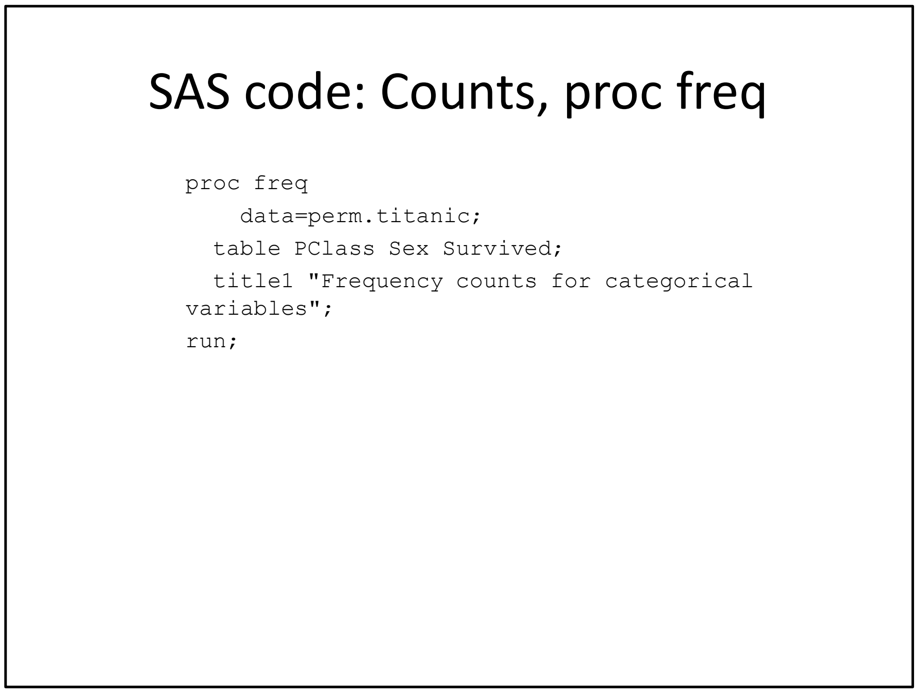
14:56 Monday, July 12, 2021 1

The first ten rows of the Titanic dataset

Obs	Name	PClass	Age	Sex	Survived
1	Allen, Miss Elisabeth Walton	1st	29	female	1
2	Allison, Miss Helen Loraine	1st	2	female	0
3	Allison, Mr Hudson Joshua Creighton	1st	30	male	0
4	Allison, Mrs Hudson JC (Bessie Waldo Daniels)	1st	25	female	0
5	Allison, Master Hudson Trevor	1st	0.92	male	1
6	Anderson, Mr Harry	1st	47	male	1
7	Andrews, Miss Kornelia Theodosia	1st	63	female	1
8	Andrews, Mr Thomas, jr	1st	39	male	0
9	Appleton, Mrs Edward Dale (Charlotte Lamson)	1st	58	female	1
10	Artagaveytia, Mr Ramon	1st	71	male	0

SAS Output

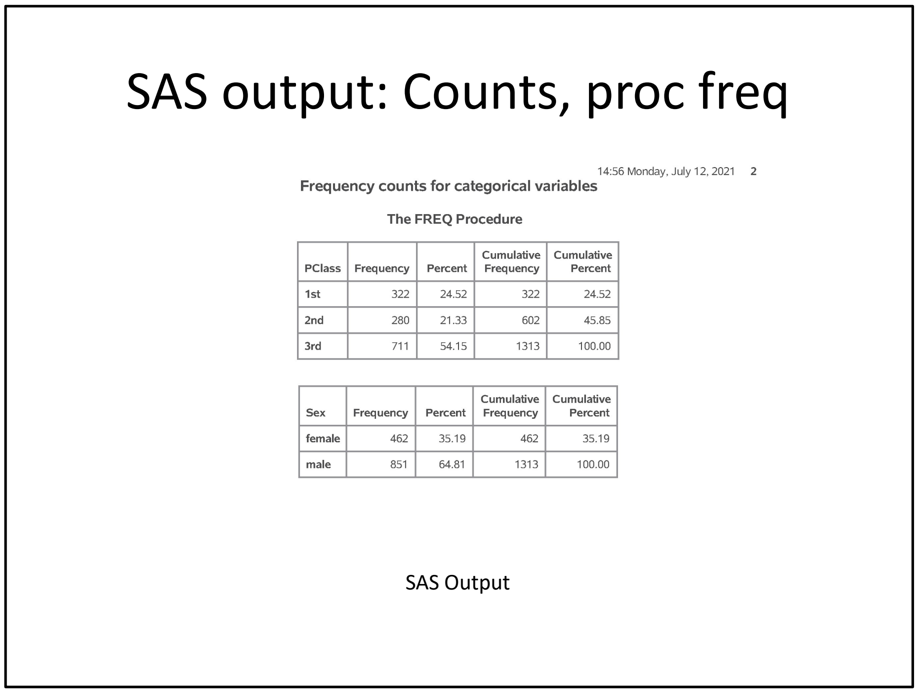
Output, page 1. If you look at the first few rows of data, you will see that the import went reasonably well. It is not always this easy. Do take notice that age is left justified. It is caused by a number of “NA” codes for missing values. You don’t see it here, but if you print a few more observations, you can see the “NA” values. It would have been easier to anticipate these ahead of time, but we’ll fix things up after the fact.



SAS code: Counts, proc freq

```
proc freq
    data=perm.titanic;
  table PClass Sex Survived;
  title1 "Frequency counts for categorical
variables";
run;
```

Notes04. For any categorical variables, your first step is to get frequency counts.



SAS output: Counts, proc freq

14:56 Monday, July 12, 2021 2

Frequency counts for categorical variables

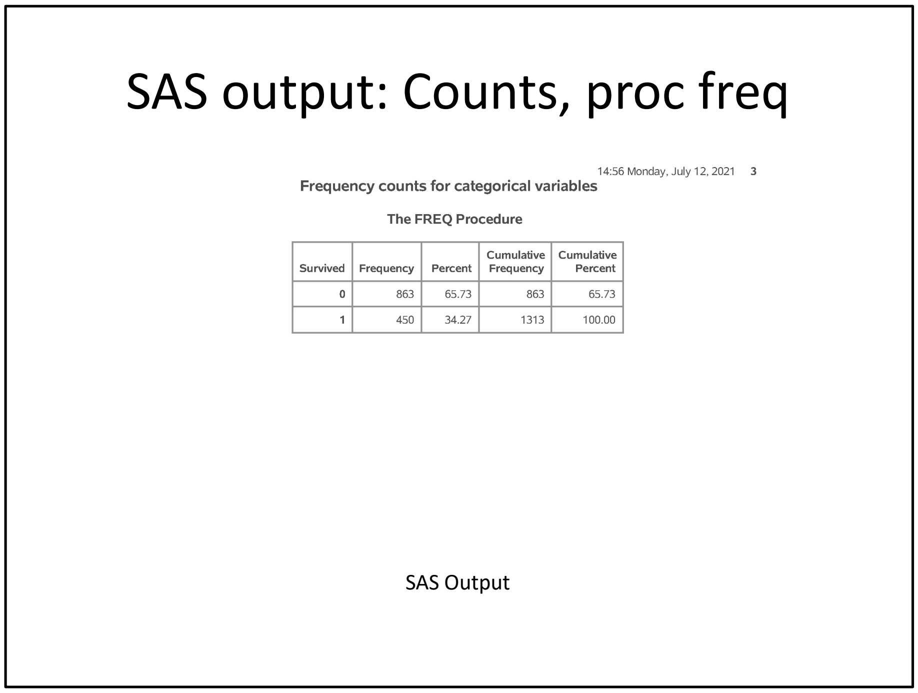
The FREQ Procedure

PClass	Frequency	Percent	Cumulative Frequency	Cumulative Percent
1st	322	24.52	322	24.52
2nd	280	21.33	602	45.85
3rd	711	54.15	1313	100.00

Sex	Frequency	Percent	Cumulative Frequency	Cumulative Percent
female	462	35.19	462	35.19
male	851	64.81	1313	100.00

SAS Output

Output, page 2. There is a mix of three passenger classes, with a lot more in third class. There are also a lot more men than women.



SAS output: Counts, proc freq

14:56 Monday, July 12, 2021 3

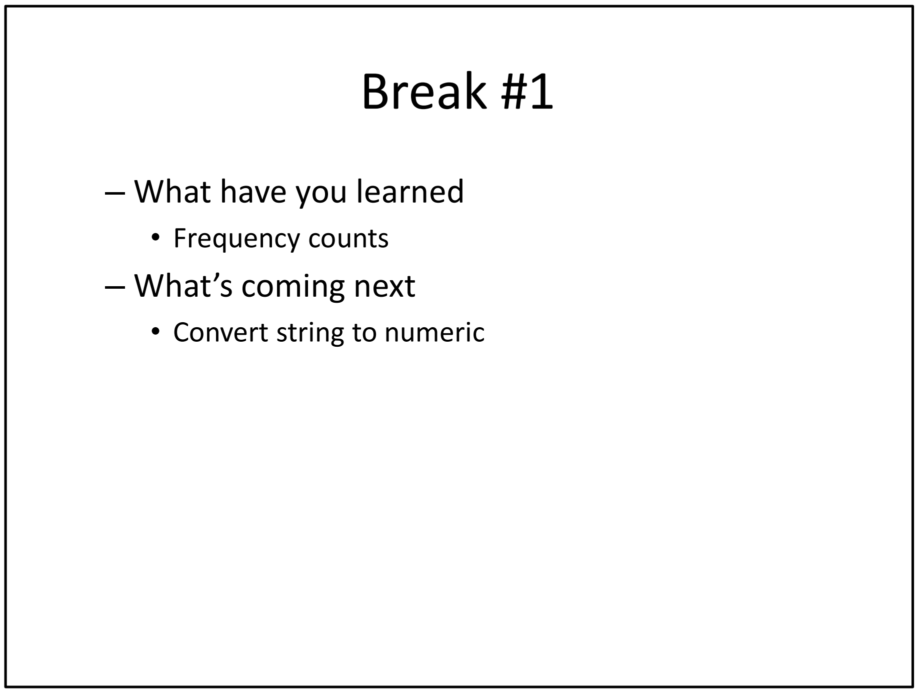
Frequency counts for categorical variables

The FREQ Procedure

Survived	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	863	65.73	863	65.73
1	450	34.27	1313	100.00

SAS Output

Output, page 3. The survived variable is a number code. You should look at the data dictionary to find out that 1=survived and 2=died. Only about a third of the passengers survived.



Break #1

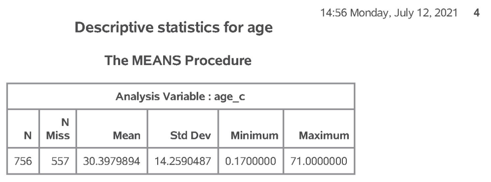
- What have you learned
  - Frequency counts
- What's coming next
  - Convert string to numeric

SAS code: Convert string to numeric, data step

```
data perm.titanic;
  set perm.titanic;
  age_c = input(age, ?? 8.);
run;

proc means
    n nmiss mean std min max
    data=perm.titanic;
  var age_c;
  title1 "Descriptive statistics for age";
run;
```

Notes05. For the one continuous variable (age) you need to convert the code “NA” to the SAS missing value code, which is a dot. The easiest way to do this is to force the data to numeric with a simple arithmetic equation like adding a zero. But you get a warning message for each occurence of NA, which can get tedious. The input function with two question marks avoids this issue.

SAS output: Convert string to numeric, data step



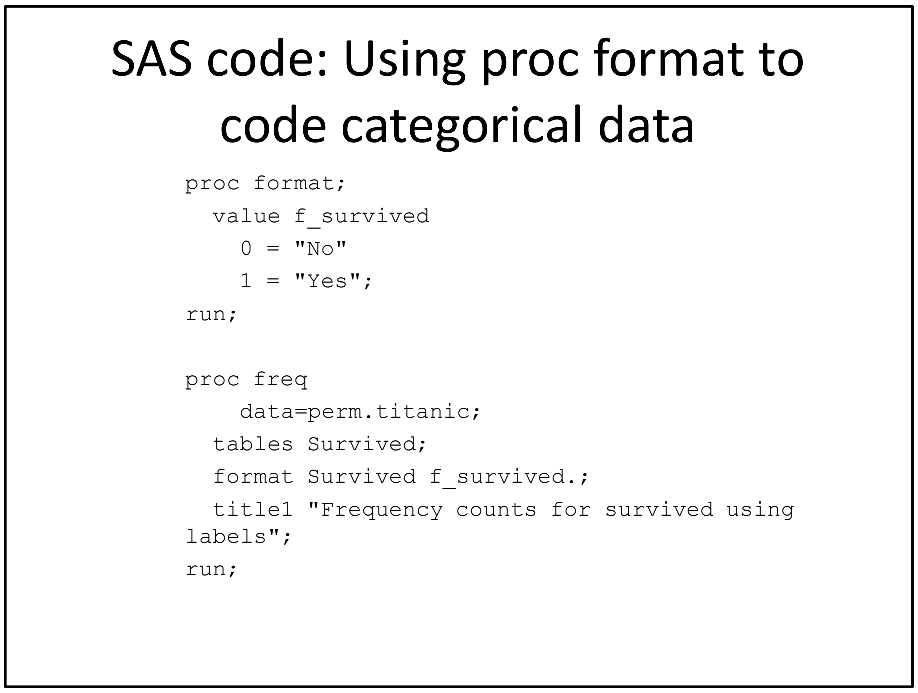
```
                                        14:56 Monday, July 12, 2021    4
           Descriptive statistics for age

              The MEANS Procedure

               Analysis Variable : age_c
       N
  N  Miss       Mean     Std Dev    Minimum    Maximum
756   557 30.3979894  14.2590487  0.1700000 71.0000000
```

SAS Output

Output, page 4. The numeric variable, age_c, can now be analyzed using proc means. The average age is about 30, and there are a large number of missing values.

Break #2

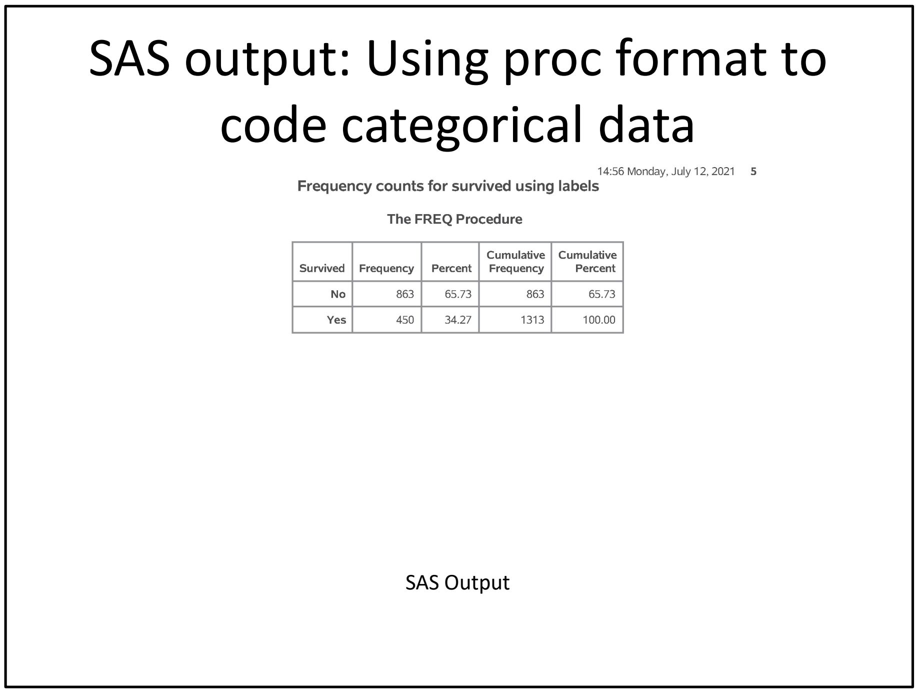
- What you have learned
  - Convert string to numeric
- What's coming next
  - Labels for number codes



SAS code: Using proc format to code categorical data

```
proc format;
  value f_survived
    0 = "No"
    1 = "Yes";
run;

proc freq
    data=perm.titanic;
  tables Survived;
  format Survived f_survived.;
  title1 "Frequency counts for survived using
labels";
run;
```

Notes06. For variables like Survived which are numbers, but the numbers represent a particular category, you can document this using a format statement.



SAS output: Using proc format to code categorical data

14:56 Monday, July 12, 2021 5

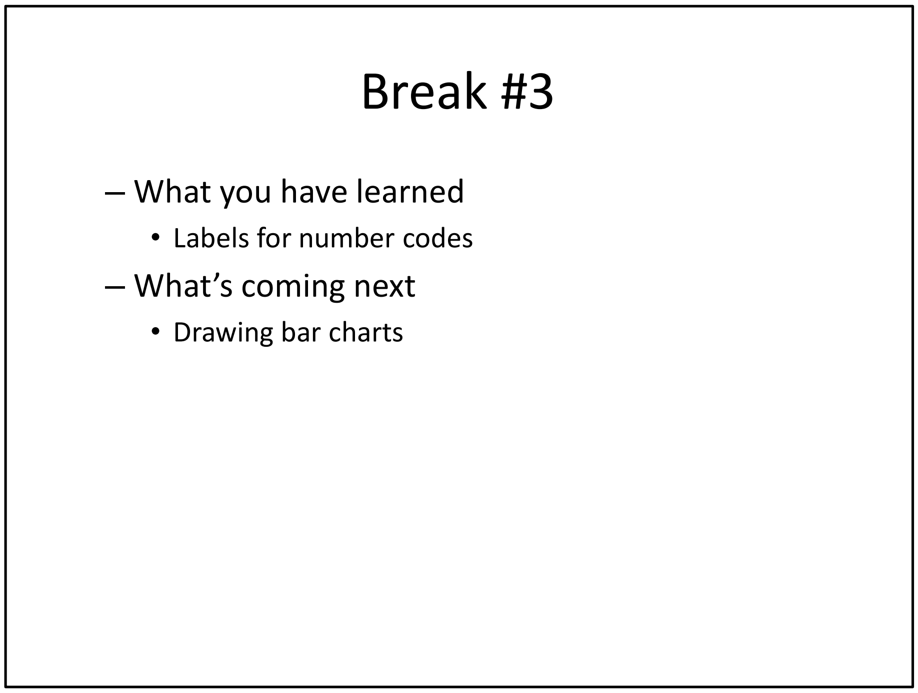
Frequency counts for survived using labels

The FREQ Procedure

Survived	Frequency	Percent	Cumulative Frequency	Cumulative Percent
No	863	65.73	863	65.73
Yes	450	34.27	1313	100.00

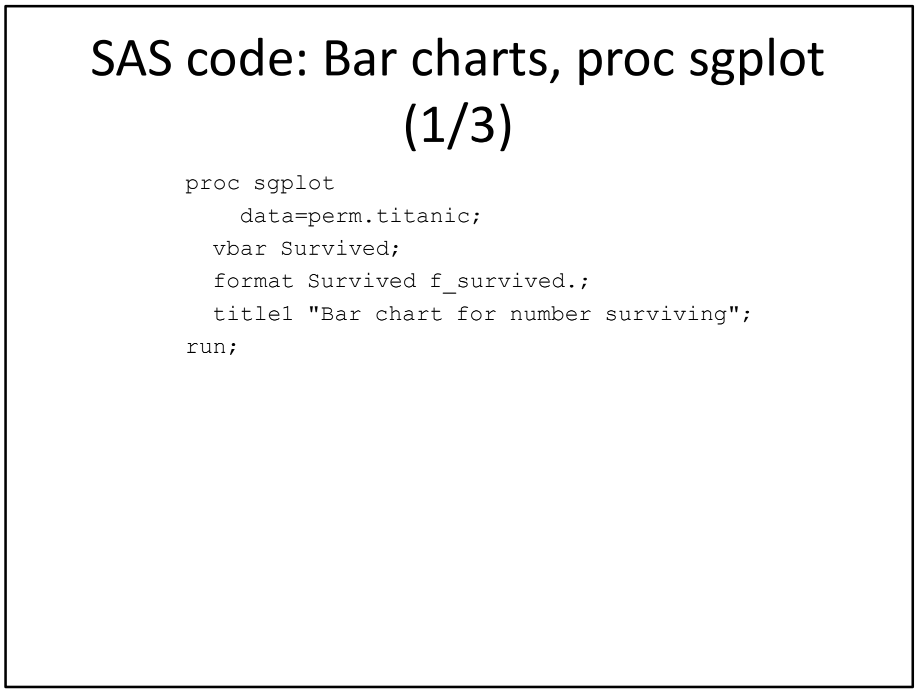
SAS Output

Output, page 5. Notice that the format statement replaces the cryptic 0-1 code with the words no and yes. It is almost always a good idea to attach labels to any categorical variable using number codes. It makes your output more readable and avoids any confusion or mixing up the codes.



Break #3

- What you have learned
  - Labels for number codes
- What's coming next
  - Drawing bar charts



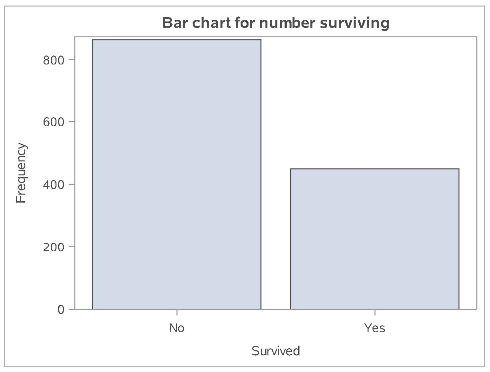
SAS code: Bar charts, proc sgplot (1/3)

```
proc sgplot
    data=perm.titanic;
  vbar Survived;
  format Survived f_survived.;
  title1 "Bar chart for number surviving";
run;
```

Notes07. I don’t normally like bar charts, but they do have their uses.

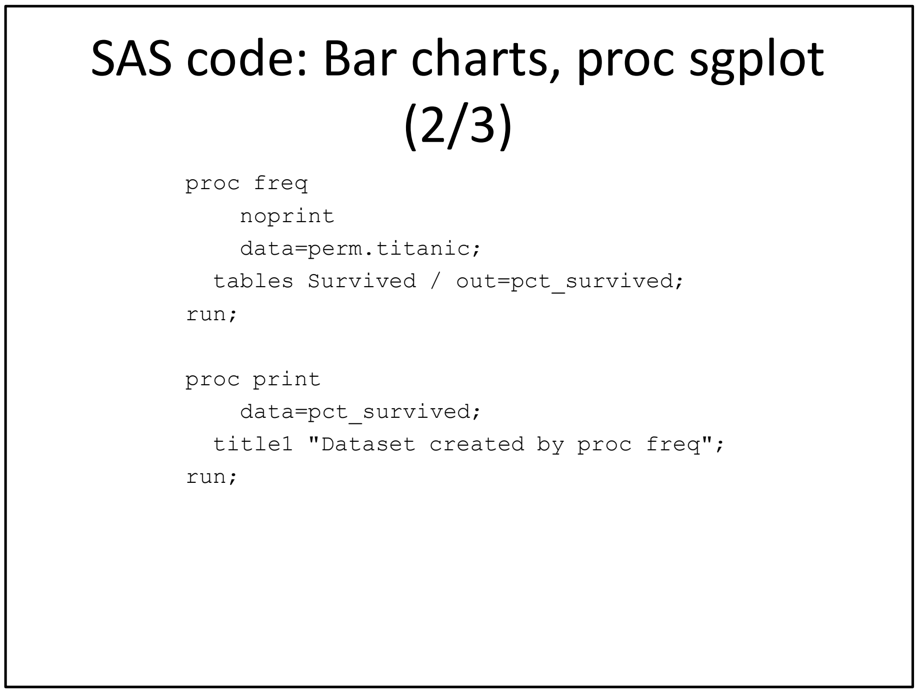
SAS output: Bar charts, proc sgplot (1/3)

14:56 Monday, July 12, 2021 **6**



SAS Output

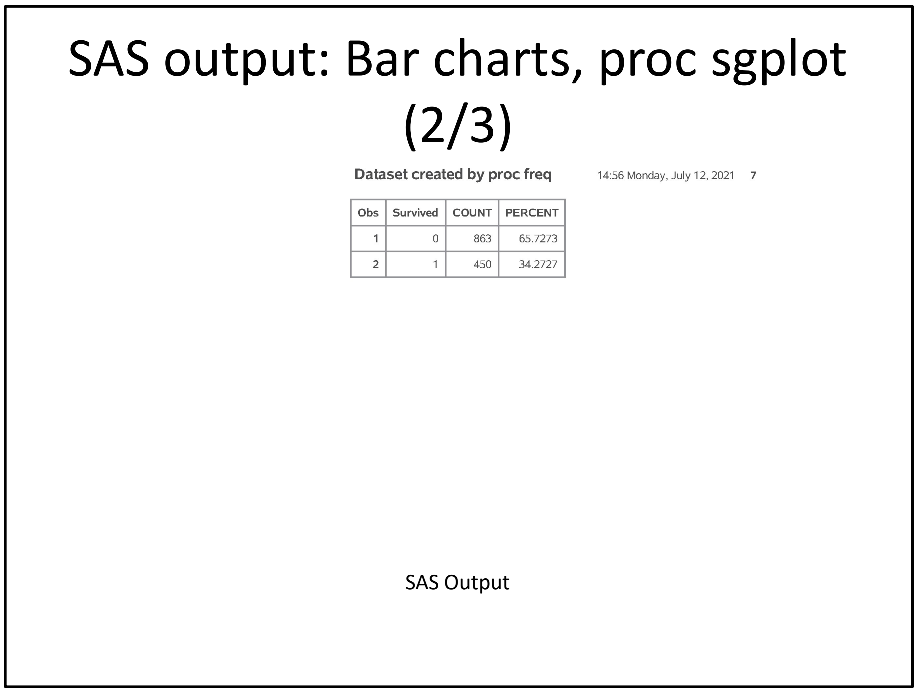
Output, page 6. This is a basic bar chart.



SAS code: Bar charts, proc sgplot (2/3)

```
proc freq
    noprint
    data=perm.titanic;
  tables Survived / out=pct_survived;
run;

proc print
    data=pct_survived;
  title1 "Dataset created by proc freq";
run;
```

Notes08. Getting percentages is a bit tricky. You have to run proc freq and output the results to a new data file, pct_survived. I am using the noprint option, because I only want the percentages for internal use. It wouldn’t have hurt anything to print out a bit extra, but I want to encourage you to limit the amount of output that you present to a consulting client.



SAS output: Bar charts, proc sgplot (2/3)

Dataset created by proc freq 14:56 Monday, July 12, 2021 7

Obs	Survived	COUNT	PERCENT
1	0	863	65.7273
2	1	450	34.2727

SAS Output

Output, page 7. You don’t need to print out this dataset, but I wanted to show it so you can understand what the file looks like. It has the “by” variable, survival, along with COUNT and PERCENT

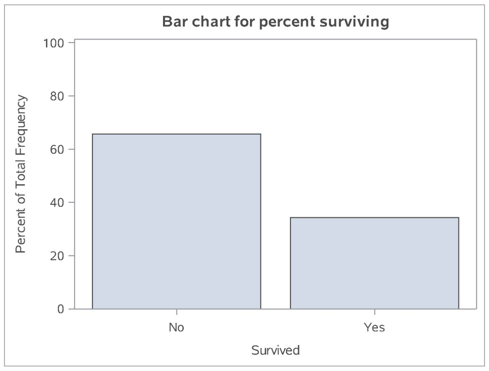
SAS code: Bar charts, proc sgplot (3/3)

```
proc sgplot
    data=pct_survived;
  vbar Survived / response=Percent;
  yaxis max=100;
  format Survived f_survived.;
  title1 "Bar chart for percent surviving";
run;
```

Notes09. You use this newly created dataset to show percentages instead of counts.

SAS output: Bar charts, proc sgplot (3/3)

14:56 Monday, July 12, 2021 **8**

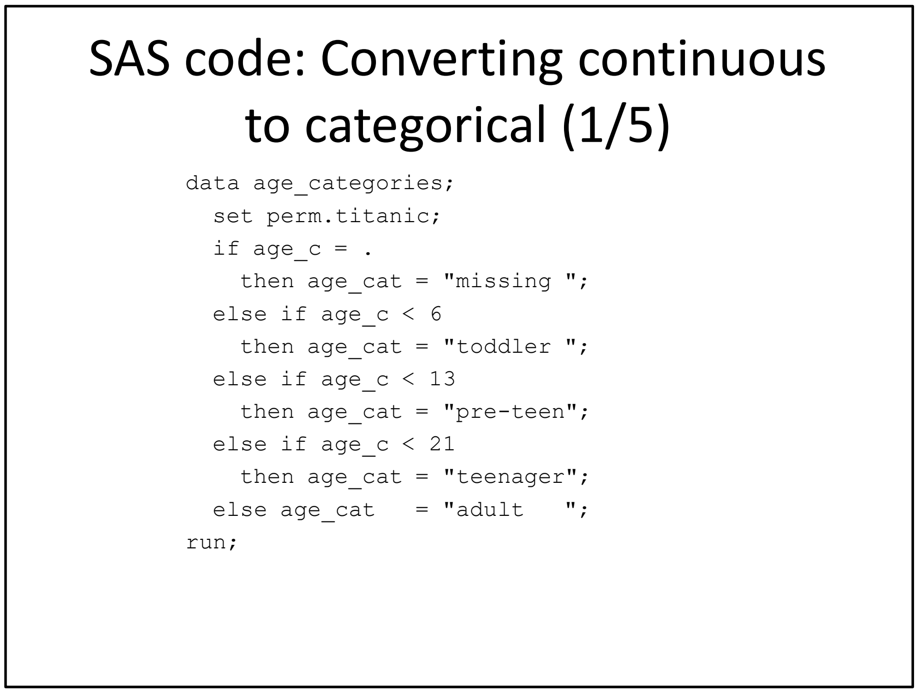


SAS Output

Output, page 8. Note that the yaxis max=100 statement expands the upper limit of the y axis
to 100%.

Break #4

- What you have learned
  - Drawing bar charts
- What's coming next
  - Converting continuous to categorical



SAS code: Converting continuous to categorical (1/5)

```
data age_categories;
  set perm.titanic;
  if age_c = .
    then age_cat = "missing ";
  else if age_c < 6
    then age_cat = "toddler ";
  else if age_c < 13
    then age_cat = "pre-teen";
  else if age_c < 21
    then age_cat = "teenager";
  else age_cat   = "adult   ";
run;
```

Notes10. If you want to create categories from a continuous variable, use a series of

if - then - else

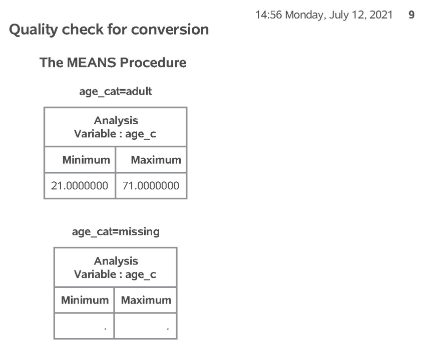
statements

SAS code: Converting continuous to categorical (2/5)

```
proc sort
    data=age_categories;
  by age_cat;
run;

proc means
    min max
    data=age_categories;
  by age_cat;
  var age_c;
  title1 "Quality check for conversion";
run;
```

Notes11. Always cross check your results against the original variable.

SAS output: Converting continuous to categorical (2/5)



```
                                   14:56 Monday, July 12, 2021   9
            Quality check for conversion

              The MEANS Procedure

                 age_cat=adult
```

Analysis Variable : age_c	
Minimum	Maximum
21.0000000	71.0000000

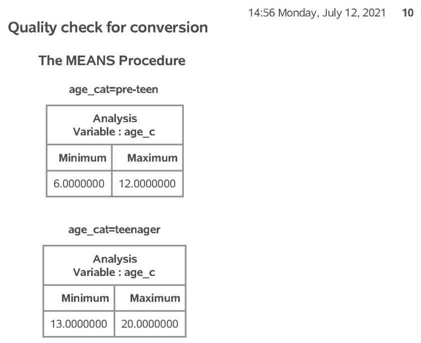
```
                age_cat=missing
```

Analysis Variable : age_c	
Minimum	Maximum
.	.

SAS Output

Output, page 9. These results look good.

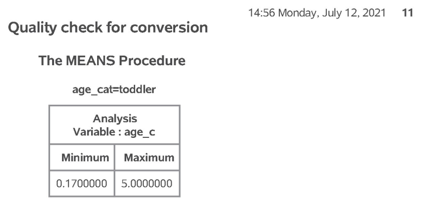
SAS output: Converting continuous to categorical (2/5)


```
                                        14:56 Monday, July 12, 2021   10
            Quality check for conversion

               The MEANS Procedure

               age_cat=pre-teen

              Analysis
           Variable : age_c
        Minimum     Maximum
       6.0000000   12.0000000

               age_cat=teenager

              Analysis
           Variable : age_c
        Minimum     Maximum
       13.0000000  20.0000000
```

SAS Output

output, page 10, And these look good also.

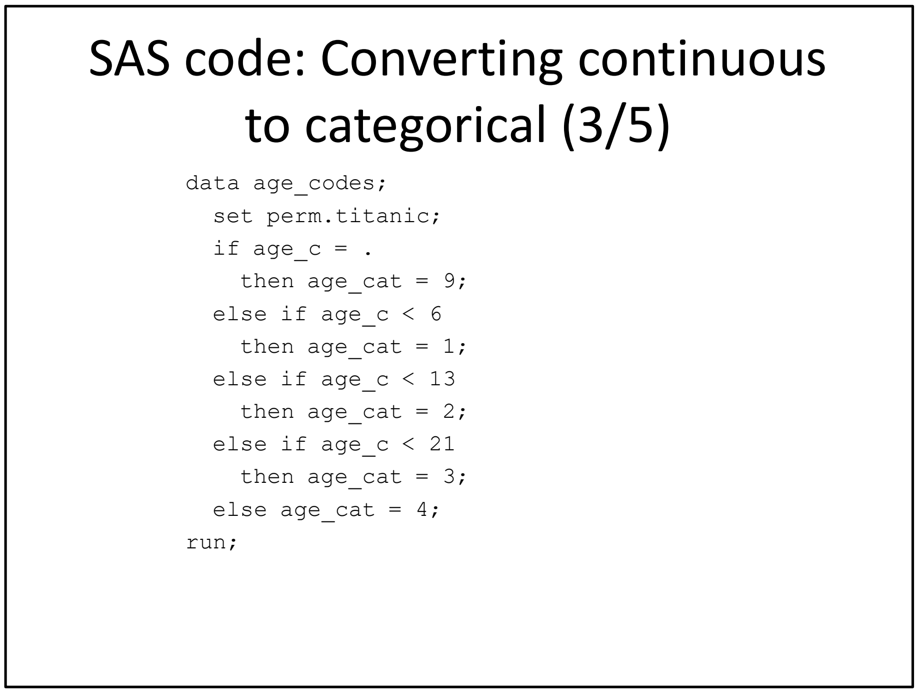
SAS output: Converting continuous to categorical (2/5)



```
                                        14:56 Monday, July 12, 2021   11
            Quality check for conversion

               The MEANS Procedure

                 age_cat=toddler

               Analysis
             Variable : age_c
          Minimum     Maximum
          0.1700000   5.0000000
```

SAS Output

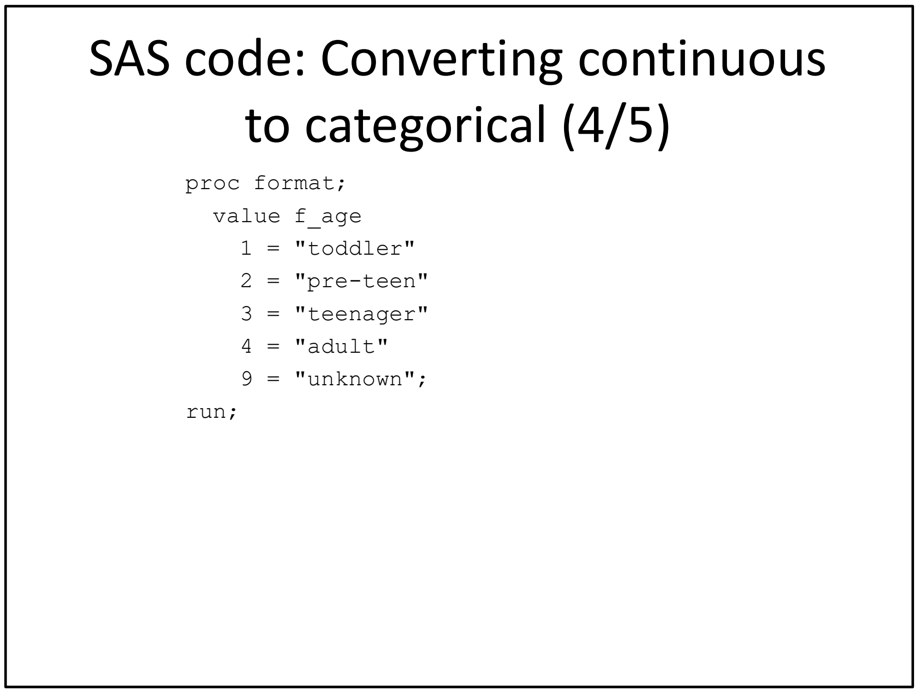
output, page 11, Notice, however, that the order for age_cat is alphabetical, which is probably not what you want.



SAS code: Converting continuous to categorical (3/5)

```
data age_codes;
  set perm.titanic;
  if age_c = .
    then age_cat = 9;
  else if age_c < 6
    then age_cat = 1;
  else if age_c < 13
    then age_cat = 2;
  else if age_c < 21
    then age_cat = 3;
  else age_cat = 4;
run;
```

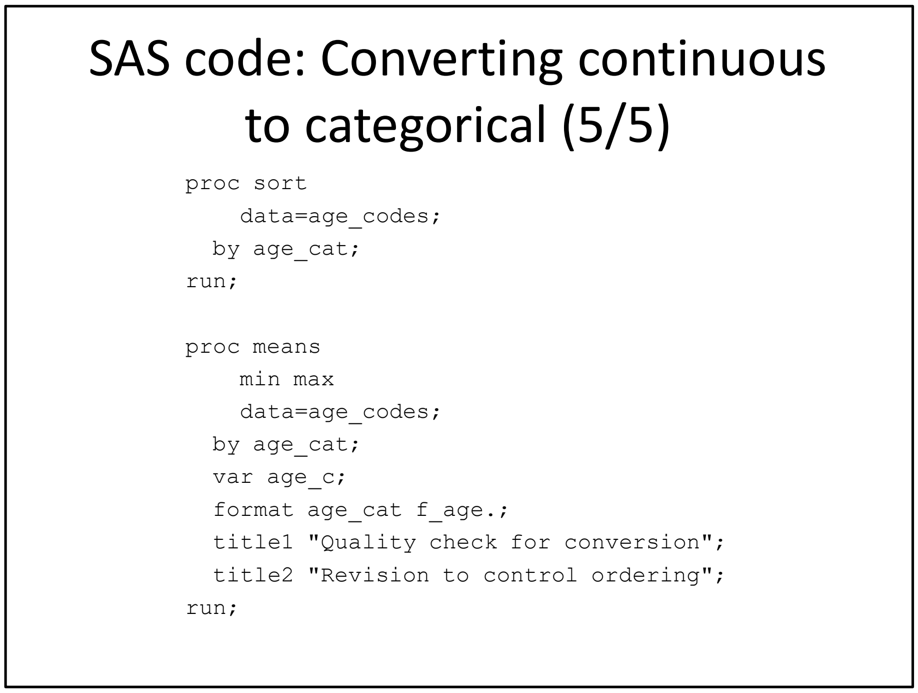
Notes12. You can control the order by using number codes and formats.;



SAS code: Converting continuous to categorical (4/5)

```
proc format;
  value f_age
    1 = "toddler"
    2 = "pre-teen"
    3 = "teenager"
    4 = "adult"
    9 = "unknown";
run;
```

Notes13. The format statement attaches a label to each number code.

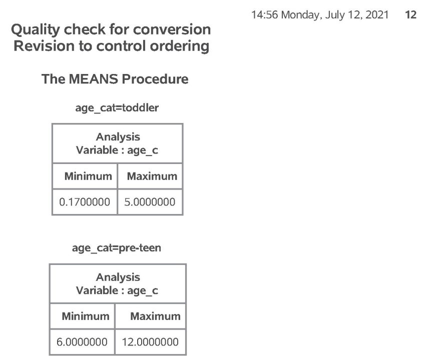


SAS code: Converting continuous to categorical (5/5)

```
proc sort
    data=age_codes;
  by age_cat;
run;

proc means
    min max
    data=age_codes;
  by age_cat;
  var age_c;
  format age_cat f_age.;
  title1 "Quality check for conversion";
  title2 "Revision to control ordering";
run;
```

Notes14. Again, a quality check is important.

SAS output: Converting continuous to categorical (5/5)



```
14:56 Monday, July 12, 2021 12
Quality check for conversion
Revision to control ordering

The MEANS Procedure

age_cat=toddler
```

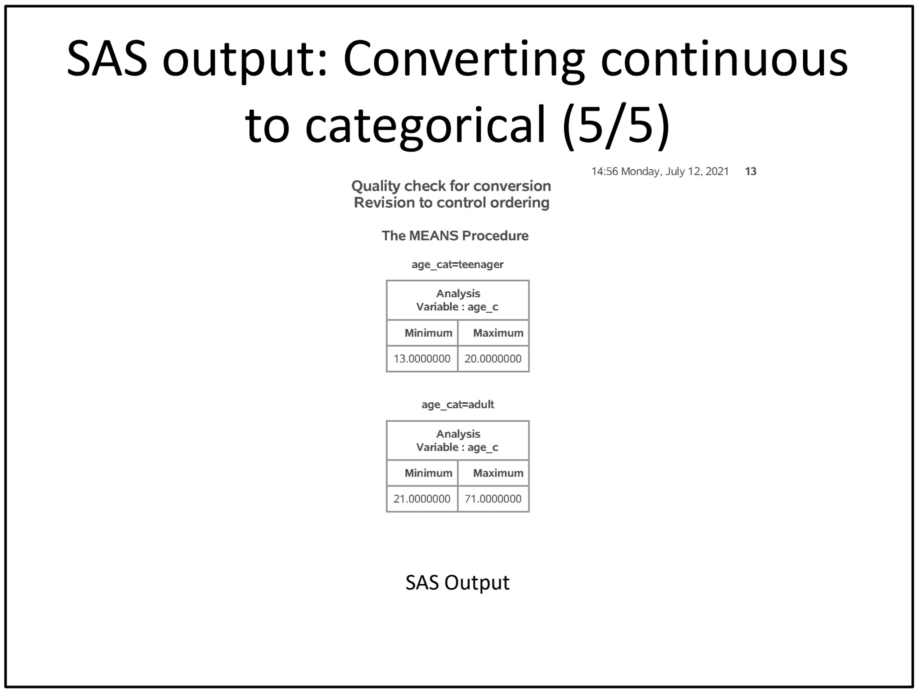
Analysis Variable : age_c	
Minimum	Maximum
0.1700000	5.0000000

```
age_cat=pre-teen
```

Analysis Variable : age_c	
Minimum	Maximum
6.0000000	12.0000000

SAS Output

Output, page 12. The tables are ordered from toddler to pre-teen,

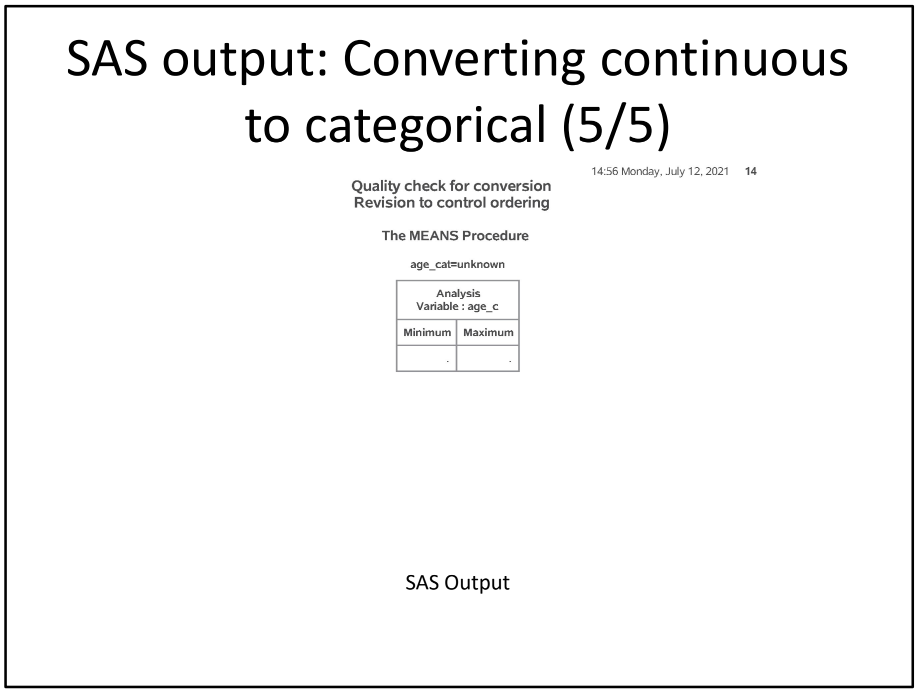


SAS output: Converting continuous to categorical (5/5)

```
                                                  14:56 Monday, July 12, 2021   13
                 Quality check for conversion
                 Revision to control ordering

                     The MEANS Procedure

                      age_cat=teenager

                    Analysis
                 Variable : age_c
           Minimum          Maximum
         13.0000000       20.0000000

                       age_cat=adult

                    Analysis
                 Variable : age_c
           Minimum          Maximum
         21.0000000       71.0000000
```

SAS Output

Output, page 13. to teenager to adult,

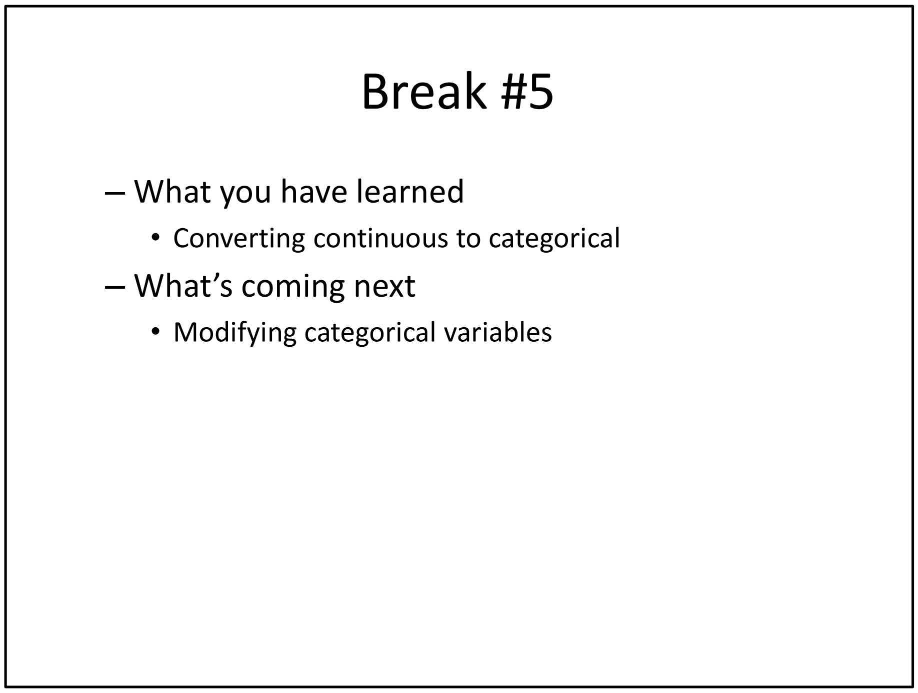


SAS output: Converting continuous to categorical (5/5)

```
                                    14:56 Monday, July 12, 2021    14
          Quality check for conversion
          Revision to control ordering

            The MEANS Procedure

              age_cat=unknown

              Analysis
           Variable : age_c
          Minimum   Maximum
                .         .
```

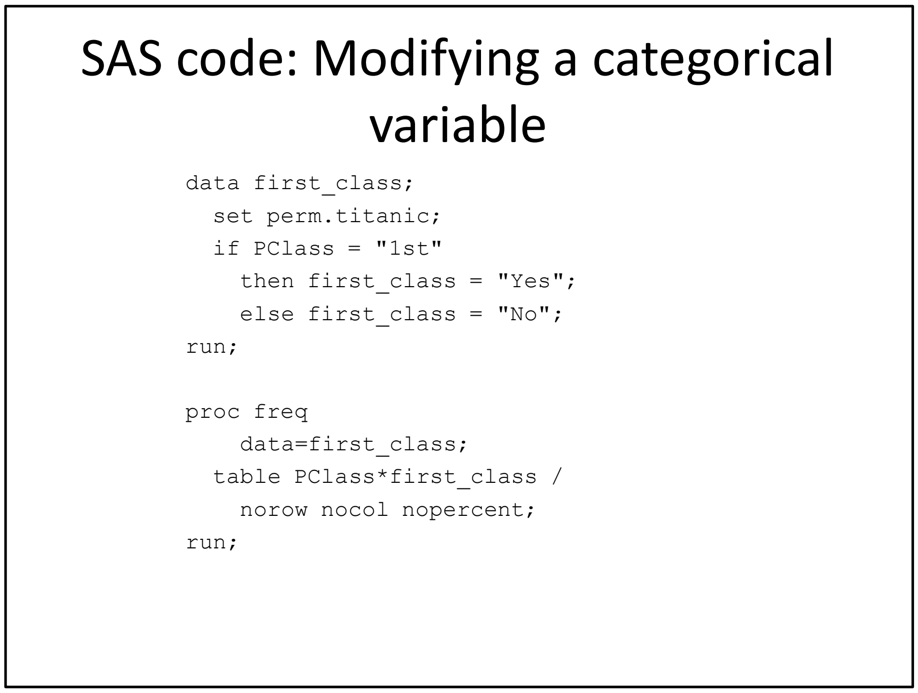
SAS Output

Output, page 14. then missing.



Break #5

- What you have learned
  - Converting continuous to categorical
- What's coming next
  - Modifying categorical variables

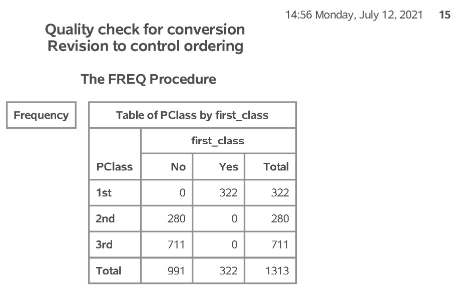


SAS code: Modifying a categorical variable

```
data first_class;
  set perm.titanic;
  if PClass = "1st"
    then first_class = "Yes";
    else first_class = "No";
run;

proc freq
    data=first_class;
  table PClass*first_class /
    norow nocol nopercent;
run;
```

Notes15. Here’s how you combine categories for a categorical variable. It uses the same “if - then - else” code.

SAS output: Modifying a categorical variable



14:56 Monday, July 12, 2021 **15**

Quality check for conversion
Revision to control ordering

The FREQ Procedure

Frequency

Table of PClass by first_class

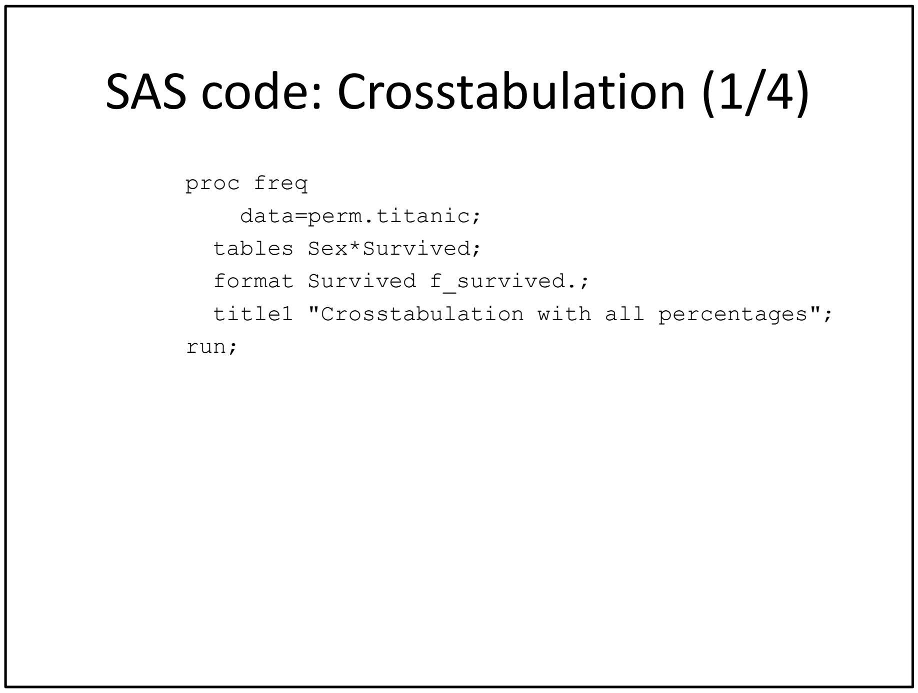
	first_class		
PClass	**No**	**Yes**	**Total**
1st	0	322	322
2nd	280	0	280
3rd	711	0	711
Total	991	322	1313

SAS Output

Output, page 15. Here is a quality check.

Break #6

- What you have learned
  - Modifying categorical variables
- What's coming next
  - Crosstabulation

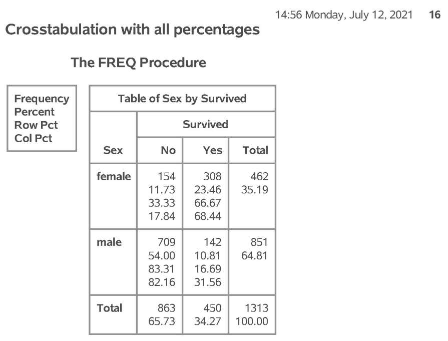


SAS code: Crosstabulation (1/4)

```
proc freq
    data=perm.titanic;
  tables Sex*Survived;
  format Survived f_survived.;
  title1 "Crosstabulation with all percentages";
run;
```

Notes16. To examine relationships among categorical variables use a two dimensional crosstabulation.

SAS output: Crosstabulation (1/4)



14:56 Monday, July 12, 2021 **16**

Crosstabulation with all percentages

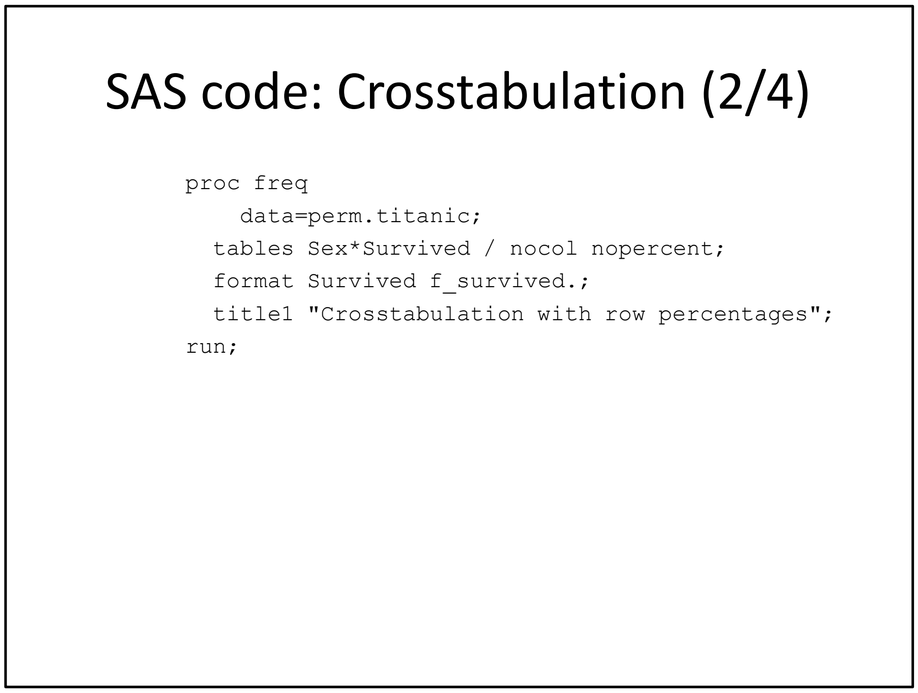
The FREQ Procedure

Frequency
Percent
Row Pct
Col Pct

Table of Sex by Survived			
	Survived		
Sex	No	Yes	Total
female	154 11.73 33.33 17.84	308 23.46 66.67 68.44	462 35.19
male	709 54.00 83.31 82.16	142 10.81 16.69 31.56	851 64.81
Total	863 65.73	450 34.27	1313 100.00

SAS Output

Output, page 15. Here is a quality check.

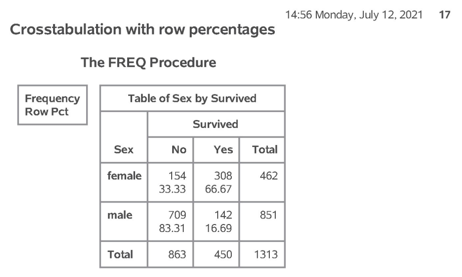


SAS code: Crosstabulation (2/4)

```
proc freq
    data=perm.titanic;
  tables Sex*Survived / nocol nopercent;
  format Survived f_survived.;
  title1 "Crosstabulation with row percentages";
run;
```

Notes17. You get row percentages by excluding column percentages (nocol) and cell percentages (nopercent).

SAS output: Crosstabulation (2/4)



14:56 Monday, July 12, 2021 17

Crosstabulation with row percentages

The FREQ Procedure

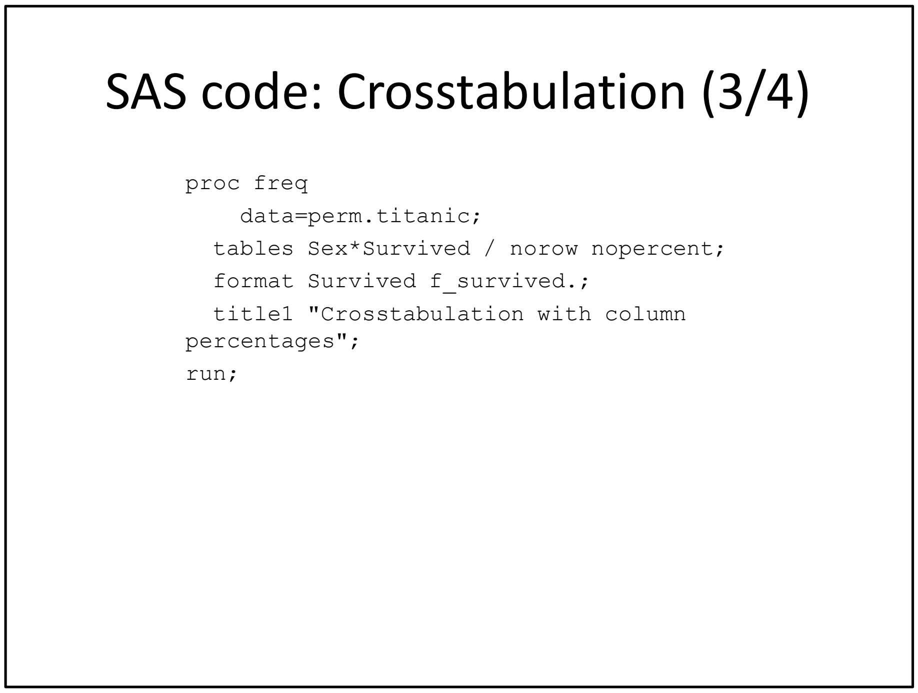
Frequency
Row Pct

Table of Sex by Survived

Sex	Survived: No	Survived: Yes	Total
female	154 33.33	308 66.67	462
male	709 83.31	142 16.69	851
Total	863	450	1313

SAS Output

Output, page 17. Notice that among the men, 83% died and 17% survived. 83% and 17% adds up to 100% within that row. Among the women 33% died and 67% survived. 33% and 67% addes up to 100% within that row.

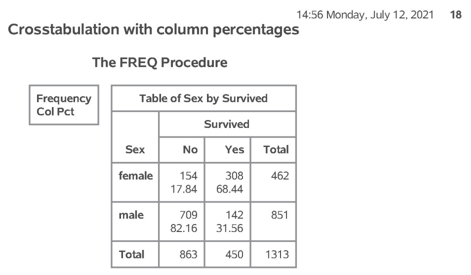


SAS code: Crosstabulation (3/4)

```
proc freq
    data=perm.titanic;
  tables Sex*Survived / norow nopercent;
  format Survived f_survived.;
  title1 "Crosstabulation with column
percentages";
run;
```

Notes18. You get column percentages by excluding row percentages (norow) and cell percentages (nopercent).

SAS output: Crosstabulation (3/4)



14:56 Monday, July 12, 2021 **18**

Crosstabulation with column percentages

The FREQ Procedure

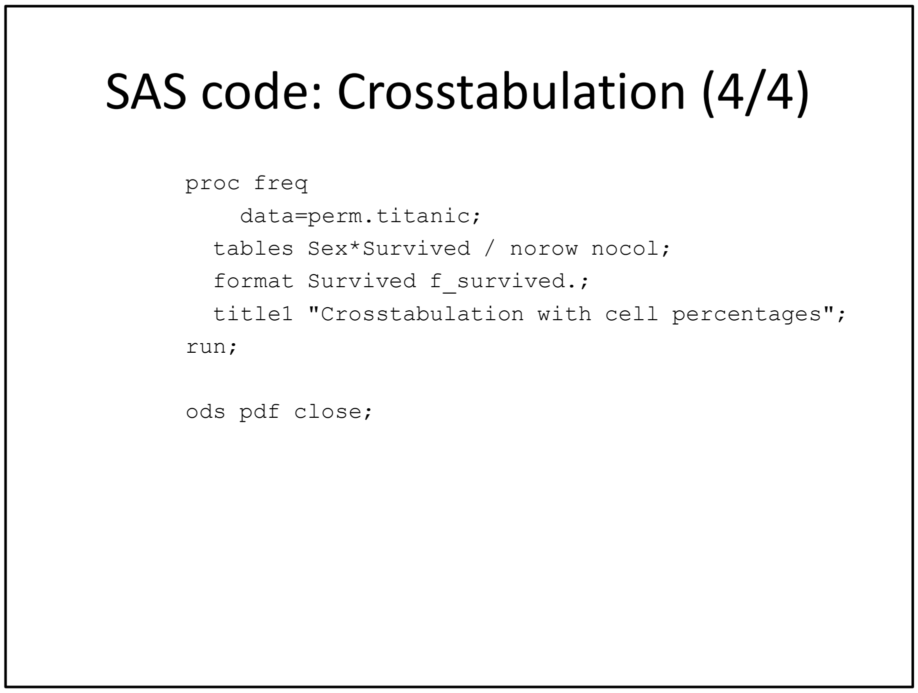
Frequency Col Pct			

Table of Sex by Survived

	Survived		
Sex	**No**	**Yes**	**Total**
female	154 17.84	308 68.44	462
male	709 82.16	142 31.56	851
Total	863	450	1313

SAS Output

Output, page 18. Column percentages add up to 100% within each column. Among those who died, 18% were women and 82% were men. 18% and 82% adds up to 100%. Among the survivors, 68% were women and 32% were men. 68% and 32% adds up to 100%.

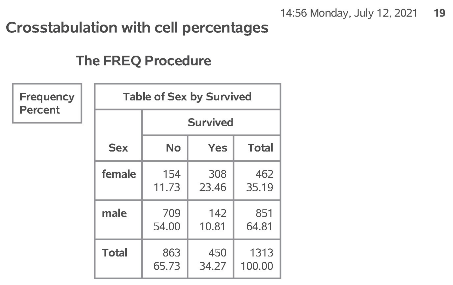


SAS code: Crosstabulation (4/4)

```
proc freq
    data=perm.titanic;
  tables Sex*Survived / norow nocol;
  format Survived f_survived.;
  title1 "Crosstabulation with cell percentages";
run;

ods pdf close;
```

Notes19. You get cell percentages by excluding row percents (norow) and column percents (nocol).

SAS output: Crosstabulation (4/4)



14:56 Monday, July 12, 2021 **19**

Crosstabulation with cell percentages

The FREQ Procedure

Frequency
Percent

Table of Sex by Survived			
	Survived		
Sex	No	Yes	Total
female	154 11.73	308 23.46	462 35.19
male	709 54.00	142 10.81	851 64.81
Total	863 65.73	450 34.27	1313 100.00

SAS Output

Output, page 19. Cell percentages add up to 100% across the entire table. There were 12% female deaths among all the passengers, 54% male deaths, 23% female survivors, and 11% male survivors. These four percentages (12%, 54%, 23%, and 11%) all add up to 100%. Which is best: row, column, or cell percents. The answer is "it depends." I have a handout that talks about these issues.

Review

– Frequency counts
– Convert string to numeric
– Labels for number codes
– Drawing bar charts
– Converting continuous to categorical
– Modifying categorical variables
– Crosstabulations